Applications of Stochastic Programming

Edited by
Stein W. Wallace and
William T. Ziemba

MPS-SIAM Series on Optimization

Applications of Stochastic Programming

MPS-SIAM Series on Optimization

This series is published jointly by the Mathematical Programming Society and the Society for Industrial and Applied Mathematics. It includes research monographs, books on applications, textbooks at all levels, and tutorials. Besides being of high scientific quality, books in the series must advance the understanding and practice of optimization. They must also be written clearly and at an appropriate level.

Series Volumes

Wallace, Stein W. and Ziemba, William T., editors, *Applications of Stochastic Programming*
Grötschel, Martin, editor, *The Sharpest Cut: The Impact of Manfred Padberg and His Work*
Renegar, James, *A Mathematical View of Interior-Point Methods in Convex Optimization*
Ben-Tal, Aharon and Nemirovski, Arkadi, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*
Conn, Andrew R., Gould, Nicholas I. M., and Toint, Phillippe L., *Trust-Region Methods*

Applications of Stochastic Programming

Edited by

Stein W. Wallace
Molde University College
Molde, Norway

and William T. Ziemba
Saunder School of Business
University of British Columbia
Vancouver, BC, Canada

siam
Society for Industrial and Applied Mathematics
Philadelphia

MPS
Mathematical Programming Society
Philadelphia

10 9 8 7 6 5 4 3 2 1

The Stochastic Programming logo on the front cover was created by Ping Lu.

Library of Congress Cataloging-in-Publication Data

Applications of stochastic programming / edited by Stein W. Wallace and William T. Ziemba.
p. cm.— (MPS-SIAM series on optimization)
Includes bibliographical references and index.
ISBN 0-89871-555-5 (pbk.)
1. Stochastic programming. I. Wallace, Stein W., 1956- II. Ziemba, W. T.
III. Series.

T57.79.A66 2005
519.7–dc22

2005042538

List of Contributors

A. Alonso-Ayuso
Escuela de Ciencias Experimentales y Tecnología
Universidad Rey Juan Carlos
28933 Móstoles (Madrid)
Spain
a.alonso@escet.urjc.es

H. Berglann
Department of Economics
University of Bergen
Fosswinckelsgate 6
5007 Bergen
Norway
helge.berglann@econ.uib.no

M. A. H. Dempster
Centre for Financial Research
Judge Institute of Management
University of Cambridge
Cambridge
UK
mahd2@cam.ac.uk

Shi-Jie Deng
School of Industrial and Systems Engineering
Georgia Institute of Technology
765 Ferst Drive
Atlanta, GA 30332
deng@isye.gatech.edu

Jitka Dupacová
Department of Probability and Mathematical Statistics
Charles University Prague
Sokolovská 83
CZ-18675 Prague
Czech Republic
dupacova@karlin.mff.cuni.cz

Robert Entriken
Stanford University
Palo Alto, CA 94305
entriken@stanford.edu

Hafize Gaye Erkan
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544
erkan@princeton.edu

Tatiana Ermolieva
International Institute for Applied Systems Analysis (IIASA)
A-2361 Laxenburg
Austria
ermol@iiasa.ac.at

Yuri Ermoliev
International Institute for Applied Systems Analysis (IIASA)
A-2361 Laxenburg
Austria
ermoliev@iiasa.ac.at

L. F. Escudero
Centro de Investigación-Operativa
Universidad Miguel Hernández
03202 Elche (Alicante)
Spain
escudero@umh.es

S. D. Flåm
Department of Economics
University of Bergen
Fosswinckelsgate 6
5007 Bergen
Norway
sjur.flaam@econ.uib.no

Emmanuel Fragnière
School of Management
University of Bath
Bath BA2 7AY
UK
Mnsef@bath.ac.uk

Karl Frauendorfer
Institute of Operations Research
University of St. Gallen
CH-9000 St. Gallen
Switzerland
karl.frauendorfer@unisg.ch

Alexei A. Gaivoronski
Department of Industrial Economics and Technology Management
Norwegian University of Science and Technology
Alfred Getz vei 1
N-7491 Trondheim
Norway
alexei.gaivoroski@iot.ntnu.no

Horand I. Gassmann
School of Business Administration
Dalhousie University
Halifax, NS B3H 3J5
Canada
hgassmann@mgmt.dal.ca

David M. Gay
AMPL Optimization, LLC
New Providence, NJ 07974
dmg@acm.org

Jacek Gondzio
Department of Mathematics and Statistics
University of Edinburgh
King's Buildings
Edinburgh EH9 3JZ
Scotland
j.gondzio@ed.ac.uk

Nicole Gröwe-Kuska
Institute of Mathematics
Humboldt-University Berlin
10099 Berlin
Germany
nicole@mathematik.hu-berlin.de

Julia L. Higle
Department of Systems and Industrial Engineering
University of Arizona
Tucson, AZ 85721
julie@SIE.Arizona.edu

E. Høeg
The Norwegian Meat Cooperative
Trondheim
Norway
erik.hoeg@gilde.no

Peter Kall
Institute for Operations Research
University of Zurich
CH-8044 Zurich
Switzerland
kall@ior.unizh.ch

Alan J. King
IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598
kingaj@us.ibm.com

P. Krokhmal
Risk Management and Financial Engineering Laboratory
Department of Industrial and Systems Engineering
University of Florida
Gainesville, FL 32611
krokhmal@ufl.edu

Jeff Linderoth
Department of Industrial and Systems Engineering
Lehigh University
200 West Packer Avenue
Bethlehem, PA 18015-1582
jtl3@lehigh.edu

Leonard C. MacLean
School of Business Administration
Dalhousie University
Halifax, NS B3H 3J5
Canada
lmaclean@mgmt.dal.ca

Helmut Mausser
Algorithmics, Inc.
185 Spadina Avenue
Toronto, ON M5T 2C6
Canada
hmausser@algorithmics.com

János Mayer
Institute for Operations Research
University of Zurich
CH-8044 Zurich
Switzerland
mayer@ior.unizh.ch

Daene C. McKinney
Department of Civil Engineering
The University of Texas at Austin
Austin, TX 78712
daene_mckinney@mail.utexas.edu

E. A. Medova
Cambridge Systems Associates Limited and Center for Financial Research
Judge Institute of Management
University of Cambridge
Cambridge
UK
eam28@cam.ac.uk

G. Mitra
Center for the Analysis of Risk and Optimisation Modelling Applications (CARISMA)
Brunel University
Uxbridge UB8 3PH
UK
mastggm@brunel.ac.uk

David P. Morton
Graduate Program in Operations Research
The University of Texas at Austin
Austin, TX 78712
morton@mail.utexas.edu

John M. Mulvey
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544
mulvey@princeton.edu

Shmuel S. Oren
Industrial Engineering and Operations Research
University of California at Berkeley
4135 Etcheverry Hall
Berkeley, CA 94720
oren@ieor.berkeley.edu

M. T. Ortuño
Departamento de Estadística e Investigación Operativa
University Complutense de Madrid
28040 Madrid
Spain
mteresa@mat.ucm.es

Gyana R. Parija
IBM Thomas J. Watson Research Center
Yorktown Heights, NY 10598
parija@us.ibm.com

A. B. Philpott
Department of Engineering Science
University of Auckland
Auckland
New Zealand
a.philpott@auckland.ac.nz

C. A. Poojari
Center for the Analysis of Risk and Optimisation Modelling Applications (CARISMA)
Brunel University
Uxbridge UB8 3PH
UK
mapgcap@brunel.ac.uk

Pavel Popela
Department of Mathematics
Brno University of Technology
Technická 2
CZ-61669 Brno
Czech Republic
popela@fme.vutbr.cz

Warren B. Powell
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544
powell@princeton.edu

Werner Römisch
Institute of Mathematics
Humboldt-University Berlin
10099 Berlin
Germany
romisch@mathematik.hu-berlin.de

Dan Rosen
Algorithmics, Inc.
185 Spadina Avenue
Toronto, ON M5T 2C6
Canada
drosen@algorithmics.com

Michale Schürle
Institute of Operations Research
University of St. Gallen
CH-9000 St. Gallen
Switzerland
michael.schuerle@unisg.ch

J. E. Scott
Center for Financial Research
Judge Institute of Management
University of Cambridge
Cambridge
UK
jes23@cam.ac.uk

Sandra L. Schwartz
SLS Consultants
6 Tamath Crescent
Vancouver, BC V6N 2C9
Canada
schwartz@interchange.ubc.ca

A. Sembos
Credit Suisse First Boston
London
UK
asembos@hotmail.com

Suvrajeet Sen
Department of Systems and Industrial Engineering
University of Arizona
Tucson, AZ 85721
sen@sie.arizona.edu

László Somlyódy
Budapest University of Technology and Economics
H-111 Budapest
Hungary
somlyody@vsct.bme.hu

G. W. P. Thompson
Center for Financial Research
Judge Institute of Management
University of Cambridge
Cambridge
UK
gwpt1@cam.ac.uk

A. Tomasgard
SINTEF Industrial Management and Technology Managment
Norwegian University of Science and Technology
Trondheim
Norway
at@iot.ntnu.no

Huseyin Topaloglu
Department of Operations Research and Industrial Engineering
Cornell University
Ithaca, NY 14853
huseyin@orie.cornell.edu

S. Uryasev
Risk Management and Financial Engineering Laboratory
Department of Industrial and Systems Engineering
University of Florida
Gainesville, FL 32611
uryasev@ise.ufl.edu

P. Valente
Center for the Analysis of Risk and Optimisation Modelling Applications (CARISMA)
Brunel University
Uxbridge UN8 3PH
UK
masrppv@brunel.ac.uk

Stein W. Wallace
Department of Quantitative Logistics
Molde University College
N-6405 Molde
Norway
stein.wallace@himolde.no

David W. Watkins, Jr.
Department of Civil and Environmental Engineering
Michigan Technological University
Houghton, MI 49931
dwatkins@mtu.edu

Roger J.-B. Wets
University of California at Davis
Davis, CA 95616
rjbw@math.ucdavis.edu

Stephen E. Wright
Miami University
Oxford, OH 45056
wrightse@muohio.edu

Stephen J. Wright
Computer Sciences Department
University of Wisconsin
1210 West Dayton Street
Madison, WI 53706
swright@cs.wisc.edu

Gary W. Yohe
Department of Economics
Wesleyan University
Middletown, CT 06459
gyohe@wesleyan.edu

Stavros A. Zenios
HERMES Center on Computational Finance and Economics
University of Cyprus
P.O. Box 20537
1678 Nicosia
Cyprus
zenioss@ucy.ac.cy

Yonggan Zhao
Nanyang Business School
Nanyang Technological University
639798
Singapore
aygzhao@ntu.edu.sg

William T. Ziemba
Saunder School of Business
2053 Main Mall
University of British Columbia
Vancouver, BC V6T 1Z2
Canada
ziemba@interchange.ubc.ca

G. Zrazhevsky
Risk Management and Financial Engineering Laboratory
Department of Industrial and Systems Engineering
University of Florida
Gainesville, FL 32611

This page intentionally left blank

Contents

This page intentionally left blank

Preface

Most practical decision problems involve uncertainty. Stochastic programming is the study of procedures for decision making under uncertainty over time. The uncertainty can be in the model's parameters or in the model itself. Parameters may be uncertain because of lack of reliable data, measurement errors, future and unobservable events, etc. The uncertainty of events, details of the problem structures and constraints, and the risk/payoff of decisions are modeled in an optimization framework. Uncertainty is modeled over time using scenarios that approximate the future. High-performance workstations and PCs are used to enable exact and approximate algorithms to determine robust decisions that hedge against future uncertainty. Then as the uncertainty becomes known period by period, recourse decisions responding to the new information can be made.

Since the early papers of Dantzig [6], Beale [1], and Charnes and Cooper [5], stochastic programming has grown into a very important subfield of mathematical programming with well-established theoretical developments. Research on algorithms and applications has also been very active, especially in recent years. There has been a growing number of specialists in the area, and knowledge is widespread among the leaders of the field. However, the full power of stochastic programming has not reached as wide an audience as is desirable, especially in application areas where the approach is superior to many other modeling avenues. Understanding of the problems of using scenario analysis (sensitivity analysis or parametric analysis) and the like (see, for example, [13]) is still not common in the optimization community. Moreover, it is difficult for nonstochastic programming specialists and beginning stochastic programming students to easily get to the frontiers. The field has excellent theoretical texts such as [3, 4, 7, 8, 10, 11, 12], as well as conference books such as [2, 14, 15], which have extensive lists of references to other books, special issues of journals, etc. There are also application books in subareas such as [16, 17, 18] for financial modeling and asset-liability management. However, a good algorithms and applications book that allows readers access to publicly available stochastic programming algorithmic systems and high-level applications in broad areas is lacking. This book provides such a resource, which is intended to be useful for beginning students, for researchers from other optimization and application areas, as a supplement in stochastic programming courses, and as a reference work for professionals in the area. The book is a contribution from many of the leading stochastic programming specialists to the field and as such is to be viewed as a collective contribution to the field.

The book has two parts. The first is a presentation of many of the publicly available stochastic programming systems that are currently operational. All of these codes have been extensively tested and developed over a period of years. They are now available for

researchers and developers who want to make models without extensive programming and other implementation costs. Using such systems, one can formulate important problems and solve them; an example, using the IBM Stochastic Solutions code of Alan King, is Geyer et al. [9], which is a pension fund application that one of us was involved with and that would not have been successfully implemented in the allowed budget without this code. We hope that there will be many more such applications in the future. These codes do not constitute all available systems. Rather they are a synopsis of the best systems with the requirement for inclusion that they be user-friendly, ready to go, and publicly available. Special thanks go to Horand (Gus) Gassmann for helping us with this section of the book with his special knowledge concerning stochastic programming codes. Gus also worked very hard refereeing the papers and getting them ready for publication in this form.

The second part of the book is a collection of a large and diversified set of application papers. These applications are in areas such as production, supply chain and scheduling, gaming, environmental and pollution control, financial modeling, telecommunications, and electricity. We were delighted that the authors of these papers shared their experience with us. Sandra Schwartz helped us prepare these papers for production in this volume. We are pleased that she and Gus joined us in the introduction and other aspects of this section. Special thanks go to three anonymous referees for many helpful comments on the various papers and introductions to this volume.

The manuscript was completed while William Ziemba was Nomura Visiting Fellow in Financial Mathematics at the Nomura Centre for Quantative Finance at the Mathematical Institute, Oxford University. Thanks are due to Dr. Sam Howison, the director, for very generous hospitality and to Keith Gillow, Sara Jolliffe, and Angela Howard for their help with this project. Thanks also to the Master, Roger Ainsworth, John Ockendon, and the other fellows, staff, and friends of St. Catherine's College for their gracious hospitality during his stay as a Christensen visiting fellow.

We hope that this book is a useful addition to the stochastic programming literature and will allow established researchers to do more, and new researchers to build and use stochastic programming models.

Stein W. Wallace
Molde, Norway

William T. Ziemba
Vancouver, BC, and Oxford, UK

Bibliography

[1] E. Beale, *On minimizing a convex function subject to linear inequalities*, J. Roy. Statist. Soc. Ser. B, 17 (1955).

[2] J. B. Birge, N. C. P. Edirisinghe, and W. T. Ziemba, eds., *Research in Stochastic Programming*, Ann. Oper. Res. 100, Baltzer Science Publishers, Amsterdam, 2000.

[3] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer-Verlag, New York, 1997.

[4] Y. Censor and S. A. Zenios, *Optimization: Theory, Algorithms and Applications*, Ser. Numer. Math. Sci. Comput., Oxford University Press, New York, 1997.

[5] A. Charnes and W. W. Cooper, *Chance constrained programming*, Management Sci., 6 (1958).

[6] G. B. Dantzig, *Linear programming under uncertainty*, Management Sci., 1 (1955), pp. 197–206.

[7] J. Dupačová, J. Hurt, and J. Stepan, *Stochastic Modeling in Economics and Finance*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002.

[8] Y. Ermoliev and R. J.-B. Wets, eds., *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin, 1988.

[9] A. Geyer, W. Herold, K. Kontriner, and W. T. Ziemba, *The Innovest Austrian Pension Fund Financial Planning Model InnoALM*, University of Vienna, 2002.

[10] J. L. Higle and S. Sen, *Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Programming*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[11] P. Kall and S. W. Wallace, *Stochastic Programming*, John Wiley, Chichester, UK, 1994.

[12] A. Prékopa, *Stochastic Programming*, Kluwer Academic Publishers, Dordrecht, The Netherlands, and Academiai Kiado, Budapest, 1995.

[13] S. Wallace, *Decision making under uncertainty: Is sensitivity analysis of any use?*, Oper. Res., 48 (2000), pp. 20–25.

[14] S. Wallace, J. Higle, and S. Sen, eds., *Stochastic Programming, Algorithms and Models*, Ann. Oper. Res. 64, Baltzer Science Publishers, Amsterdam, 1996.

[15] R. J.-B. Wets and W. T. Ziemba, eds., *Stochastic Programming: State of the Art* 1998, Ann. Oper. Res. 85, Baltzer Science Publishers, Amsterdam, 1999.

[16] S. A. Zenios, ed., *Financial Optimization*, Cambridge University Press, Cambridge, UK, 1993.

[17] W. T. Ziemba, *The Stochastic Programming Approach to Asset-Liability and Wealth Management*, AIMR, Charlottesville, VA, 2003.

[18] W. T. Ziemba and J. M. Mulvey, eds., *World Wide Asset and Liability Modeling*, Cambridge University Press, Cambridge, UK, 1998.

This page intentionally left blank

Part I

Stochastic Programming Codes

This page intentionally left blank

Chapter 1

Stochastic Programming Computer Implementations

Horand I. Gassmann, Stein W. Wallace,† and William T. Ziemba‡*

Stochastic programming is decision making under risk. This means that some of the model coefficients are random variables with known or estimated distributions whose realizations are revealed (perhaps gradually) after some or all of the decisions have been made.

Since most chapters in this book use similar models, we collect the most common ones here for easier reference and to avoid some duplication later. Further information is contained in the textbooks by Birge and Louveaux [2], Censor and Zenios [4], Kall and Wallace [7], and Prékopa [8].

The starting point is the (static) mathematical program

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq 0, \\ & l \leq x \leq u, \end{aligned}$$

where $f : R^n \to R$, $g : R^n \to R^m$, and x, l, and u are in R^n. The presence of random variables in this problem necessitates changes to the components as follows:

1. The timing and sequence of decisions and revealing of specific realizations of random variables is crucial. A stage is a point in time where some decision variables are set. A stage is followed by an event epoch where some random variables are fixed

*School of Business Administration, Dalhousie University, Halifax, NS, B3H 3J5, Canada (hgassmann@mgmt.dal.ca).

†Molde University College, N-6402 Molde, Norway (stein.wallace@himolde.no).

‡Faculty of Commerce, University of British Columbia, Vancouver, BC, V6T 1Z2, Canada (ziemba@interchange.ubc.ca).

at particular values according to a known probability distribution. Some further decisions can then be made at the next stage, etc.

2. The objective function is replaced by an expected value,

$$\min \mathrm{E}(f(x_1, \xi_1, x_2, \xi_2, \ldots, x_T, \xi_T)),$$

or some other function of the decisions and random variables.

3. Constraints containing random coefficients that are revealed only after the last decision has been made are subject to a probabilistic interpretation taking one of two forms:

$$\begin{aligned} \mathrm{P}\{g_i(x, s) > 0\} &\leq \epsilon, \\ \mathrm{E}_s\{g_i(x, s) \mid g_i(x, s) > 0\} &\leq h. \end{aligned}$$

4. Constraints containing random coefficients revealed before the last decision is taken are required only to hold almost surely (a.s.).

Special cases of this problem are the chance-constrained problem of Charnes and Cooper [3] and the multistage recourse problem formulated independently by Dantzig [5] and Beale [1].

The one-stage chance constrained problem is

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g_1(x) \leq 0, \\ & \mathrm{P}_\xi\{g_2(x, \xi) > 0\} \leq \epsilon, \\ & l \leq x \leq u, \end{aligned}$$

where the probability is taken with respect to the distribution of the random variable ξ. For approaches to multistage chance-constrained models, see Prékopa [8].

The multistage recourse problem is

$$\begin{aligned} \min \quad & f_1(x_1) + \mathrm{E}_{\xi_1}\Big[\min f_2(x_1, \xi_1, x_2(x_1, \xi_1)) \\ & \qquad + \mathrm{E}_{\xi_2|\xi_1}\big[\min f_3(x_1, \xi_1, x_2(x_1, \xi_1), \xi_2, x_3(x_1, \xi_1, x_2(x_1, \xi_1), \xi_2)) \\ & \qquad + \cdots + \mathrm{E}_{\xi_{T-1}|\xi_1,\ldots,\xi_{T-2}}[\min f_T(x_1, \xi_1, \ldots, x_T(x_1, \xi_1, \ldots, \xi_{T-1})]\ldots\big]\Big] \\ \text{s.t.} \quad & g_1(x_1) \leq 0, \\ & g_2(x_1, \xi_1, x_2(x_1, \xi_1)) \leq 0 \quad \text{a.s.}, \\ & \ldots \\ & g_T(x_1, \xi_1, x_2(x_1, \xi_1), \xi_2, \ldots, x_T(x_1, \xi_1, x_2(x_1, \xi_1), \xi_2, \ldots, \xi_{T-1})) \leq 0 \quad \text{a.s.}, \\ & l_1 \leq x_1 \leq u_1, \\ & l_2(\xi_1) \leq x_2(x_1, \xi_1) \leq u_2(\xi_1) \quad \text{a.s.}, \\ & \ldots \\ & l_T(\xi_1, \ldots, \xi_{T-1}) \leq x_T(x_1, \xi_1, \ldots, \xi_{T-1}) \leq u_T(\xi_1, \ldots, \xi_{T-1}) \quad \text{a.s.} \end{aligned} \tag{1.1}$$

This problem can also be written in the *recursive form*

$$\begin{aligned} \min \quad & f_1(x_1) + \mathcal{Q}_1(x_1) \\ \text{s.t.} \quad & g_1(x_1) \le 0, \\ & l_1 \le x_1 \le u_1, \end{aligned} \tag{1.2}$$

where $\mathcal{Q}_1$ and similarly $\mathcal{Q}_2, \ldots, \mathcal{Q}_{T-1}$ are defined by

$$\begin{aligned} &\mathcal{Q}_t(x_1, \xi_1, x_2(x_1, \xi_1), \ldots, x_t(x_1, \ldots, \xi_{t-1})) \\ &\quad = \mathrm{E}_{\xi_t|\xi_1,\ldots,\xi_{t-1}} \mathcal{Q}_t(x_1, \xi_1, x_2(x_1, \xi_1), \ldots, x_t(x_1, \ldots, \xi_{t-1}), \xi_t), \\ &\quad t = 1, \ldots, T-1, \end{aligned} \tag{1.3}$$

and for $t = 1, \ldots, T-1$

$$\begin{aligned} &\mathcal{Q}_t(x_1, \xi_1, x_2(x_1, \xi_1), \ldots, x_t(x_1, \ldots, \xi_{t-1}), \xi_t) \\ &\quad = \min \quad f_{t+1}(x_1, \xi_1, x_2(x_1, \xi_1), \ldots, x_{t+1}(x_1, \ldots, \xi_t)) \\ &\qquad\qquad + \mathcal{Q}_{t+1}(x_1, \xi_1, x_2(x_1, \xi_1), \ldots, x_{t+1}(x_1, \ldots, \xi_t)) \\ &\qquad \text{s.t.} \quad g_{t+1}(x_1, \xi_1, x_2(x_1, \xi_1), \ldots, x_{t+1}(x_1, \ldots, \xi_t)) \le 0, \\ &\qquad\qquad l_{t+1}(\xi_1, \ldots, \xi_t) \le x_{t+1}(x_1, \ldots, \xi_t) \le u_{t+1}(\xi_1, \ldots, \xi_t). \end{aligned} \tag{1.4}$$

The recursion is anchored at T by setting $\mathcal{Q}_T \equiv 0$. (If there are other end effects to be considered, we assume that they are captured in the definition of f_T.)

If all the random variables are finitely distributed and the objective and constraints are linear functions, then the multistage recourse problem can be rewritten as the large-scale structured equivalent deterministic problem

$$\begin{aligned} \min \quad & c_1 x_1 + \sum_{i=2}^{N_2} p_i c_i x_i + \sum_{i=N_2+1}^{N_3} p_i c_i x_i + \cdots + \sum_{i=N_{T-1}+1}^{N_T} p_i c_i x_i \\ \text{s.t.} \quad & A_{11} x_1 \le b_1, \\ & A_{21i} x_1 + A_{22i} x_i \le b_i, \quad i = 2, \ldots, N_2, \\ & A_{31i} x_1 + A_{32i} x_{a_2(i)} + A_{33i} x_i \le b_i, \quad i = N_2 + 1, \ldots, N_3, \\ & \cdots \\ & A_{T1i} x_1 + A_{T2i} x_{a_2(i)} + \cdots + A_{T,T-1,i} x_{a_{T-1}(i)} \le b_i, \quad i = N_{T-1} + 1, \ldots, N_T, \\ & l_i \le x_i \le u_i, \quad i = 1, \ldots, N_T, \end{aligned} \tag{1.5}$$

where A_{sti}, b_i, c_i, l_i, and u_i are realizations of the random variables and the notation $a_t(i)$ is used to describe the unique predecessor in stage t corresponding to the history of events leading up to the particular realization i. The probabilities p_i are *path probabilities*, that is, the probability of observing the entire sequence of realizations leading up to node i. Hence we have $\sum_{i=N_t+1}^{N_{t+1}} p_i = 1$ for $t = 1, \ldots, T-1$.

This formulation makes use of the fact that the evolution of the data process can be represented in the finite case by an event tree, whose nodes can be numbered consecutively period by period. A small example with three stages is given in Figure 1.1. Here $a_2(4) = a_2(5) = 2$, $a_2(6) = 3$. Most of the models in this book take this form.

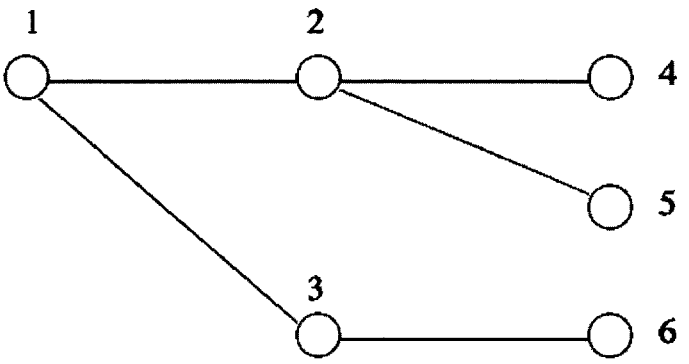


Figure 1.1. *A three-stage event tree.*

Some special cases of the linear multistage recourse problem deserve separate mention. If $A_{sti} = 0$ for $s < t - 1$, the problem is said to possess a *staircase structure*. This structure simplifies many decomposition algorithms and is assumed by some solvers. Other solvers assume that the problem possesses *complete recourse*, which means that for every choice of decision variables $x_1, \ldots, x_t$ and every evolution $\xi_1, \ldots, \xi_{T-1}$ of the data process there exist recourse decisions $x_{t+1}, \ldots, x_T$ for which the problem is feasible. Problems for which A_{tti} is constant for all i are said to have *fixed recourse*. If $A_{tti} = [I \quad -I]$ for all i, the problem is said to have *simple recourse*. In such problems, which occur frequently in practice, there is either a shortage or a surplus, so the second-stage problem has an obvious solution. Further details can be found in the book by Birge and Louveaux [2].

Because of the size of stochastic programs, specialized software packages that can exploit the structure are much faster (often by several orders of magnitude) than general optimization solvers. Additionally, the size of the models requires specialized aid in problem formulation, data and model management, solver selection, report generation and analysis, and related topics. This first part of the book discusses some of these issues.

Even in today's world of Internet browsing and sophisticated graphical interfaces, mathematical programming problems must be presented to a computer solver in a specific format that it understands. For deterministic problems, standards have evolved so that one formulation of a problem can be used with a variety of solvers. To the extent that a deterministic equivalent problem exists, stochastic programming problems can be formulated using these same standards. For instance, a linear recourse problem is essentially a large-scale linear program, for which an adequate format exists in the well-known MPS format.

As mentioned, stochastic programs are often highly structured, and it is essential for efficient solution to describe and use this structure. The chapter by Gassmann (Chapter 2) describes an extension to the MPS format that allows the capture and description of stochastic data. Most solvers of stochastic programs can access problems formulated in this so-called SMPS format.

Table 1.1 gives an overview of the most easily available software tools. (The list is not complete.) Systems marked with an asterisk (*) in the last column are discussed in more detail in the rest of Part I.

The most frequently used stochastic programming model is the multistage recourse

Table 1.1. *Current development of software tools for SP.*

Name	Affiliation	System name
J. J. Bisschop et al.	Paragon Decision Technology	AIMMS
A. Meeraus et al.	GAMS	GAMS
B. Kristjansson	Maximal Software	MPL
R. Fourer et al.	Northwestern University	AMPL
M. A. H. Dempster et al.	Cambridge University	*STOCHGEN
E. Fragnière et al.	University of Geneva	*SETSTOCH
A. King et al.	IBM	*OSL/SE
H. I. Gassmann et al.	Dalhousie University	MSLiP
G. Infanger et al.	Stanford University	DECIS
P. Kall et al.	University of Zürich	*SLP-IOR
G. Mitra et al.	Brunel University	*SPInE
A. Gaivoronski	Norwegian University of Science and Technology	*SQG

formulation, and the most successful method for solving such problems is by some variant of Benders decomposition. Two commercial solvers exist: OSL/SE, which is described in the chapter by King et al. (Chapter 3), and DECIS, developed by Infanger of Stanford University. DECIS can handle two-stage problems only; OSL/SE is suitable for multiple stages.

A somewhat less stable implementation is MSLiP, available to researchers from Gassmann (hgassman@mgmt.dal.ca). Users can also submit problems to the NEOS solver (http://www-neos.mcs.anl.gov/neos/server-solvers.html) over the Internet. This requires the preparation of three input files (in SMPS format) which can be transmitted using a web browser. The solution will be returned by e-mail. For fully interactive sessions, a Java add-on can be downloaded from the NEOS site.

Two-stage recourse problems with continuous distributions pose a special challenge because the deterministic equivalent cannot be written explicitly. Higle and Sen [6] developed a form of Benders decomposition applicable to this class of problems; a different algorithm based on stochastic quasi gradients was developed by Gaivoronski. His implementation is described in a separate chapter (Chapter 4).

The subproblems arising in Benders decomposition exhibit a high degree of parallelization. One very affordable approach to the parallel solution of such problems is described in Chapter 5 by Linderoth and Wright.

A variety of solvers are available through the SLP-IOR environment that can support the entire life cycle of a problem, from the earliest implementation to the solution, maintenance, modification, and archiving. This program is described in Chapter 6 by Kall and Mayer. SLP-IOR supports problem development in the algebraic modeling language GAMS, it reads and writes files in SMPS format, and it is connected to numerous solvers for various classes of stochastic programming problems.

But the process of writing an MPS file can be tedious. Algebraic modeling languages (AMLs) have been developed since the 1980s to help with this process and to perform consistency checks, etc. They use notation closely tied to the algebraic formulation of the model, often transcribing mathematical symbols into character strings that can be represented on an ordinary keyboard. Algebraic modeling languages also occasionally use a

simplified file format to communicate directly with the solvers. Examples of AMLs are GAMS, AMPL, MPL, AIMMS, Modler, and LPL.

Any deterministic equivalent problem can be formulated using an AML. However, as with the MPS format, the special structure of stochastic programs is lost in this process. To date there is no convenient way to express stochastic programs within an AML in such a way that they could be transferred directly to a specialized stochastic programming solver. This is an active area of research, and some partial results are described in the papers by Fragniere and Gondzio and Valente, Mitra, and Poojan (Chapters 7 and 8, respectively).

The final two Part I chapters by Dempster, Scott, and Thompson (Chapter 9) and Gassmann and Gay (Chapter 10) describe integrated production environments that link together multiple commercially available software packages, such as spreadsheets, databases, algebraic modeling languages, and solvers, in an attempt to provide the diverse users of stochastic models with interfaces that are tailored to their specific needs. Both systems employ nested decomposition solvers; Dempster, Scott, and Thompson use the specialized solver STOCHGEN, while Gassmann and Gay explore the feasibility of implementing the decomposition directly in the algebraic modeling language AMPL and using the general-purpose LP solver CPLEX to solve the subproblems.

Bibliography

[1] E. M. L. BEALE, *On minimizing a convex function subject to linear inequalities*, J. Roy. Statist. Soc. Ser. B, 17 (1955), pp. 173–184.

[2] J. R. BIRGE AND F. LOUVEAUX, *Introduction to Stochastic Programming*, Springer-Verlag, New York, 1997.

[3] A. CHARNES AND W. W. COOPER, *Chance-constrained programming*, Management Sci., 5 (1959), pp. 73–79.

[4] Y. CENSOR AND S. A. ZENIOS, *Optimization: Theory, Algorithms and Applications*, Ser. Numer. Math. Sci. Comput., Oxford University Press, New York, 1997.

[5] G. B. DANTZIG, *Linear programming under uncertainty*, Management Sci., 1 (1955), pp. 197–206.

[6] J. HIGLE AND S. SEN, *Stochastic Programming, Algorithms and Models*, Kluwer Academic Publishers, Hingham, MA, 1996.

[7] P. KALL AND S. W. WALLACE, *Stochastic Programming*, John Wiley, Chichester, UK, 1994.

[8] A. PRÉKOPA, *Stochastic Programming*, Kluwer Academic Publishers, Dordrecht, The Netherlands, and Academiai Kiado, Budapest, 1995.

Chapter 2

The SMPS Format for Stochastic Linear Programs

*Horand I. Gassmann**

This is a brief introduction to the SMPS format, which can be used to formulate a wide variety of stochastic linear and integer programming problems. A full description of the format can be found in [3] or on the SMPS web page [2]. An earlier version of the format was published by Birge et al. [1].

The use of the format is illustrated with snippets of sample input throughout this chapter. However, the format is very complex, and an exhaustive illustration of every nuance would greatly increase the length of this presentation. For that reason, only the most commonly used features are shown. Additional examples may be gleaned from [2].

The SMPS format was designed to make it easy to convert existing deterministic linear programs into stochastic ones by adding information about the dynamic and stochastic structure. This is done using three text files: the core file, the time file, and the stochastics file (or stoch file). Each file consists of several sections, which must appear in predefined order. The first record of each section is a header record, which is followed by zero or more data records. Empty sections (containing no data) may be omitted. The records in each file are organized into fields according to the MPS line layout [4].

In the MPS format, one must distinguish between header lines and data lines. Header lines contain up to three name fields, normally in these positions:

columns 1–14	first word field
columns 15–24	second word field
columns 40–49	third word field

Data lines contain up to three name fields, two numeric fields, and a code field, normally in these positions:

*School of Business Administration, Dalhousie University, Halifax, NS, B3H 3J5, Canada (horand.gassmann@dal.ca).

columns 2 and 3	code field
columns 5–12	first name field
columns 15–22	second name field
columns 25–36	first numeric field
columns 40–47	third name field
columns 50–61	second numeric field

Names are normally restricted to eight characters and may contain any ASCII symbol; numerical input is limited to twelve characters (including digits, signs, exponents, and a decimal point). Free format input can be used if these limits are too restrictive.

The core file

The core file lists all the deterministic information of the problem, in standard MPS format. Core file information includes the name and type of each constraint and variable, a column-ordered list of coefficients in the constraint matrix, right-hand-side coefficients, and any bounds and ranges on the variables and slacks. The core file also provides placeholders for all the stochastic elements (which must be mentioned in the core file and be given a preliminary value that may or may not be meaningful). The core file thus represents a deterministic problem, which may be thought of as a typical scenario, an average case, a baseline case, or something similar.

Further information about the MPS format may be found in the IBM documentation [4]. The extensions to integer variables and quadratic objectives described there are also supported in SMPS. There also is a network format, based on the NETGEN format by Klingman, Napier, and Stutz [7]. It is even possible to mix LP and network sections in the same file.

The time file

The purpose of the time file is to allow breaking down of the core file scenario into nodes corresponding to the individual stages. The time file contains the following sections:

Section	Purpose
TIME	Gives a name to the problem in name field 2
PERIODS	Gives the name of each time period (in order)
ROWS	(optional) Specifies for each row the period to which it belongs
COLUMNS	(optional) Specifies for each column the period to which it belongs
ENDATA	End of problem data

TIME section

The second name field on the TIME header record contains the name of the problem. This name is verified against the name in the core file.

PERIODS section

The header record contains the value PERIODS in the first name field. If the core file is in time order, then the second name field can be left blank or given the value IMPLICIT. The ROWS and COLUMNS sections are not needed in this case.

Each data record contains the name of a period in the third name field. If the time file is in implicit format, then the first name field contains the name of the first column in this period, and the second name field contains the name of the first row. If the time file is in explicit format, the first two name fields are not used. The objective row must always belong to the first period.

Below is an example of the start of a time file in temporal order. In this example all columns in the core file that appear between COL1 (inclusive) and COL5 (exclusive) make up the first period, named PER1, while COL5 and all the remaining columns belong to period PER2. Similarly, ROW1 and ROW2 (along with the objective) are the rows belonging to the first period; all the others are second-period rows.

Example

```
*23456789 123456789 123456789 123456789 123456789 123456789
PERIODS        IMPLICIT
    COL1       ROW1                    PER1
    COL5       ROW3                    PER2
```

If the core file is not given in temporal order (for instance, if it was generated by an algebraic modeling language), then the keyword EXPLICIT must be placed in the second name field of the header record and the ROWS and COLUMNS sections must be present.

ROWS section

The data records in this section contain the name of a row mentioned in the core file in the first name field and the name of a time period in the second name field. The ROWS section is optional if the core file is given in temporal order.

Example

```
*23456789 123456789 123456789 123456789 123456789 123456789
ROWS
    ROW1       PER1
...
```

COLUMNS section

The data records in this section contain the name of a column mentioned in the core file in the first name field; the second name field contains the name of a time period set up in the preceding PERIODS section. The COLUMNS section is optional if the core file is given in temporal order.

The stoch file

The stoch file allows the solver to build a deterministic equivalent, which requires information about the random variables. If all the random variables are finitely distributed, this task amounts to the construction of an event tree. The event tree can be described in three different ways: scenario by scenario, node by node, or implicitly using marginal distributions. Implicit descriptions are also available if the random variables are continuously distributed, where explicit descriptions would obviously fail.

The stoch file may contain the following sections:

Section	Purpose
STOCH	Gives a name to the problem in name field 2
SIMPLE	For problems with simple recourse (optional)
ROBUST	For quadratic penalties (optional)
PLINQUAD	Piecewise linear-quadratic penalties (optional)
CHANCE	For chance constraints (optional)
ICC	For integrated chance constraints (optional)
SCENARIOS	Gives the values of stochastic elements one scenario at a time
NODES	Gives the values of stochastic elements one node at a time
INDEP	Marginal distribution of independent random variables
BLOCKS	Marginal distribution of a random vector
DISTRIB	Distribution of auxiliary random variables
ENDATA	End of problem data

The three ways of describing the event tree are mutually exclusive. That is, a stoch file cannot contain SCENARIOS, NODES, and INDEP/BLOCKS sections simultaneously. INDEP, BLOCKS, and DISTRIB sections may be mixed freely, although the order must be such that enough information is available to construct the event tree during a single pass through the file.

STOCH section

The second name field on the STOCH header record contains the name of the problem. This name is verified against the name in the core file.

SIMPLE section for simple recourse

Simple recourse problems use very special recourse matrices, which should not have to be set up by the user explicitly. This section allows the user to set up appropriate penalties for violating a constraint. The corresponding slack and surplus variables along with their constraint coefficients are automatically generated by the system.

In most applications the recourse costs will be deterministic, but if the situation calls for stochastic recourse costs, the user can reference the generated column names in a distribution section later. For that reason the SIMPLE section precedes the INDEP, BLOCKS, and DISTRIB sections.

Example

```
*23456789 123456789 123456789 123456789 123456789 123456789
SIMPLE
    SIMPLE1   ROW1       10.0                        5.0
    SIMPLE1   ROW2       50.0
```

Assume that ROW1 is an E-type row (equality constraint) and ROW2 has type G greater than or equal to contraint. Then each unit of shortage in ROW1 incurs a penalty of 10.0 units in the objective, and each unit of surplus has a penalty of 5. Surplus in ROW2 is penalized by 50 units.

Similar mechanisms exist to specify purely quadratic penalties (as used in robust optimization; see [8]) and the piecewise linear-quadratic penalties of [5].

CHANCE section for chance-constrained problems

There are individual (one-dimensional) and joint (multidimensional) chance constraints, with slightly differing syntax. CC1 is a placeholder for this particular set of constraints.

Example

```
*23456789 123456789 123456789 123456789
CHANCE          INDIV
  G CC1         ROW1          0.95
  L CC1         ROW2          0.10
CHANCE          JOINT
 JG CC1                       0.98
    ROW3
    ROW4
 JL CC1                       0.001
    ROW5
    ROW6
    ROW7
```

Let us assume that all rows mentioned above (ROW1–ROW7) are set up as G-type rows in the core file, with the algebraic equivalents

$$\sum_j a_{ij}x_{ij} \geq b_i, \quad i = 1, \ldots, 7.$$

Then the four chance constraints defined by this construct are

$$\Pr\left(\sum_j a_{1j}x_{1j} \geq b_1\right) \geq 0.95,$$

$$\Pr\left(\sum_j a_{2j}x_{2j} \geq b_2\right) \leq 0.10,$$

$$\Pr\left(\sum_j a_{3j}x_{3j} \geq b_3 \wedge \sum_j a_{4j}x_{4j} \geq b_4\right) \geq 0.98,$$

$$\Pr\left(\sum_j a_{5j}x_{5j} \geq b_5 \wedge \sum_j a_{6j}x_{6j} \geq b_6 \wedge \sum_j a_{7j}x_{7j} \geq b_7\right) \leq 0.001.$$

ICC section for integrated chance constraints

This type of constraint was introduced by Klein Haneveld [6]. Assume that a deterministic version of the problem features the constraint

$$\sum_j a_j x_j \leq b.$$

Then an integrated chance constraint is of the form

$$\mathrm{E}\left(\sum_j a_j x_j \,\middle|\, \sum_j a_j x_j > b\right) \Delta\, d,$$

where Δ is an arbitrary relation ($\geq$, $\leq$, or $=$). The format is very similar to that of the individual chance constraints. (Details can be found in [2].)

SCENARIOS section

This section describes an explicit event tree scenario by scenario. There is a unique node in the event tree that belongs to the first period. This node is sometimes called the *root node*. Scenarios are identified with the leaf nodes of the tree. One could think of the scenarios as paths from the root node to the leaves, but it is easier to deal with them in the following simplified manner. One scenario (the root scenario) represents a path from the root node to one of the leaves. All other scenarios branch from a parent scenario at some time before the final period. Information up to the branch period is shared between a scenario and its parent scenario; only after the branch occurred will there be duplicate information. This point of view is advantageous because it allows for a reduction of redundancy in the tree, which compresses the size of the stoch file.

Each scenario is identified by its name, the name of its predecessor in the event tree, the period in which the branch occurred, and the probability of seeing this particular scenario. Any data items not specifically mentioned in the stoch file are inherited by each scenario from its parent scenario, which in turn may inherit them from its parent scenario, and so on. The base scenario inherits its data from the core file.

Example

```
*23456789 123456789 123456789 123456789 123456789
SCENARIOS
 SC SCEN_A     'ROOT'     0.3            PERIOD1
```

```
    COL2        ROW3        1.0
UP  BOUND1      COL5        1.0
SC  SCEN_B      SCEN_A      0.2             PERIOD3
UP  BOUND1      COL5        2.0
SC  SCEN_C      SCEN_A      0.1             PERIOD2
    COL2        ROW3        2.0
UP  BOUND1      COL5        2.0
```

This is a description of the distribution of two stochastic elements: the matrix coefficient in position COL2/ROW3 and the upper bound on variable COL5, which we assume to occur in stages 2 and 3, respectively. In scenario *A* both elements take a value of 1.0, while in scenario *B* the upper bound on COL5 is 2.0. (Both scenarios share the information in stage 2, including the coefficient in position COL2/ROW3, as well as the optimal decisions.) In scenario *C* both elements have value 2.0. Scenario A happens with probability 0.3, scenario B with 0.2, and C with 0.1. Since these add to less than one, there is at least one more scenario in this tree.

NODES section

This section describes the construction of an event tree one node at a time, which is very convenient when the depth of the tree is not uniform (containing so-called trap states) or when the number of rows or columns associated with a particular period depends on the history of the data process. It is possible to treat such stochastic problem dimensions in a scenario-wise description by introducing a number of empty rows and columns, but these phantom elements (phantom rows, phantom columns, and phantom nodes) introduce unnecessary overhead into the problem formulation. The NODES section provides an alternate way to describe the event tree, without the use of phantom elements.

For an example of how the NODES section can be used, refer to Gassmann and Schweitzer [3] or the SMPS web page [2].

INDEP section

This section describes the modeling of independent random variables. The SMPS format allows for three different forms of the distribution: discretely distributed, possessing a special distribution with no more than two parameters, or accessing a user-defined subroutine. The header record distinguishes among these forms by placing appropriate keywords into the second name field (see table below). The third keyword may contain the value ADD, MULTIPLY, or REPLACE to indicate how the value of the random variable is to modify the information found in the core file. The default is REPLACE.

The two-parameter distributions in Table 2.1 are supported.

Example 1

```
*23456789 123456789 123456789 123456789 123456789 123456789
INDEP           DISCRETE
    COL1        ROW8        6.0             PERIOD2     0.5
    COL1        ROW8        8.0             PERIOD2     0.5
```

Table 2.1. *Types of random variables.*

Keyword in second name field of header record	Distribution	First nnmeric field	Second numeric field
DISCRETE	Discrete distribution	Value	Probability*
UNIFORM	Continuous uniform on $[a, b]$	a	b
NORMAL	Normal distribution	mean μ	variance σ^2
GAMMA	Gamma distribution (on $[0, \infty)$) $f(x) = \frac{1}{b\Gamma(c)}(\frac{x}{b})^{c-1}e^{x/b}$	scale b	shape c
BETA	Beta distribution (on [0, 1]) $f(x) = \frac{x^{n-1}(1-x)^{m-1}}{B(n,m)}$	n	m
LOGNORM	Lognormal distribution	$\mu = \mathrm{E}(\ln L)$	$\sigma^2 = \mathrm{Var}(\ln L)$
(any other)	User-defined subroutine	(not used)	(not used)

*Possible values for each random element must be listed in sequence, and the probabilities must sum to 1.

In this example, the entry COL1/ROW8 takes value 6.0 with probability 0.5 and 8.0 with probability 0.5.

Example 2

```
*23456789 123456789 123456789 123456789 123456789 123456789
INDEP          DISCRETE                    ADD
    COL1       ROW8        -1.0            PERIOD2      0.5
    COL1       ROW8         1.0            PERIOD2      0.5
```

Assuming that the core file contains a value of 7.0 for the coefficient COL1/ROW8, the entry COL1/ROW8 will again take value 6.0 with probability 0.5 and 8.0 with probability 0.5.

Example 3

```
*23456789 123456789 123456789 123456789 123456789 123456789
INDEP          UNIFORM
    COL1       ROW8         8.0            PERIOD2      9.0
INDEP          NORMAL
    RHS        ROW9        10.0            PERIOD2      4.0
```

In this example the random entry COL1/ROW8 is uniformly distributed over the interval [8.0, 9.0], and the right-hand side in ROW9 is normally distributed with mean 10.0 and variance 4.0.

The SMPS format provides for a mechanism by which the user can generate the realizations by accessing a user-defined subroutine. Any modifier on the header record different from the keywords for univariate or multivariate distributions, that is, other than DISCRETE, UNIFORM, NORMAL, GAMMA, BETA, LOGNORM, MVNORMAL, LINTRAN, is taken to be the name of a subroutine, which must be provided and distributed by the user.

BLOCKS section

This section allows the user to set up multidimensional random vectors that may have correlated components but are independent of other random elements in the problem. The SMPS format allows for discrete distributions, multivariate normal distributions, and user-defined subroutines.

Example

```
*23456789 123456789 123456789 123456789
BLOCKS          DISCRETE
 BL BLOCK1      PERIOD2     0.5
    COL1        ROW6       83.0
    COL2        ROW8        1.2
 BL BLOCK1      PERIOD2     0.2
    COL2        ROW8        1.3
 BL BLOCK1      PERIOD2     0.2
    COL1        ROW6       84.0
 BL BLOCK1      PERIOD2     0.1
    COL1        ROW6       84.0
    COL2        ROW8        0.0
```

In this example, the two matrix coefficients ROW6/COL1 and ROW8/COL2 take values of (83, 1.2), (83, 1.3), (84, 1.2), and (84, 0) with respective probability 0.5, 0.2, 0.2, and 0.1. (Note that the second and third realizations of BLOCK1 mention only one of the two elements; the other element's value is inherited from the first realization.)

Sometimes the user may want to specify stochastic coefficients determined by underlying random phenomena (factors) of lower dimension. A simple form of this relationship is an affine transformation, which can be written as $v = Hu + c$, where the matrix H and the vector c are deterministic and u is a random vector. This device is also available in SMPS. For further details as well as examples, consult Gassmann and Schweitzer [3] or the SMPS web page [2].

DISTRIB section

This section describes the use of random variables (univariate and multivariate) that do not correspond directly to stochastic elements within the current problem. This device is needed to adequately model the linear transformations of the last section, since the MPS data record does not allow sufficient fields to specify correlations between random elements. The main purpose of a DISTRIB section is to provide an alias (a single name) by which the random variable can be referenced in a later BLOCKS section.

There may be many DISTRIB sections; it is assumed that each of them defines random variables independently of the others. Conversely, each new group of random variables must be introduced by a new DISTRIB header line.

Example

```
*23456789 123456789 123456789 123456789 123456789 123456789
DISTRIB       MVNORMAL
 BL BL1
    U1                    0.0                         1.0
    U2                    0.0                         1.0
 CV
    U1          U2        0.5
```

This example defines a bivariate normal random vector whose components have mean 0, variance 1, and covariance 0.5. The vector can be referred to as BL1, its first component as U1, and its second component as U2.

Linking constraints

Most constraints in stochastic programming are of the "replicating" kind. This means that each constraint contains variables from at most one node in each period. The constraint is thought to belong to the latest period in which it has variables and is replicated for each realization of the random elements.

In some special instances (for example, to model risk constraints in financial applications) it may be useful to link together variables belonging to different nodes in the same period. (For an example, consult [9].) The easiest way to include such linking or global constraints in a stochastic program is to assign the constraint to the first period. (This is best done using the explicit form of the time file.)

Solver capabilities and error recovery

The SMPS format is extremely flexible and can define a wide variety of problems, far exceeding the capability of any existing stochastic programming solver. It should not be assumed, therefore, that a given problem can be solved with a particular solver nor that in fact there exists any solver capable of solving the problem. However, any SMPS parser should recover gracefully when it encounters a section or construct its solver is not prepared to deal with.

Acknowledgments

This research was supported by a grant from the National Science and Engineering Research Council of Canada (NSERC). The author thanks Stein W. Wallace for many helpful comments.

Bibliography

[1] J. R. Birge, M. A. H. Dempster, H. I. Gassmann, E. A. Gunn, A. J. King, and S. W. Wallace, *A standard input format for multiperiod stochastic linear programs*, Math. Program. Soc. Committee Algorithms Newsletter, 17 (1987), pp. 1–19.

[2] H. I. GASSMANN, *The SMPS Format for Stochastic Linear Programs*, School of Business Administration, Dalhousie University, Halifax, NS, Canada, 2001; available online from http://www.mgmt.dal.ca/sba/profs/hgassmann/SMPS2.htm.

[3] H. I. GASSMANN AND E. SCHWEITZER, *A comprehensive input format for stochastic linear programs*, Ann. Oper. Res., 104 (2001), pp. 89–125.

[4] *Passing Your Model to OSL Using Mathematical Programming System (MPS) Format*, New York, 2001; available online from http://www6.software.ibm.com/sos/features/featur11.htm.

[5] A. J. KING, *An implementation of the Lagrangian finite-generation method*, in Numerical Techniques for Stochastic Optimization, Y. Ermoliev and R. J.-B. Wets, eds., Springer Ser. Comput. Math 10, Springer-Verlag, Berlin, 1988.

[6] W. K. KLEIN HANEVELD, *Duality in Stochastic Linear and Dynamic Programming*, Lecture Notes in Economics and Math. Systems 274, Springer-Verlag, Berlin, 1986.

[7] D. KLINGMAN, A. NAPIER, AND J. STUTZ, *NETGEN: A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems*, Management Sci., 20 (1974), pp. 814–821.

[8] J. M. MULVEY, R. J. VANDERBEI, AND S. A. ZENIOS, *Robust optimization of large-scale systems*, Oper. Res., 43 (1995), pp. 264–281.

[9] M. C. STEINBACH, *Tree-sparse convex programs*, Math. Methods Oper. Res., 56 (2003), pp. 347–376.

This page intentionally left blank

Chapter 3

The IBM Stochastic Programming System

*Alan J. King,** *Stephen E. Wright,*† *Gyana R. Parija,** *and Robert Entriken*‡

3.1 Introduction

IBM's stochastic programming product, Optimization Solutions and Library Stochastic Extensions (OSLSE), was developed at IBM Research's Thomas J. Watson Research Center in Yorktown Heights, New York, during 1990–2002. It is a library of subroutines that may be linked with user-written C/C++ programs to model and solve multiperiod stochastic linear programs with recourse. Features include quadratic objectives, integer variables, empirical tree generation, and a flexible nested decomposition solver. A parallel version of the nested decomposition solver is also available.

The current version (version 3) has been extensively revised from its initial 1998 release. It now uses the OSL version 3 C/C++ infrastructure for problem data management and solver utilities. OSLSE may be freely downloaded with a 60-day try-and-buy license from the OSL website, http://www-3.ibm.com/software/data/bi/osl/index.html. Free academic licenses are available for students and academic researchers.

3.2 A brief history

As with most scientific software projects, the development of the IBM stochastic programming system was conditioned by the needs of its customers. The major influences were a request from the Frank Russell Company to establish the solvability of industrial scale

*IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 (kingaj@us.ibm.com, parija@us.ibm.com).

†Miami University, Oxford, OH 45056 (wrightse@muohio.edu).

‡Stanford University, Palo Alto, CA 94305 (entriken@stanford.edu).

stochastic programs using a simplex-algorithm–based approach, and an extended consultation with the Allstate Insurance Corporation's Research Center in Menlo Park, California, on a strategic asset allocation system.

3.2.1 The Frank Russell problem

IBM Research's stochastic programming project originated with a request in late 1989 from Roger Wets and Bill Ziemba, consultants with the Frank Russell Company, to explore the solvability of a multiperiod stochastic program arising from an insurance asset-liability management problem [3]. The stochastic program arrived in the form of three large files in a version of the SMPS format [2] (see Chapter 2 of this volume). It had 10 periods and 2048 scenarios and was well beyond the capabilities of any stochastic programming software in existence at that time.

Bob Entriken, then a postdoc at IBM Research, wrote an SMPS reader and matrix generator for the Frank Russell problem based on the node-arc metaphors from his thesis [5]. He also suggested a simple solution idea: to solve the first 75 scenarios, transfer the first-stage solution to a larger 250-scenario problem, and so bootstrap our way to solving the full 2048-scenario problem. Alan King implemented this idea on an IBM vector mainframe. The total runtime was around 26 hours, which was mostly spent inverting the enormous matrices. This rather crude approach did demonstrate that a simplex method could be adapted to reliably solve stochastic programs of an industrial size, and Frank Russell went on to develop the world's first commercial asset-liability management product.

3.2.2 Allstate asset allocation project

Allstate Insurance, like many sophisticated firms in the financial industry, analyzes statistics of returns from its investments and losses from its insurance lines of business to develop insight into its risk management and investment activities. During 1991–1993, Allstate and IBM collaborated on the design and implementation of an asset-liability management solution for the allocation of investment surplus into asset categories. The members of the Allstate team were Tom Ward and Bill Love, and those of the IBM team were Mike Haydock, Nancy Soderquist, Luke Smith, and Alan King.

The major components of the Allstate asset allocation solution were a simulator for stochastic asset returns and liability flows, a dynamic balance sheet to generate the multiperiod accounting measurements, and a utility function composed of combinations of (up to 26 different) piecewise-linear penalty and reward functions. The simulator could be thought of as arising from a three-factor model, although the linkages between factors and the asset/liability flows were nonlinear. There were six time periods with interperiod intervals ranging from one month to more than one year. There were 11 asset classes, most of which were bonds of various maturities and grades. Trading in any asset class was possible at the beginning of every time period. The most complex part of the dynamic balance sheet was to account for taxes of various types. Finally, the liability exposure represented Allstate's stochastic property losses from extreme weather in the southern United States. The main use of the Allstate asset allocation system was to understand the trade-off among three standard accounting quantities—net income, growth of shareholder's equity, and risk-based capital—

and one statutory quantity—the ratio of premium income to statutory surplus (an important measurement of portfolio quality in the insurance industry). The major insight sought was guidance on optimal investment allocations to the equity and fixed income maturity classes under a variety of economic outlooks.

A single run of the program consisted of specifying a set of economic scenarios for the simulation run, identifying active components of the utility function, solving several hundred parametrized stochastic linear programs (each with several hundred thousand constraints and variables), and recording features of the optimal solution along the parametric frontier. In the course of its investigation, the Allstate team ran hundreds and hundreds of such runs.

Besides the challenge of building software to efficiently process enormous amounts of stochastic programming data, the Allstate project also posed new functional requirements for IBM's nascent stochastic programming system. These were to process simulations into an empirical scenario tree and to generate a "frontier" of optimal solutions by parametric variation of the components of the utility function. In addition to those functions needed to support the processing of SMPS data, these became core components of the stochastic programming system under development at IBM Research, described in the next section.

3.2.3 First generation: SPOSL

King adopted Entriken's SMPS reader/generator, and from 1990 to 1995, with major assistance from Steve Wright during his 1991–1993 postdoc appointment, developed a general-purpose stochastic programming system with both SMPS and direct input formats and a nested L-shaped decomposition method. The elegant SPL-file object was entirely designed and written by Wright. He also designed the parallel decomposition code at Miami University during the 1993–1994 academic year and implemented it under a summer contract at IBM in 1994. Hercules Vladimirou assisted with an early implementation of a two-stage L-shaped method in 1992. During a summer student visit in 1993, Chris Donohue contributed a prototype of the code that bundles simulations into scenario trees. The program came to be called SPOSL [11] and was distributed under license to roughly a dozen university students and professors.

3.2.4 Product release: OSLSE

IBM released the stochastic programming system in 1998 as part of the Optimization Solutions and Library family of products [7], calling it the OSL Stochastic Extensions (OSLSE). Gyana Parija was the lead developer supporting the initial product release. He translated the software to new OSL C/C++ interfaces, and on joining IBM Research in 2000, he undertook the testing and implemention of integer programming techniques for stochastic programs.

3.3 Stochastic input formats

OSLSE supports two types of input formats: *SMPS* and *internal arrays*. The underlying optimization problem can be linear, convex (or concave) quadratic, and/or mixed integer. The distribution types supported are *independent discrete*, *block discrete*, and *scenarios*. In addition, OSLSE supplies a *tree generation* utility that bundles simulations into scenarios.

3.3.1 SMPS input

The SMPS format was created chiefly to support the creation of problem test sets. (See Chapter 2 of this volume for a definition.) It describes a multiperiod stochastic program using specially structured files. The Core file contains an MPS description of a linear program. The Time file indicates how the variables and constraints of the linear program may be divided into time stages. The Stoch file describes the probability distributions of the elements of the Core linear program that are random in the stochastic program. In this section we outline special features or idiosyncracies of the OSLSE implementation of the SMPS standard.

The Core file. Integer variables may be indicated by using delimiters in the Core file in the manner described in [9]. OSLSE automatically propagates these throughout the stochastic program, so that an integer variable in the second stage of the Core file (say) will result in the generation of an integer variable for each second-stage node in the stochastic program. OSLSE supports free-format input of Core files.

The Time file. OSLSE supports only the `IMPLICIT` keyword in the PERIODS statement. This means that rows and columns in the core file must be sorted in period order. A future release will support the `EXPLICIT` keyword (the direct interface already supports it).

The Stoch file. The Stoch file sections supported by OSLSE are `SCENARIOS`, `INDEP DISCRETE`, and `BLOCK DISCRETE`. There must be only one section type in a given Stoch file. ADD and REPLACE options for all these sections are allowed. The internal data structures in OSLSE allow the Stoch file to contain matrix data elements that are not defined in the Core file.

Infinite core bound warning. An idiosyncracy of OSLSE arises from the design of its internal data structures where stochastic data are stored as a scenario tree whose values *add* to Core values. This design has certain advantages but one important drawback: if a Stoch file element replaces or adds to an infinite bound in the Core file, then a warning is printed and the stochastic element is ignored.

3.3.2 Input via internal arrays

Processing SMPS files can be slow, and writing SMPS files can be exacting. Our experience is that most developers will want to write stochastic programming applications in a programming language such as C/C++ or in a modeling language such as GAMS or AMPL. For this reason OSLSE provides user interfaces for the input of stochastic program data. Calling sequences for these functions are described in the appendices. A simple driver showing their use is described in Appendix 3.A.

Core and time data. The function `ekks_createCore()` passes the linear programming data and time stage information. Time stage information is passed explicitly. All indexing in OSLSE is Fortran style, so the lowest row, column, or time-stage index is 1. OSLSE supports quadratic and mixed-integer core problems. One may pass nonnegative diagonal quadratic matrix entries and set the core problem type to quadratic to invoke a quadratic solver. Integer variables may be declared by a call to `ekks_setIntegersAtCore()`.

Stoch data. The function `ekks_addScenario()` passes stochastic data in the *scenarios* data format. All indexing in OSLSE is Fortran-style, so the lowest scenario index is 1. A repeat call to `ekks_addScenario()` with the same scenario index merely adds to the probability weight for that scenario. The number of rows and columns in the scenario data must be identical to their counterparts in the core data. The value of the variable `replace` indicates whether the stochastic data adds to or replaces the core data. Future implementations will support functions `ekks_addIndep()` and `ekks_addBlock()` to allow one to pass independent discrete distributions directly to OSLSE.

3.3.3 Generating a scenario tree by bundling simulations

OSLSE includes an independent set of routines for the purpose of building a scenario tree out of a sequence of simulations. The basic idea is to implicitly generate a subfiltration from a finite partition of the support of the random variables and use simulation to assign empirical weights to the nodes of the subfiltration.

For a concrete example, suppose each member of a sequence of random variables $\{X_t\}_{t=1}^T$ has a sample space that is a subset of $\mathbf{R}^d$. Partitioning each coordinate into (say) five intervals generates a finite subfiltration with 5^{dT} possible paths. Simulations can be used to build an empirical scenario tree as follows. Each simulation $\{\xi_t\}_{t=1}^T$ is mapped to a sequence of *events*, $\{e_t\}_{t=1}^T$, which are integers from $1, \ldots, 5^d$ recording the label of the d-rectangle that ξ_t lies in. When a new simulation is added (by calling `ekks_addEventToTree()`), its initial event sequence is compared against all event sequences so far received, and the longest match found is recorded as the parent of the incoming event sequence. Weights of the unmatched nodes in the incoming sequence are initialized at one, and the weights of the matched nodes of the parent sequence are incremented by one.

The result is a scenario tree that describes the empirical distribution of the sequence of simulations relative to the event subfiltration. The event data and scenario numbers, branching node, and probability weights can be retrieved by `ekks_getScenarioFromTree()`. To pass the scenario to the stochastic program, one maps the event index to scenario data values (say, to the centroid of the d-rectangle or the expected value conditioned on the d-rectangle) and calls `ekks_addScenario()`.

Simulation relies on the laws of large numbers to reduce the effort needed to approximate aspects of the solution. Unfortunately, the dimension of a stochastic programming solution is so large that one cannot expect laws of large numbers effects in any stage but the first. In the Allstate case, experiments were performed to determine the effectiveness of this kind of sampling of scenario trees. It was found that the error distributions of the first-stage solution vector (considered one dimension at a time) did appear to show central limit properties: errors were distributed normally about a central value with a variance that

was decreasing linearly in the number of simulations.

3.4 The decomposition solver

OSLSE implements a flexible nested decomposition solver to take advantage of the structural properties of stochastic programs. It is difficult to discuss the solver without reference to a presentation of the method, so for completeness we give a brief overview here.

3.4.1 The L-shaped method

Consider an L-shaped linear program

$$\begin{array}{lll} \min\limits_{(x_1,x_2)} & c_1x_1 + c_2x_2 & \\ \text{s.t.} & A_{11}x_1 & \in [y_1^-, y_1^+], \\ & A_{21}x_1 + A_{22}x_2 & \in [y_2^-, y_2^+], \\ & x_1 & \in [x_1^-, x_1^+], \\ & x_2 & \in [x_2^-, x_2^+]. \end{array} \tag{3.1}$$

Form the *parent subproblem* by minimizing out the second set of variables,

$$\begin{array}{lll} \min\limits_{(x_1,\theta_1)} & c_1x_1 + \theta_1 & \\ & A_{11}x_1 & \in [y_1^-, y_1^+], \\ \text{s.t.} & x_1 & \in [x_1^-, x_1^+], \\ & \theta_1 & \geq f_2(x_1). \end{array} \tag{3.2}$$

The impact of the second set of variables is represented abstractly in the parent by the constraint involving the value of the *child subproblem*,

$$f_2(\bar{x}_1) = \left\{ \begin{array}{lll} \min\limits_{x_1,x_2} & c_2x_2 & \\ & x_1 & = \bar{x}_1, \\ \text{s.t.} & A_{21}x_1 + A_{22}x_2 & \in [y_2^-, y_2^+], \\ & x_2 & \in [x_2^-, x_2^+]. \end{array} \right. \tag{3.3}$$

The L-shaped method algorithm iteratively approximates the abstract constraint by computing *optimality cuts* and *feasibility cuts* from the optimality conditions of the child subproblem for successive parent proposals. (In passing the child problem 3.3 to OSL, it turns out to be convenient to make the constraints $x_1 = \bar{x}_1$ explicit. The cuts then turn out to be the reduced costs associated with this set of constraints. OSL will in any case substitute the fixed values $\bar{x}_1$ where required.)

The parent subproblem after the kth step of the method has the form

$$\begin{array}{lll} \min\limits_{(x_1,\theta_1)} & c_1x_1 + \theta_1 & \\ & A_{11}x_1 & \in [y_1^-, y_1^+], \\ & x_1 & \in [x_1^-, x_1^+], \\ \text{s.t.} & \theta_1 e + F^k x_1 & \geq f^k, \\ & G^k x_1 & \geq g^k, \end{array} \tag{3.4}$$

where e is the vector of ones with dimension equal to the number of rows in the matrix F^k. The rows of the matrices F^k and G^k and the right-hand-side vectors f^k and g^k are formed by the collection of optimality and infeasibility cuts, respectively. Van Slyke and Wets [15] showed that the following algorithm terminates in a finite number of steps.

Decomposition algorithm

0. Solve (3.4).

1. If infeasible, original problem (3.1) is infeasible. STOP.
2. If unbounded, continue with ray $\bar{x}_1^k + t\bar{u}_1^k, t \geq 0$.
3. If optimal, continue with solution $(\bar{x}_1^k, \bar{\theta}_1^k)$.

1. Solve the child subproblem (3.3) with parent proposal $\bar{x}_1^k$.

1. If unbounded, original problem is unbounded. STOP.
2. If infeasible, generate infeasibility cut from the sum of infeasibilities $f_2^I(\bar{x}_1^k)$ and reduced costs $\bar{\delta}_1^k$:
$$-\bar{\delta}_1^k x_1 \geq f_2^I(\bar{x}_1^k) - \bar{\delta}_1^k \bar{x}_1^k. \tag{3.5}$$
Add coefficients as a new row of matrix G^k and element of right-hand-side g^k. Return to step 0 with $k \mapsto k+1$.
3. If optimal with optimal value $f_2(\bar{x}_1^k)$ and parent was bounded, go to the Optimality test. Otherwise go to the Unboundedness test.

Optimality test. Test the optimality gap $f_2(\bar{x}_1^k) - \bar{\theta}_1^k$.

1. If it is within tolerance, optimal solution is found. STOP.
2. Otherwise, generate optimality cut from the reduced costs $\bar{\delta}_1^k$:
$$\theta_1 - \bar{\delta}_1^k x_1 \geq f_2(\bar{x}_1^k) - \bar{\delta}_1^k \bar{x}_1^k. \tag{3.6}$$
Add the coefficients as a new row of matrix F^k and element of right-hand-side f^k. Go to step 0 with $k \mapsto k+1$.

Unboundedness test. Form the child recession subproblem from child subproblem (3.3). (Set all finite bounds to zero, all positive column infinite bounds to 1, and all negative infinite column bounds to -1.) Solve with the proposal $\bar{u}_1^k$.

1. If infeasible, generate infeasibility cut and add coefficients to matrix G^k and right-hand-side g^k. Go to step 0 with $k \mapsto k+1$.
2. If optimal with solution $\bar{u}_2^k$ and $c_1\bar{u}_1^k + c_2\bar{u}_2^k \geq 0$, generate optimality cut and add coefficients to matrix F^k and right-hand-side f^k. Go to step 0 with $k \mapsto k+1$.
3. Otherwise problem is unbounded. STOP.

3.4.2 The decomposition solver in OSLSE

The solver `ekkse_nestedLSolve()` in OSLSE implements a flexible, nested L-shaped method [12]. The implementation is *flexible* in the sense that subproblems may contain any number of nodes from the scenario tree. The implementation is *nested* in the sense that the L-shaped method may be applied to child subproblems.

Subproblem generation. Subproblems are automatically generated in one of three ways. One, specify that the scenario tree be cut at an arbitrary time stage (`CUTATSTAGE`). Two, specify that no subproblem contain more than a given number of rows (`CUTBYROWSIZE`). Three, specify that there be no more than a fixed number of subproblems (`CUTBYMAXNODES`). Each way of generating subproblems has advantages. Decomposition methods tend to be slower than direct solvers for small problems, so it may make sense to generate subproblems of a size that is efficient for the underlying solver. Specifying the number of rows for each subproblem (`CUTBYROWSIZE`) controls the size of the subproblems directly. Specifying the number of subproblems (`CUTBYMAXNODES`) may affect the speed of convergence of the decomposition algorithm by reducing the communication overhead. But, however desirable such properties may seem, the advantages could be undercut by having subproblems that are unbounded. OSLSE applies the "augmented root" construction when cutting at a time stage (`CUTATSTAGE`), in which the root subproblem is combined with the first leaf subproblem. This may bound the root subproblem and so speed up the convergence.

Solver directives. The decomposition solver uses the OSL simplex solvers. The user may specify directives to the LP solvers, such as scaling or the use of a crash or presolve. If the problem type is *quadratic*, then OSL's quadratic solver will be invoked. A future implementation of OSLSE will allow advanced users to provide their own solver through the OSI solver interface [4].

Termination. The decomposition solver terminates if unboundedness or infeasibility has been detected, when the optimality test succeeds, when the maximum number of root iterations has been exceeded, or when no new cuts were generated for the incumbent proposal from the root subproblem. In the latter case, the progress of the algorithm has stalled with an optimality gap that is larger than desired by the user. In such a case the algorithm indicates that an optimal solution has been found but reports the actual optimality gap attained. The user can specify that the decomposition solver be followed by a simplex solver. This problem will be hot-started from the solution found by the decomposition solver.

Access to solutions. The user can retrieve or print primal and dual solutions by node and scenario and also can print the distribution of objective function values. The solution vectors are in the order determined by the core data.

3.4.3 The parallel decomposition solver

OSLSE supports the running of the decomposition solver in parallel environments. This feature is automatically invoked by setting the parallel switch on `ekkse_setParallelOn()`

before calling the decomposition solver. The parallel solver detects how many machines are in the computing environment and invokes a solver process called `OSLSE_Parallel` on each one. The subproblems are then divided among the solver processes and initialized. Each solver process loops over the subproblems assigned to it and performs whatever actions are required by the decomposition algorithm. Cuts, proposals, and subproblem status are passed using a message-passing infrastructure.

Currently the PVM [13] and MPI [1] message-passing environments are supported. IBM Research is also considering implementing OSLSE's parallel solver in a grid computing environment [6].

3.5 Stochastic mixed-integer problems

OSLSE supports a branch-and-bound algorithm for the solution of stochastic mixed-integer programs [14]. Integer variables are declared either through the Core file or through the function `ekks_setIntegersAtCore()`. The type of the integer may be specified as binary, general integer, or SOS2. Once the integers are specified, the function `ekks_markIntegers()` propagates the integer variables through the stochastic program following the convention that the Core problem specifies a template for the stochastic program. This will mark as integer every stochastic variable corresponding to a core variable marked as integer. (The user may override the markings by accessing the stochastic program as an EKKModel object and using OSL methods to change the integer variable settings.) The function `ekks_markIntegersWithStagePriorities()` influences the selection of branching variables. Finally the function `ekks_branchAndBound()` applies OSL's branch-and-bound mixed-integer programming algorithm to the resulting stochastic program.

3.6 Miscellaneous special features of OSLSE

OSLSE is a library designed much like its parent OSL. All data are stored privately in a `Stoch` object. Public modules set/get parameters and data and perform various processes on the `Stoch` object. The basic uses of OSLSE are described in the public documentation [8]. In this section we highlight a few special features of OSLSE.

SPL file. OSLSE stores the `Stoch` object in a binary formatted SPL file, which can be used to separate data input from repeated solution trials. At the conclusion of data input one can save the stochastic program to a permanent SPL file by calling `ekks_writeNativeData()`. The solver part may then begin by calling `ekks_readNativeData()`. This is an extremely efficient way to regenerate the `Stoch` object because data blocks in the SPL file can be copied directly to solver memory regions. The SPL file is a compact scenario description of the stochastic program and so will be much smaller than the SMPS description using scenario format.

SMPS and MPS file writers. It is sometimes useful to be able to debug stochastic programming data input logic by examining the SMPS files or even the MPS files for the deter-

ministic equivalent LP. OSLSE includes functions `ekks_outMatrixSMPS` to write out the data in SMPS scenarios format and `ekks_outMatrixMPS` to write out the MPS file.

Note added in proof. The product family OSL, including OSLSE, is no longer available from IBM as of March, 2004. The authors invite all readers to join in the open source project Stochastic Modeling Interface (SMI) hosted by the COIN-OR initiative at www.coin-or.org. We are in the process of integrating stochastic programming functionality in the COIN optimization framework, but we were not able to provide details in time for this publication.

Appendix 3.A Sample input with internal arrays

This section illustrates the use of internal arrays to pass problem data on the `KandW` example, which is a modified version of a production planning problem from the text [10]. A refinery can blend two raw materials into two different products. They must decide how much raw material to purchase and stock so that they can be blended to satisfy the demand for the products in two future periods. The demand has to be completely satisfied, and in case of raw material shortage the products can be outsourced at a higher cost. There is an inventory constraint on how much raw material can be stocked in total. The algebraic statement of the problem is

$$\begin{aligned}
\min_{x,y} \quad & \sum_{t=1}^{3}\sum_{i=1}^{2} c_{it}x_{it} + E\sum_{t=2}^{3}\sum_{j=1}^{2} f_{jt}y_{jt} \\
\text{s.t.} \quad & \sum_{t=1}^{3}\sum_{i=1}^{2} x_{it} \qquad\qquad \leq b, \\
& \sum_{i=1}^{2} a_{ij}x_{it} + y_{jt} \quad \geq \tilde{d}_{jt}, \qquad t = 2,3, \quad j = 1,2.
\end{aligned}$$

The sample problem driver for the data as given in the Samples section of the OSLSE guide and reference [8] is as follows:

```
/* Core Model */
int ncol=8, nrow=5, nels=16;
/* Sparse matrix data */
int    mcol[]={1, 2, 3, 4, 1, 2, 3, 4, 1, 2,   3, 4,   5,6,7,8};
int    mrow[]={1, 1, 1, 1, 2, 2, 4, 4, 3, 3,   5, 5,   2,3,4,5};
double dels[]={1, 1, 1, 1, 2, 6, 2, 6, 3, 3.4, 3, 3.4, 1,1,1,1};
/* Objective */
double dobj[]={ 2.0, 3.0, 2.0, 3.0, 7.0, 12.0, 10.0, 15.0 };
/* Column bounds */
double dclo[]={ 0.0, 0.0, 0.0, 0.0, 0.0,  0.0,  0.0,  0.0 };
double dcup[]={ INF, INF, INF, INF, INF,  INF,  INF,  INF };
/* Row bounds */
double drlo[]={ -INF, 0.0, 0.0, 0.0, 0.0 };
double drup[]={ 50.0, INF, INF, INF, INF };
/* Stages */
int nstg=3;
int rstg[]={ 1,2,2,3,3 };
```

```
int cstg[]={ 1,1,1,1,2,2,3,3 };
/* Scenarios */
int nscen=9;
double dp[]={ .06, .15, .09, .12, .16, .12, .12, .12, .06 };
/* local variables */
int rc,is,ia,ib,ii;
EKKContext *env = ekks_initializeContext();
EKKStoch *stoch=ekks_newStoch(env,"KandWs",nscen);
/* Create Core */
rc=ekks_createCore(stoch,nstg,rstg,cstg,nrow,ncol,nels,
        dobj,drlo,drup,dclo,dcup,mrow,mcol,dels,0 );
/* Generate Scenarios using REPLACE_CORE_VALUES */
drlo[1]=200; drlo[2]=180; drlo[3]=200; drlo[4]=180;
for(is=0; is<nscen; is++){
    ia=0; ib=2;
    rc=ekks_addScenario(stoch,is+1,ia,ib,dp[is],1,
            nrow,ncol,nels,dobj,drlo,drup,dclo,dcup,
            mrow,mcol,dels,REPLACE_CORE_VALUES );
    ia=is+1; ib=3;
    for( ii=is+1; ii<is+3; ii++){
        drlo[3] -= 20; drlo[4] -= 20;
        rc=ekks_addScenario(stoch,ii+1,ia,ib,dp[ii],1,
                nrow,ncol,nels,dobj,drlo,drup,dclo,dcup,
                mrow,mcol,dels,REPLACE_CORE_VALUES );
    }
    drlo[3]=200; drlo[4]=180; drlo[1] -= 20; drlo[2] -= 20; is+=2;
}
/* Solve and test optimal value */
rc=ekks_describeFullModel(stoch,1);
rc=ekkse_nestedLSolve(stoch,CUTATSTAGE,0);
assert(fabs(ekk_getRobjvalue(ekkse_getCurrentModel(stoch))-2613.000)<0.001);
ekks_deleteStoch(stoch);
ekks_endContext(env);
```

Appendix 3.B OSLSE interfaces for passing problem data

3.B.1 Core and time data

```
/* Pass Core model using in-core-memory data */
int ekks_createCore(EKKStoch *stoch,
        int nstg, int *rstag, int *cstag, /* stage info                */
        int irow, int icol, int iels,     /* # rows, cols, elements    */
        double *dobj,                     /* objective                 */
        double *drlo, double *drup,       /* row bounds                */
        double *dclo, double *dcup,       /* col bounds                */
        int *mr, int *mc, double *dels,   /* matrix triples            */
        double *dqdg);                    /* diag quadratic terms      */

/* Set Core problem  type to LP or QP */
typedef enum {linear=1, quadratic=2} CoreType;
int ekks_setCoreProblemType(EKKStoch *stoch, CoreType type);
```

3.B.2 Stochastic data

```
/* Define Scenario using in-core-memory data */
```

```
int ekks_addScenario(EKKStoch *stoch,
        int nscn,                          /* scenario number          */
        int ianc,                          /* # of ancestor scenario   */
        int istg,                          /* branching stage          */
        double dprob,                      /* probability              */
        int itype,                         /* matrix type (not used)   */
        int nrow, int ncol, int nels,
        double *dobj,
        double *drlo, double *drup,
        double *dclo, double *dcup,
        int *mrow, int  *mcol, double *dels,
        int replace );                     /* replace = ADD_TO_CORE_VALUES
                                             or        REPLACE_CORE_VALUES */
/* Reset probability weights */
int ekks_changeProbability(EKKStoch *stoch, double *dp);

/* Replace tree with samples from original scenario tree                  */
/* samplemode =0 - uniform random sampling (the only supported mode)      */
int ekks_generateSamples(EKKStoch *stoch, int numsamples, int samplemode);
```

3.B.3 Generating scenario tree from simulations

```
/* Initialize EKKTree structure for a maximum of mxsmp scenarios          */
/* and create first event from the array of data values                   */
/* arr[] with narr entries.  The length of the arr[] array, narr,         */
/* is the maximum number of branch nodes between root and leaf.           */
EKKTree * ekks_createScenarioTree(int *arr, int narr, int mxsmp, double dp);

/* Add event to EKKTree object. Finds the event anc_arr[] with longest */
/* sequence max{k: arr[0] = anc_arr[0], ... , arr[k] = anc_arr[k]},    */
/* and adds the branch (arr[k+1] ... arr[narr-1]).                     */
int  ekks_addEventToTree(EKKTree *tree,int *arr, int narr, double dp);

/* Get the scenario branching information for event number nsmp.       */
/* The meaning of the returned data is the same as in ekks_addScenario */
int ekks_getScenarioFromTree(EKKTree *tree, int nsmp,
        int *ianc, int *istg, double *dprob, int *nscn,
        int **arr);                        /* points to array values */

/* Single function versions of ekks_getScenarioFromTree() */
int ekks_getScenarioAncestorFromTree(EKKTree *tree,int nsmp);
int ekks_getScenarioBranchStageFromTree(EKKTree *tree,int nsmp);
double ekks_getScenarioWeightFromTree(EKKTree *tree,int nsmp);
int ekks_getScenarioNumberFromTree(EKKTree *tree,int nsmp);
int ekks_getDataLengthFromTree(EKKTree *tree);
int* ekks_getScenarioDataFromTree(EKKTree  *tree,int nsmp);
```

Appendix 3.C Reading and writing SPL, SMPS, and MPS files

```
/* Read or write using OSLSE Native Data model (very fast) */
int ekks_readNativeData(EKKStoch *stoch, char *splfilename);
int ekks_writeNativeData(EKKStoch *stoch, char *splfilename);
```

```
/* read SMPS data and return type of distribution in stoch file */
int ekks_readSMPSData(EKKStoch *stoch,
        const char *corefile, const char *timefile, const char *stochfile);
/* Generate SMPS files --- useful for debugging! */
int ekks_outMatrixSMPS(EKKStoch *stoch,
       int smpstype, /* SCENARIOS, BLOCK_DISCRETE, INDEP_DISCRETE */
       int  replace,  /* ADD_TO_CORE_VALUES, REPLACE_CORE_VALUES */
       const char *corefile, const char *timefile, const char *stochfile);
/* Generate MPS file */
int ekks_outMatrixMPS(EKKStoch *stoch, const char *mpsfile);
```

Appendix 3.D OSLSE interfaces for the decomposition solver

```
/* The Nested L-shaped Method solver decomposes using the cutstrategy:  */
#define CUTATSTAGE      1    /* equivalent to ekks_bendersLSolve    */
#define CUTBYROWSIZE    2    /* sets max # rows in subproblems      */
#define CUTBYMAXNODES   3    /* sets max # of subproblems           */

int ekkse_nestedLSolve(EKKStoch *stoch, int cutstrategy, int startmode);

/* Get/set cut period for cut-at-stage decomposition */
int   ekks_getCutPeriod(EKKStoch *stoch);
void   ekks_setCutPeriod(EKKStoch *stoch, int period);

/* Get/set number of rows for cut-by-row-size decomposition */
int  ekks_getMinNumRows(EKKStoch *stoch);
void  ekks_setMinNumRows(EKKStoch *stoch, int numrows);

/* Get/set number of models for cut-by-max-models decomposition */
int  ekks_getMaxsubmodels(EKKStoch *stoch);
void  ekks_setMaxsubmodels(EKKStoch *stoch, int maxmodels);
```

3.D.1 Solver directives

```
/* set direction of optimization */
void  ekks_setMinimize(EKKStoch *stoch);
void  ekks_setMaximize(EKKStoch *stoch);

/* Get/set maximum number of major iterations in decomposition */
int ekks_getMaxiter(EKKStoch *stoch);
void ekks_setMaxiter(EKKStoch *stoch, int maxiter);

/* Get/set gap used to assess convergence of decomposition. */
double OSLLINKAGE ekks_getRelOptimalityGap(EKKStoch *stoch);
void ekks_setRelOptimalityGap(EKKStoch *stoch, double tol);

/* Set scaling on/off for subproblems in decomposition. */
/* It is usually a good idea to set it on. */
void ekks_setScaleOn(EKKStoch *stoch);
void ekks_setScaleOff(EKKStoch *stoch);
/* Set OSL presolve type on/off for subproblems. */
void ekks_setPresolve(EKKStoch *stoch,int presolvetype);
void ekks_setPresolveOff(EKKStoch *stoch);
```

```
/* Set OSL crash type on/off for subproblems */
void ekks_setCrash(EKKStoch *stoch, int crashmode);
void ekks_setCrashOff(EKKStoch *stoch);

/* Set OSL Simplex solver type for subproblems. */
void  ekks_setSimplexAlg(EKKStoch *stoch, int alg);

/* After decomposition solve/don't solve with simplex solver */
void ekks_setFinalSimplexSolverOn(EKKStoch *stoch);
void ekks_setFinalSimplexSolverOff(EKKStoch *stoch);

/* Set large bounds on/off (may affect speed of decomposition). */
void ekks_setLargeBoundsOn(EKKStoch *stoch);
void ekks_setLargeBoundsOff(EKKStoch *stoch);
```

3.D.2 Access to solutions

```
/* Solution Status
   (0 - optimal, 1 - infeasible, 2 - unbounded, >=3 - incomplete) */
int ekks_getStatus(EKKStoch *stoch);
/* Objective Value */
double ekks_getRobjvalue(EKKStoch *stoch);

/* Getting and Printing Solutions */
/*   mode = 0 get column solution */
/*   mode = 1 get row solution    */

/* OSLSE sorts the matrix.  index[] shows the internal index number. */
int ekks_getNodeSolution(EKKStoch *stoch, int scenario, int stage,
        int mode, double *solution, int *index);
int ekks_getScenarioSolution(EKKStoch *stoch, int scenario,
        int mode, double *solution, int *index);
int ekks_getNodeDualSolution(EKKStoch *stoch, int scen, int stg,
        int mode, double *solution, int *index);
int ekks_getScenarioDualSolution(EKKStoch *stoch, int scen,
        int mode, double *solution, int *index);

/* Printing */
int ekks_printObjectiveDistribution(EKKStoch *stoch);
int ekks_printNodeSolution(EKKStoch *stoch, int scen, int stg,
         int mode);
int ekks_printNodeDualSolution(EKKStoch *stoch, int scen, int stg,
         int mode);

/* Stochastic LP dimensions */
int ekks_estimateLPSize(EKKStoch *stoch);
int ekks_getNumScenarios(EKKStoch *stoch);
int ekks_getNumStages(EKKStoch *stoch);
int ekks_getNumNodes(EKKStoch *stoch);
int ekks_getCoreNumcols(EKKStoch *stoch);
int ekks_getCoreNumrows(EKKStoch *stoch);
int ekks_getNumcolsAtStage(EKKStoch *stoch,int stg);
int ekks_getNumrowsAtStage(EKKStoch *stoch,int stg);
```

Appendix 3.E OSLSE interfaces for stochastic mixed integer problems

```
/* ****
ekks_setIntegersAtCore marks a set of columns in the core model as integers.
The integer array intType[numints] specifies the types of each integer
(1 - binary, 2 - general integer, 3 - SOS2, 0 - continuous).
**** */
int ekks_setIntegersAtCore(EKKStoch *stoch,
        int numints, int *intnums, int *intType);

/* propagate integer variables through EKKStoch and return total */
int ekks_markIntegers(EKKStoch *stoch);
/* propagate with priorities and pseudocosts */
int ekks_markIntegersWithStagePriorities(EKKStoch *stoch,
        int *stagePriority, double *upperPseudoCost, double *lowerPseudoCost);

/* solve with branch and bound */
int ekks_branchAndBound(EKKStoch *stoch,
        const char *matrixFile, const char *basisFile);
```

Bibliography

[1] Argonne National Laboratory, *MPI: The Message Passing Interface Standard*, Technical Report, Argonne, IL, 1995; available online from http://www-unix.mcs.anl.gov/mpi/index.html.

[2] J. R. Birge, M. A. H. Dempster, H. I. Gassmann, E. A. Gunn, A. J. King, and S. W. Wallace, *A standard input format for multiperiod stochastic linear programs*, Math. Program. Soc. Committee on Algorithms Newsletter, 17 (1987), pp. 1–19.

[3] D. Cariño, T. Kent, D. Meyers, C. Stacy, M. Sylvanus, A. Turner, K. Watanabe, and W. T. Ziemba, *The Russell-Yasuda Kasai model: An asset/liability model for a Japanese insurance company using multistage stochastic programming*, Interfaces, 24 (1994), pp. 29–49.

[4] *Open Solver Interface*, Technical Report, Common Optimization Interface for Operations Research (COIN-OR), 2001; available online from http://www-124.ibm.com/developerworks/opensource/coin/index.html.

[5] R. Entriken, *The Parallel Decomposition of Linear Programs*, Ph.D. thesis, Department of Operations Research, Stanford University, Stanford, CA, 1989.

[6] *The Globus Project*, Technical Report, The Globus Project, 2001; available online from http://www.globus.org/.

[7] *IBM Optimization Solutions and Library*, Technical Report, IBM, New York, 1995; available online from http://www-3.ibm.com/software/data/bi/osl/index.html.

[8] *IBM Optimization Solutions and Library: Stochastic Extensions*, Technical Report, IBM, New York, 1998; available online from http://www-3.ibm.com/software/data/bi/osl/features/stex.html.

[9] IBM, *Optimization Solutions and Library: Guide and Reference*, New York, 2001; available online from http://www6.software.ibm.com/sos/features/libuser.htm.

[10] P. KALL AND S. W. WALLACE, *Stochastic Programming*, John Wiley, Chichester, UK, 1994.

[11] A. KING, *SP/OSL Version* 1.0 *Stochastic Programming Interface Library User's Guide*, Research Report RC19757, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1994.

[12] A. J. KING AND S. E. WRIGHT, *A Flexible-Partition, Nested L-Shaped Method for Linear Programming*, preprint, IBM Research, Yorktown Heights, NY, 2002.

[13] *PVM: Parallel Virtual Machine*, Technical Report, Oak Ridge National Laboratory, Oak Ridge, TN, 1993; available online from http://www.epm.ornl.gov/pvm/.

[14] G. R. PARIJA, S. AHMED, AND A. J. KING, *On bridging the gap between stochastic integer programming and MIP solver technologies*, INFORMS J. Comput., 16 (2004), pp. 73–83.

[15] R. M. VAN SLYKE AND R. WETS, *L-shaped linear programs with applications to optimal control and stochastic programming*, SIAM J. Appl. Math., 17 (1969), pp. 638–663.

Chapter 4

SQG: Software for Solving Stochastic Programming Problems with Stochastic Quasi-Gradient Methods

*Alexei A. Gaivoronski**

4.1 Introduction

Stochastic programming problems are characterized by a variety unknown in other fields of optimization. For this reason a universal solution approach uniformly suitable for the vast majority of stochastic programming problems is unlikely to emerge. Until now the largest effort was dedicated to the development of solution techniques and software systems suitable for the solution of large-scale linear stochastic programming models with two or more time periods and a discrete set of possible values of random parameters organized in scenario trees. These systems, some of which are described in [2, 19, 25, 22, 28], proved to be very successful in their application domains; see also the separate chapters on OSLSE by King et al. (Chapter 3), SLP-IOR by Kall and Mayer (Chapter 6), and SPInE by Valente et al. (Chapter 8) in this volume. However, a considerable part of stochastic optimization problems remains outside their scope, most notably nonlinear problems, problems with continuous distributions, and problems related to optimization of simulation models.

This paper describes an implementation of a complementary set of methods called stochastic quasi-gradient methods (SQG). These methods are specifically designed having in mind continuous distributions of random parameters and nonlinear optimization problems. They are suited for optimization of simulation models where analytical relations between the objective function and parameters are difficult to trace. Another application area of such methods consists of multiperiod dynamic stochastic models with parametrized decision rules. Supply chain management, financial applications, telecommunications, and energy generation are among the sources of such problems.

The most important notion here is the stochastic gradient, which is a statistical estimate

*Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Alfred Getz vei 1, N-7491, Trondheim, Norway (alexei.gaivoronski@iot.ntnu.no).

of the gradient of the objective function. It provides a direction for iterative updating of the current approximation to the optimal solution. Another crucial component is the stepsize used for the movement along this direction. SQG methods can exploit the structure of the models by using gradient estimation schemes specific to a given application domain, while in the absence of such schemes general estimation procedures can be applied, such as the averaging of finite difference approximations; for other examples see section 4.2 and [13].

The rest of the chapter is organized as follows. Section 4.2 contains a brief description of the methodology behind SQG methods. Here we describe the general algorithmic scheme and highlight the issues that are important for implementation. In section 4.3 we describe the target application areas where SQG methods have an advantage over other stochastic programming techniques. Section 4.4 is focused on the architecture of the current implementation. Our philosophy is to use available off-the-shelf software as the building blocks and development environment both for user models and for SQG implementation. We wanted to give some meaningful examples of the use of SQG to make it easier for an interested reader to form an opinion about the system capabilities. For this reason we included sections 4.5 and 4.6, which contain examples from supply chain management and finance. We conclude with a summary.

Prospective users can get information about obtaining SQG software by writing to alexei.gaivoronski@iot.ntnu.no.

4.2 Brief description of the methodology

In this section we outline the solution approach that lies behind the SQG system, show its place in the general context of stochastic programming methods, and give an informal discussion of its various features and extensions that are important from the point of view of practical use.

SQG methods were considered first in [7]; see also [8, 12, 33, 13, 26, 35, 30]. They have their roots in stochastic approximation [31, 24] and in the mathematical programming algorithms. They solve the problem

$$\min_{x \in X} \mathbb{E}_\omega f_0(x, \omega), \tag{4.1}$$

where $\omega \in \mathbb{R}^k$ is the vector of random parameters and $x \in X \subseteq \mathbb{R}^n$ is the vector of decision variables. The set X defines the set of feasible solutions and usually is a convex set of simple structure. For example, X can be defined by upper and lower bounds on the components of the vector x or it can be defined by linear constraints $Ax \leq b$, where A is an $m \times n$ matrix. Different modifications allow consideration of the feasible set defined by convex inequalities

$$\mathbb{E}_\omega f_i(x, \omega) \leq 0, \quad i = 1 : m. \tag{4.2}$$

On this level of generality the problem (4.1)–(4.2) describes a large set of dynamic and static stochastic optimization problems. In the vast majority of the practically interesting problems the expectation operator present in (4.1)–(4.2) makes problematic the direct computation of the functions $F_i(x) = \mathbb{E}_\omega f_i(x, \omega)$. For this reason considerable effort was dedicated to the development of specialized algorithms; see [37, 10, 23, 3] for surveys.

These methods fall into two categories: deterministic equivalents and iterative sampling algorithms.

Deterministic equivalents start by approximating the problem (4.1)–(4.2) by a problem in which the original probability distribution of the random parameters ω is substituted by a discrete distribution concentrated in a finite number of points ω^i which describe different scenarios. Consequently, the original problem is replaced by the deterministic optimization problem of special structure where expectations in (4.1)–(4.2) are replaced by sums. The approximating problem is solved either by direct application of deterministic optimization algorithms or by specialized techniques which exploit the structure of the problem. Specialized approaches like Benders decomposition [36] are especially successful in the case of linear stochastic problems with recourse with two or more periods, where direct application of linear programming software can encounter difficulties due to the high dimension of the approximating problem.

Iterative sampling algorithms deal directly with stochasticity by constructing statistical estimates of functions $F_i(x)$, their gradients, and Hessians. Such estimates are obtained by generating observations or scenarios ω^i during the solution process. These estimates are used instead of exact values in the iterative algorithms that have their roots in linear or nonlinear programming. SQG methods belong to this class; another example is the stochastic decomposition of [20]. In the simplest case they start from some initial point x^0 and update the current approximation x^s to the optimal solution of problem (4.1) by making a step ρ_s in the direction opposite to the current estimate ξ^s of the gradient of $F_0(x)$ at point x^s and projecting the resulting point onto the set X:

$$x^{s+1} = \pi_X(x^s - \rho_s \xi^s). \tag{4.3}$$

Here π_X is the projection operator on X which transforms an arbitrary $z \in \mathbb{R}^n$ into the point $\pi_X(z) \in X$ such that

$$\|z - \pi_X(z)\| = \min_{x \in X} \|z - x\|. \tag{4.4}$$

Consequently, the structure of the set X should allow relatively fast solution of subproblem (4.4) because it should be solved several hundred times during the optimization process. Practically, this means that the set X should be defined by linear constraints because in this case the projection problem (4.4) becomes a quadratic programming problem for which fast efficient algorithms exist. Nonlinear constraints of type (4.2) can be treated by extensions of algorithm (4.3) which employ penalty functions, duality approaches, or linear approximations.

Selection of the step direction. The most important part of an implementation of (4.3) consists of the construction of statistical estimates ξ^s of the gradient of the objective function $F_0(x)$ from (4.1). Considerable flexibility is allowed for the selection of such estimates, the formal requirement being that

$$\mathbb{E}(\xi^s | \mathbb{B}_s) = F_{0x}(x^s) + a_s, \tag{4.5}$$

where $\mathbb{B}_s$ is the σ-field defined by the history of the process, for example, by the sequence $x^0, \ldots, x^s$; a_s is the bias which should vanish asymptotically; and $F_{0x}(x^s)$ is the gradient of the function $F_0(x) = \mathbb{E}_\omega f_0(x, \omega)$ at the point x^s. This means that an estimate should

in average tend to the value of the gradient at the current point, while no requirements are made for the precision of the estimate. The vector ξ^s which satisfies the property (4.5) is called the *stochastic gradient*. In the case where the classical gradient does not exist, like in the case of convex but nonsmooth functions, ξ^s can be the estimate of the appropriate generalization of the gradient and is called *stochastic quasi gradient*.

Considerations about the precision of the estimates. In practice, the SQG algorithms can be tailored to work with very imprecise estimates, for example, with $\xi^s = f_{0x}(x, \omega^s)$, where ω^s is a *single* observation of the random vector ω. Even estimates with increasing and asymptotically unbounded variance are allowed; such estimates appear sometimes in the optimization of simulation models. Condition (4.5) can be relaxed even further to allow biased estimates which can result from dependent observations. In many cases it is enough to replace (4.5) by the requirement of a positive scalar product between the conditional expectation of ξ^s and $F_{0x}(x^s)$.

At the same time SQG methods can use more precise estimates when they are available in order to obtain faster speed of convergence. The general recommendation is to exploit the structure of the problem to minimize the computational effort necessary for obtaining an estimate of a given precision. This is because in many real problems the largest part of the computing effort is spent on the computation of the gradient estimates. Considerable research effort was dedicated to the development of gradient estimates that exploit the structure of specific classes of problems; see [1, 9, 17, 30].

Precision of the estimates ξ^s can be changed deliberately during the optimization process to reduce the total computational effort. During the initial phase the current point x^s is usually far away from the optimal solution, and even fairly imprecise estimates obtained with only a few values of observations of the random parameters ω or with a single observation can bring the current point closer to the optimum. In the later stages when the point x^s has reached some vicinity of the optimal solution the precision of the stochastic gradient ξ^s can be increased by making more observations of ω in each iteration.

Second-order methods. Second-order information about behavior of the function $F_0(x^s)$ can also be incorporated into the SQG methods. The basic iteration scheme (4.3) is modified as follows:

$$x^{s+1} = \pi_X(x^s - \rho_s A^s \xi^s), \tag{4.6}$$

where the matrix A^s carries the second-order information which can be extracted, for example, from the estimates of the Hessian of $F_0(x^s)$ or, more generally, from the local second-order approximation of $F_0(x^s)$; see [9, 35] for details. However, the stochastic case does not exhibit the dramatic increase in the speed of convergence that is usually associated with deterministic second-order methods. Due to the nature of stochastic problems the asymptotic speed of convergence with respect to the number of observations is approximately the same for both schemes (4.3) and (4.6). Besides, it may be computationally demanding to extract the second-order information in the stochastic case.

Selection of stepsize. Another important component of the SQG algorithm is the stepsize ρ_s in (4.3). Convergence of the algorithm to the optimal solution critically depends on the

correct selection of the stepsize. Again, considerable flexibility exists with respect to this selection. The most important consideration is that the stepsize selection rule should be coordinated with the precision of the stochastic gradient ξ^s. If the variance of ξ^s is bounded from above and from below away from zero, then the stepsize should tend to zero, but not excessively fast:

$$\rho_s \geq 0, \quad \rho_s \to 0, \quad \sum_{s=0}^{\infty} \rho_s = \infty. \tag{4.7}$$

This condition is known from the time of [31] and is supplemented by the requirement of convergence of the sum of squared stepsizes if almost sure convergence is required. Requirement that the stepsize tend to zero is not necessary if the precision of the stochastic gradients ξ^s is increasing with iterations. On the other hand, it should tend to zero faster if the variance of stochastic gradient grows with iterations, which can happen in some finite difference schemes and in optimization of some simulation models. Practical experience suggests that a piecewise constant stepsize is a good choice. The schedule which governs the change of the stepsize can be fixed beforehand, or it can depend on the information gathered during the optimization process.

Multiextremal problems. In most cases SQG methods will find a local minimum of the problem (4.1). However, it is possible to use modifications specifically designed to help an algorithm to skip many and sometimes even all the local minima which do not coincide with the global one. For example, the problem in section 5 exhibits a great many nonessential local minima introduced by the discrete sampling. One possibility is to use smoothing for filtering out the nonessential local minima; see also the example from [16] for an application of this technique to portfolio optimization. Another approach is to follow [9] and construct convex approximations which can cut off local minima. Still another way is to add large and infrequent perturbations to the step direction ξ^s following ideas of simulated annealing. This issue is treated in more detail in section 4.5, where an example solution of a multiextremal problem with the SQG method is considered.

4.3 Target application areas

In principle, a wide class of stochastic optimization problems (4.1)–(4.2) is solvable by SQG methods. At the same time some problems have features which make them better candidates for solution with SQG methods, while others are solved more efficiently with the methods based on deterministic equivalents. Let us illustrate this point by considering a stochastic linear program with recourse

$$\min_x c^\top x + \mathbb{E}_\omega Q(x, \omega), \tag{4.8}$$

$$Ax = b, \tag{4.9}$$

where

$$Q(x, \omega) = \min_y d^\top(\omega) y, \tag{4.10}$$

$$W(\omega) y = h(\omega) - R(\omega) x. \tag{4.11}$$

In the case when the distribution of ω is approximated by a finite number of scenarios $(p_i, \omega^i)_{i=1}^N$ and N is not excessively large, the best solution approach is to solve a deterministic equivalent by Benders decomposition. At the same time, from the duality theory for linear programming it follows that $Q(x, \omega)$ is a convex function of x with subgradient

$$\partial Q(x, \omega) = -R^\top(\omega)u^*(x, \omega),$$

where $u^*(x, \omega)$ is the set of all optimal solutions of the problem dual to (4.10)–(4.11) for fixed ω and x. For this reason it is possible to apply the SQG method (4.3) to solution of this problem with

$$\xi^s = c - R^\top(\omega^s)u^*(x^s, \omega^s),$$

where ω^s is an independent observation of random variables ω. Will this method be competitive with Benders decomposition? It depends on the problem parameters. If the dimension of the problem (4.10)–(4.11) is relatively large compared to the total number of scenarios, then Benders decomposition is clearly a better choice. On the other hand, SQG will become competitive for problems with relatively small dimension of (4.10)–(4.11) and a very large number of scenarios. Besides, SQG can be applied directly to problem (4.8)–(4.9) in the case of continuous distributions because it does not require a costly approximation procedure.

Below we describe some of the areas where SQG methods can have an advantage over other techniques.

Nonlinear problems. Besides SQG, another viable approach to the solution of nonlinear stochastic optimization problems consists of approximating the probability distribution by a discrete distribution, replacing the integrals in (4.1)–(4.2) by sums, and solving the resulting nonlinear optimization problem using nonlinear programming software. This approach can be efficient when the problem has only a few random parameters and it is possible to obtain a reasonably accurate approximation using a relatively small number of scenarios. If the number of random parameters increases, it may become very difficult to obtain a good approximation of the problem using a reasonable number of scenarios, and SQG methods will have an advantage.

Optimization of simulation models. Many problems in optimization of manufacturing processes, telecommunication networks, and supply chain management can be described by simulation models which depend on a finite number of decision parameters. At the same time these problems are extremely difficult to represent by normative linear and nonlinear programming models. This is especially true for the cases when the problem is described by a discrete event simulation model or one of the network models originated in computer science, like Petri nets, neural nets, or Bayesian nets. At the same time, in such cases it is possible to exploit the model structure to develop gradient estimation procedures; see [4, 14, 17, 21, 30, 32].

Finding optimal parametric policies in dynamic models. Dynamic stochastic optimization problems represent a serious challenge for numerical procedures. Such problems

can be formulated as follows:

$$\min_{x^1,\dots,x^T} \mathbb{E}_{\omega^1,\dots,\omega^T} \sum_{t=1}^{T} f_0^t(x^t, \omega^t), \tag{4.12}$$

$$f_i^t(x^{t+1}, x^t, \omega^t) \le 0, \quad i = 1:m, \quad t = 1:T-1, \tag{4.13}$$

where the minimization is performed with respect to functions

$$x^t = x^t(x^{t-1}, \omega^{t-1}), \quad t = 2, \dots, T. \tag{4.14}$$

The deterministic equivalent approach starts with observing that decision functions from (4.14) can be equivalently represented as $x^t = x^t(x^1, \omega^1, x^2, \omega^2, \dots, x^{t-1}, \omega^{t-1})$. Approximating the joint distribution of $\omega^1, \dots, \omega^{t-1}$ by a scenario tree brings this problem back to the realm of finite-dimensional optimization. In the linear case the resulting problem can be solved by nested Benders decomposition. This approach can be successful when the number of time periods is small and/or uncertainty can be represented by a relatively small scenario tree. However, in many other cases the size of the problem will be prohibitive because the number of nodes in the scenario tree grows as K^T, where K is the number of discrete values used for approximating the distribution of ω^t, which corresponds to the number of branches from any given node of the scenario tree. In this case an alternative approach can be used which, instead of approximating uncertainty, approximates the control policies (4.14) by a function with a fixed analytical form

$$x^t = \psi(a, x^{t-1}, \omega^{t-1}) \tag{4.15}$$

that depends on a vector a of parameters. After substituting (4.15) into (4.12)–(4.13), we obtain a finite-dimensional optimization problem whose complexity depends very moderately on the number of time periods:

$$\min_{x^1,a} \mathbb{E}_{\omega^1,\dots,\omega^T} \sum_{t=1}^{T} \varphi_0^t(x^1, a, \omega^1, \dots, \omega^t), \tag{4.16}$$

$$\varphi_i^t(x^1, a, \omega^1, \dots, \omega^t) \le 0, \quad i = 1:m, \quad t = 1:T-1. \tag{4.17}$$

Problem (4.16)–(4.17) will be nonlinear except in trivial cases and it can be solved by SQG methods. This approach works with a reduced set of policies but allows for richer representation of uncertainty. Which approach is better depends on a particular problem. Fix-mix portfolios in finance and (s, S) inventory policies are examples of such parametrized dynamic policies. In the case of asset and liability management of an insurance company such policies were studied in [15].

Problems with large or infinite number of scenarios. Finally, we observe here that SQG methods can be applied directly to problem (4.1)–(4.2) with continuous distributions. This is important because in many applied problems random parameters take a continuous set of values, and approximation with a finite number of scenarios is made only to facilitate the solution process. SQG methods do not depend on scenario approximation and require only the capability to observe the realizations of random variables.

But these attractive properties of SQG methods come at a price. Convergence to the optimal solution always occurs in a nonmonotonic and oscillatory manner due to the effects of randomness. Practical experience suggests that convergence occurs quite rapidly to the vicinity of the optimal solution and further improvement of the solution occurs quite slowly. The size of this vicinity depends on a particular problem, but as a rule of a thumb one can assume that it is defined by 2 to 5% of the difference between the value of the objective function at the starting point and the optimal value of the objective function. Very often such precision is sufficient, taking into account that the distributions of the random parameters are rarely known with higher precision.

4.4 Outline of the SQG system

The early version of SQG was developed for Unix systems and is described in [11] and [13]. The current version of SQG is developed for Windows platforms using the MATLAB environment [27]. It is available both as a MATLAB toolbox and as a stand-alone product which does not require MATLAB to be installed on the user machine. The choice of MATLAB as the development environment is justified by the following considerations:

- MATLAB takes care of a large part of the routine programming tasks, allowing the developer to concentrate on the specifics of algorithm implementation.
- MATLAB with additional toolboxes allows the developer to utilize a rich set of functions, which is useful for implementation of numerical and statistical algorithms.
- It is relatively easy to compile or rewrite the critical parts of MATLAB code in lower-level programming languages such as C or C++ to speed up the execution.
- If necessary, it is relatively easy to connect MATLAB code with specialized optimization libraries which can be used as the building blocks for implementation of the standard optimization tasks required by SQG. For example, to implement the projection operation in (4.3) we can choose between custom developed subroutines, the quadratic programming solver from the MATLAB Optimization toolbox, another solver available in the NAG library [29], or even solvers from callable CPLEX or XPRESS optimization engines. Implementation of stochastic programs with recourse can use commercially available solvers for computing stochastic gradients through the solution of linear programming problems (4.10)–(4.11).
- It is easy to connect MATLAB with Excel spreadsheets, possibly enhanced by Visual Basic add-ons for input of data and for model development.
- We have found that the relatively inefficient interpretative MATLAB language does not constitute a problem because the major part of the computation time is spent during computation of the observations of the objective function $f_0(x, \omega)$ performed by the model supplied by the user. The SQG system provides development guidelines that can considerably ease the user tasks.

The top-level architecture of the SQG system is shown in Figure 4.1. The core components are shown with solid borders, while additional components are shown with dashed borders. In what follows we describe each of the components in more detail.

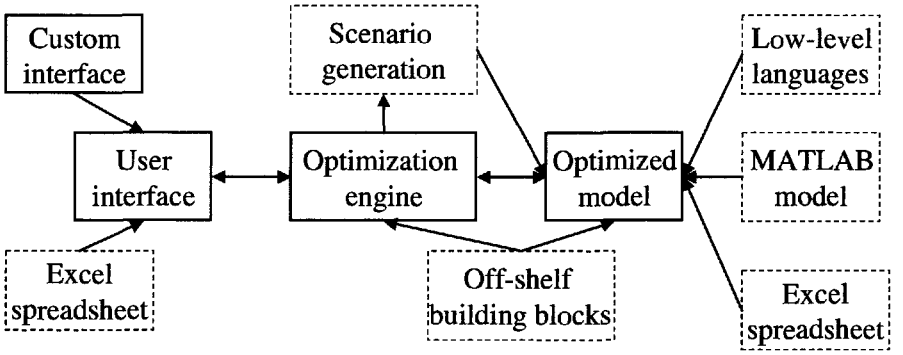


Figure 4.1. *The top-level architecture of the SQG system.*

The basic stand-alone version of SQG does not require additional software products, although some of the features can be enhanced by use of the products mentioned above.

4.4.1 Optimization engine

The optimization engine is the heart of the system which controls all other components and implements the iteration scheme (4.3). Its structure is shown in Figure 4.2, where the parts of the optimization engine and connections between them are shown with solid lines, while interactions with the other parts of the system are shown in dashed lines. In what follows we briefly describe the components of the optimization engine.

Initialization and data output. This block is responsible for setting up the optimization model from the information provided by the user, setting up the structure of the solution algorithm on the basis of information about the algorithm parameters contained in the parameter file or the parameter sheet, setting up connections with other components of the system, like Excel spreadsheets, setting up structures which keep records about the optimization process, and initializing the graphic user interface.

After initialization the iterative cycle is composed from the following parts.

Get stochastic gradient. This is one of the most important components of the optimization engine because it is responsible for computing a statistical estimate of the stochastic gradient ξ^s which defines the direction of the next move. Different possibilities are implemented because there is no universal way which is uniformly better for all the problems of interest.

In the simplest case the *optimized model* supplied by the user takes care of the computation of the estimate. Then the task of this block consists of calling the interface routine which gets the estimate from the optimized model for the current values of the decision variables and for new observations of the random parameters. In all other cases the interface routine is called to obtain one or more observations of the sample objective function $f(x, \omega)$ which are used to obtain the estimate of ξ^s. These function observations are uti-

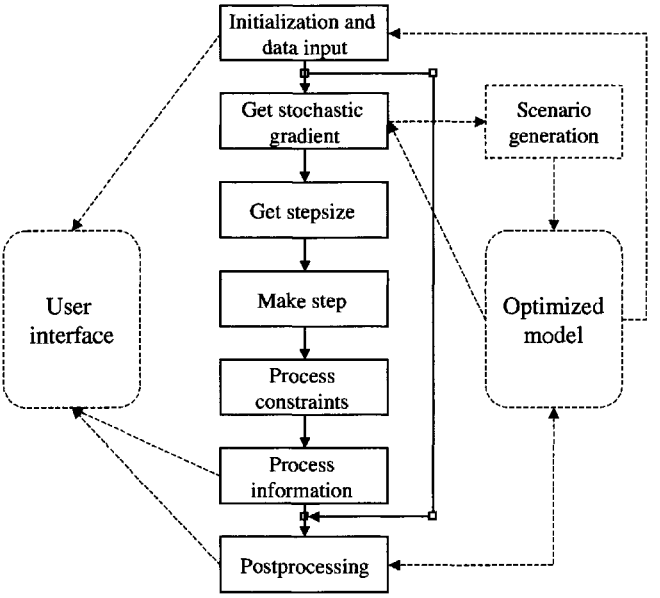


Figure 4.2. *Structure of the optimization engine.*

lized in several deterministic and stochastic finite difference estimation schemes. Another option utilizes the local approximation of $\mathbb{E}f(x,\omega)$ using current and possibly previous observations of the sample objective function following [9]. The second-order versions of this method can be considered to belong to the class of second-order SQG methods.

The estimates computed in this manner can be used directly or can be combined with the moving averages of previous estimates and with stochastic smoothing which can be useful for avoiding local minima; see the discussion in section 4.5. Besides, large-size rare probability moves are implemented to facilitate the search for the global minima in the nonconvex case.

Get stepsize. This component is responsible for selection of the length of the move ρ_s along the direction selected during the previous step. Several programmed rules are implemented along with several piecewise constant adaptive rules which condition the change of stepsize on the information gathered during the progress of the optimization process. Vector and matrix stepsizes are also available which are useful for scaling the problem.

Selection of a specific stepsize rule together with the selection of the specific stochastic gradient rule defines the concrete algorithm which implements the general iterative scheme (4.3). Around 20 such algorithms are implemented in the SQG system.

Make step. This is a very simple component which makes the next move using the stochastic gradient ξ^s and stepsize ρ_s computed by the components described above. More involved rules for the case of parallel movement of several points are under development.

Process constraints. Different projection algorithms are implemented here. Projection on bounds is a very simple operation; projection on a single linear constraint is made through a very efficient customized algorithm. Quadratic programming is used for projection on general linear constraints, while nonlinear constraints are treated by penalty functions. More efficient approaches for the treatment of nonlinear constraints are under implementation.

Process information. This component updates the values of functionals defined on the trajectories of the optimization process and checks the stopping criteria. Examples of such functionals are the moving averages of the samples of the objective function, stochastic gradients, current points x^s, and measures of oscillations of the optimization process. The values of these functionals are utilized in adaptive stepsize rules, in stopping rules, and to represent the optimization process to the user.

Postprocessing. This component analyzes the results of the optimization and performs the actions necessary for graceful termination or preparation for another run with a possibly different algorithm. One option here is to improve the estimate of $\mathbb{E}f(x,\omega)$ by repeated sampling at the final point. Comparing this estimate with a similar estimate at the starting point, one may evaluate the progress of the optimization process with reasonable precision. Another option allows plots of linear sections of the objective function in the vicinity of the final point.

The SQG system uses off-the-shelf routines for performing standard optimization tasks. Such tasks include the solution of linear and quadratic programming problems necessary for implementing some options in the *Process constraints* and *Get stochastic gradient* components. Several routines from the MATLAB optimization toolbox and the NAG numeric library are used. The system remains fully functional without these routines, with the exception of the options in question. The user can substitute other routines according to interface rules described in the manual.

4.4.2 Optimized model and scenario generation

To utilize SQG for the solution of his problems, the user has to describe the optimized model. This description consists of three parts: model header, model body, and model data. The model body should provide to SQG the observations of the sample objective function or its gradient. The user can select from three possibilities:

1. *MATLAB function.* The function should start with the line

```
function [fval,grad]=OptimizedModel(x,scenario)
```

followed by the function body which computes the value of $f(x,\omega)$ for the fixed value of decision variables x and the `scenario` value of the random variables ω and assigns this value to the variable `fval`. The second output variable `grad` gives a user the possibility of passing to SQG the value $f_x(x,\omega)$ of the gradient of the sample objective function. If the user chooses to use this possibility, it can speed up the solution process. However, in most cases it will be too tedious for the user to

perform such a task. In such cases the empty value should be assigned to `grad` in the body of function `OptimizedModel` through the statement

```
grad=[];.
```

- The value of x is supplied to `OptimizedModel` by SQG during the call. For the `scenario` value there are two possibilities.
- Generation of subsequent independent observations of ω (scenarios) can be performed by the user as part of `OptimizedModel`. In this case SQG uses `scenario` input to tell `OptimizedModel` when the new independent value of ω should be generated and when the previous value should be used.
- SQG generates scenarios values using the *Scenario generation* component and passes them to `OptimizedModel` in the variable `scenario`. In this case the user should supply to SQG the description of the distributions of the random parameters ω in the model definition placed in an Excel spreadsheet. Several distributions and scenario generation techniques are implemented in the *Scenario generation* component, and the number is growing.

2. *C/C++ function compiled as DLL (dynamic link library).* The rules for defining such functions are similar to the case of MATLAB functions.

3. *Excel spreadsheet which calculates the values of the sample function $f(x, \omega)$ and places it in a specified cell.* Such spreadsheets can use custom functions of arbitrary complexity written in Visual Basic. In this case SQG starts the Excel server and during each iteration places the current values of the decision variables in the specified ranges of the spreadsheet. If the *Scenario generation* component of SQG is used, the current scenarios of ω are also passed to the spreadsheet. The spreadsheet calculates the value of $f(x, \omega)$; SQG retrieves this value from the specified cell.

Model data are placed in an Excel spreadsheet and should contain the data used for the generation of the linear constraints and parameters of the distributions of the random parameters if the *Scenario generation* component of SQG is used.

The model header describes the general model properties and options and defines the cell ranges for the model data and the interface with the model body. It uses a simple declarative model language for this purpose. The model header for the supply chain example from section 4.5 is provided in Figure 4.3.

In this example the header, body, and data of the model are contained in the same spreadsheet. Figure 4.3 shows the part of the spreadsheet with the header. Keywords are shown in boldface. The first line says that the decision variables are contained in the range named InventoryLevels. During the optimization process SQG will put in this range the current values of variables x. The second and third lines define two groups of random parameters contained in the ranges Demand and DemandC2. SQG will generate scenarios for these variables and put them in these ranges. The random parameters are distributed uniformly with bounds for the uniform distributions contained in the ranges DemandBounds and DemandBoundsC2, respectively. The variables have time dimension which in the model body is oriented along columns. The fourth line says that constraints are described in the

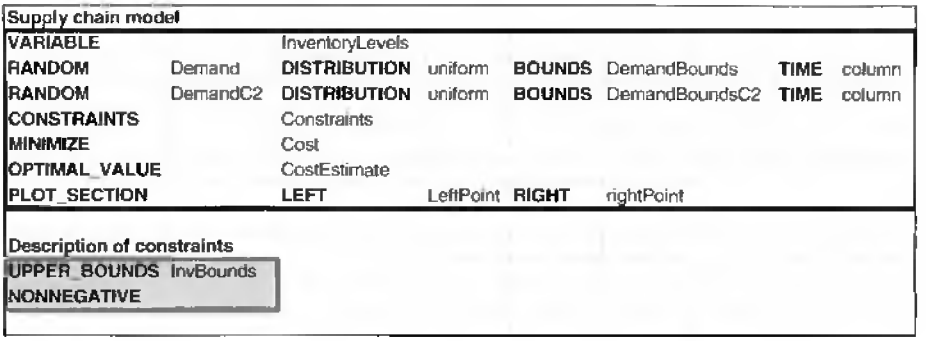


Figure 4.3. *Model header for supply chain example.*

range Constraints. This range is shown in the lower part of the header. It specifies that variables are nonnegative and have the upper bounds contained in the range InvBounds. The last two lines refer to postprocessing. They tell the system that the estimate of $\mathbb{E} f(x, \omega)$ should be placed in the range CostEstimate and that a plot of the linear section of the objective should be made between points contained in the ranges LeftPoint and rightPoint.

In some cases the user will be able to provide SQG not only with the observations of the objective function but also with the values of its gradient. This information can be passed to SQG in a way similar to that described

This architecture gives a lot of flexibility to the user, who can use the means for model definition which best suit a given problem and at the same time utilize in the best way the user's modeling and programming skills.

4.4.3 User interface and optimization modes

The purpose of the graphical user interface is to help the user solve the problem by providing the means for defining the problem, informing the user about the progress of the optimization process, and providing the user with the capability of intervention in the optimization process. The user interface consists of two parts:

1. *Model definition interface.* This is provided by an Excel spreadsheet and is used for defining the model header, model data, and, optionally, the model body.

2. *Optimization process interface.* This is a custom interface which is integrated with the optimization engine and provides the following capabilities:

   - selection of the algorithm, which consists of selection of the type of stochastic gradient and stepsize to use;
   - selection of algorithm parameters;
   - display of the optimization progress, which consists of displaying the function estimates, selected variables, and algorithm performance functionals; and
   - dynamic change by the user of algorithm parameters and algorithm based on the progress of the optimization process.

Not all users will want or be able to exploit all the possibilities offered by the optimization process interface. For this reason SQG can run in two modes, which differ by the amount of user intervention required:

- *Automatic mode.* In this case the user has only to specify the model to optimize. SQG will use the default algorithm and default values of algorithm parameters.

- *Interactive mode.* This gives the user the ability but not the obligation to change algorithms and algorithm parameters in the course of the optimization.

Suppose now that the user has to solve a set of similar problems and is not satisfied with performance of the default algorithm with default parameters. In this case one can use the interactive mode to solve a typical problem from this set, change defaults, and proceed with solving the other problems in automatic mode.

4.5 Example 1: Supply chain management

The design of supply chains gives rise to an important class of decision problems which can be formulated naturally as stochastic programming problems. The major source of uncertainty is demand because even the demand projections over short time horizons can be off the mark by 50% or more. Prices and disturbances in technological processes can also be important sources of uncertainty. The importance of the adequate treatment of uncertainty was long recognized in this environment, and a number of ad hoc approaches were developed, like maintenance of buffer stocks and various other inventory management policies.

The objective of planning is to satisfy conflicting aims of maintaining specific time targets for the satisfaction of demand while minimizing inventory, production, and ordering costs. Usually a given production unit or a given inventory was considered independently from other elements of the supply chain, which considerably simplified the analysis of such systems [34]. This approach is insufficient in the current drive toward ever stricter performance targets. Stochastic optimization can contribute considerably to the formulation of more robust supply chain design and management policies with multiple uncertainties and multiple components.

At the same time supply chains combine several features discussed in section 4.3 and in particular they exhibit strong nonlinearities and nontrivial dynamic behavior. This makes them a natural candidate for application of SQG methods which can be used for finding optimal dynamic parametric policies and for combining simulation with optimization. In this section we provide one such example.

The supply chain which we consider in this example is depicted in Figure 4.4. This supply chain satisfies random demand D_P for finished product P. Manufacture of this product is a complex process, the last part of which is the assembly from different components performed at plant Assembly P. The most critical of these components are C1 and C2, which are used in quantities n_1 and n_2 for production of one unit of P. These components are sent to the assembly line from Inventory C1 and Inventory C2. Inventory C1 is replenished by sending orders to external suppliers according to an (s_1, S_1) inventory management policy. That is, an order is issued when the size z_1 of inventory becomes less than s_1 and the size

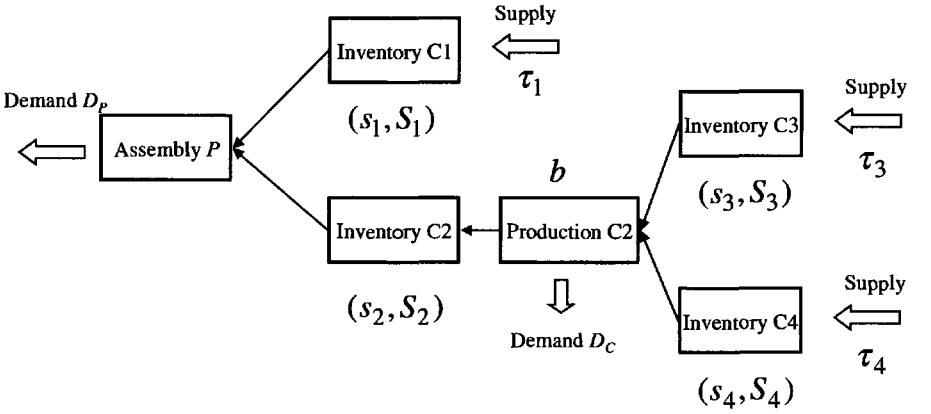


Figure 4.4. *Example of supply chain.*

of the order is $S_1 - z_1$. This order is filled after τ_1 days, which can be a random variable. Inventory C2 is managed similarly according to an (s_2, S_2) policy. However, instead of sending orders to an external supplier it sends them to the internal production facility Production C2. Besides internal orders from Inventory C2 this facility satisfies external random demand D_C for C2. Production C2 has constant production rate b and produces C2 only when there are pending orders from Inventory C2 or unsatisfied external demand. Orders from Inventory C2 have priority over the external demand. In its turn, C2 consists of a number of other components; the most important such components are C3 and C4, which are used in quantities n_3 and n_4 for one unit of C2. These components are sent to Production C2 from Inventory C3 and Inventory C4. These inventories send orders for C3 and C4 to external suppliers according to (s_3, S_3) and (s_4, S_4) inventory management policies, respectively. The random time delays between issuing the order and its satisfaction are τ_3 and τ_4, respectively. Observe that we do not consider explicitly the time delay τ_2 for the orders of C2 because it is endogenous to the model.

The management objective of this supply chain is to satisfy demands D_P and D_C while minimizing the total costs. These costs are composed of three components: inventory costs and ordering costs for components C1, C2, C3, and C4 and backlog costs for demands D_P and D_C. Backlog costs arise when the time targets for demand satisfaction are not satisfied. The behavior of this system is simulated during one year with the unit time interval of one day. This simulation provides the observations of the total yearly costs. The observations of the average daily costs $f_0(x, \omega)$ are obtained by dividing these costs by the number of days in the simulation horizon. The vector of decision variables x is composed of the inventory management parameters (s_i, S_i), $i = 1, \ldots, 4$. The vector of random parameters ω includes daily demands D_P and D_C over the time horizon of one year and order lead times $\tau_1, \ldots, \tau_3, \tau_4$ for all orders. Following the architecture described in section 4.4, the simulation model was implemented as an Excel spreadsheet. The SQG system was used to obtain the values of decision variables which minimize the averaged daily costs $\mathbb{E} f_0(x, \omega)$.

Although this example contains only eight decision variables, it presents a serious challenge to optimization technology. First, the uncertainty is a very substantial feature of

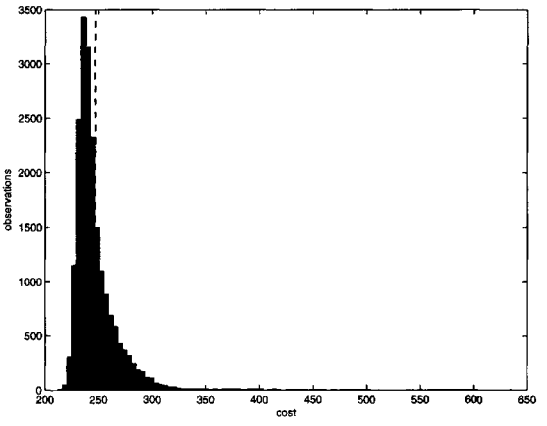


Figure 4.5. *Histogram of $f(x, \omega)$ close to the optimal solution.*

this problem, as Figure 4.5 illustrates.

Figure 4.5 shows the histogram of $f(x, \omega)$ in the vicinity of the optimal solution obtained after 20,000 observations, where the vertical dashed line indicates the position of $\mathbb{E}f(x, \omega)$. Note the skewness of the distribution of $f(x, \omega)$ which has outliers exceeding 600 while $\mathbb{E}f(x, \omega) = 247.0 \pm 0.3$. This feature will require longer simulation times and represents an additional challenge to the optimization process.

Even more serious is the fact that the sample objective function $f_0(x, \omega)$ contains multiple local minima and many of them are located far away from the global one. Moreover, $f_0(x, \omega)$ has numerous discontinuities which result from the fact that a small change in decision variables may lead to a change in the number of orders and, consequently, in a finite jump in the total costs. These features will be inherited also by any approximation of $\mathbb{E}f_0(x, \omega)$ by a weighted sum of $f_0(x, \omega^i)$ over a finite number of observations $\omega^i, i = 1; N$ for any reasonable value of N. The number of local minima and discontinuities in such sums will be even larger than in the individual functions $f_0(x, \omega^i)$, although the size of jumps will decrease. For this reason any approach based on the approximation of uncertainty by a finite number of scenarios with subsequent use of nonlinear programming codes is bound to fail. These difficulties are partly inherited even by the expected cost function $\mathbb{E}f_0(x, \omega)$, which is continuous but may exhibit numerous local minima. This point is illustrated in Figure 4.6.

Figure 4.6 shows dependence of the total cost estimate on the value of S_2 for fixed values of other parameters. The thin curve shows the sample cost $f_0(x, \omega)$ for a particular observation of ω, while the thick curve shows the estimate $\hat{F}(x)$ of $\mathbb{E}f_0(x, \omega)$ obtained by averaging 1000 observations of $f_0(x, \omega)$. Within the scale of this graph this estimate would be practically indistinguishable from $\mathbb{E}f_0(x, \omega)$ because the size of the 95% confidence interval does not exceed 12. The figure shows the highly irregular behavior of $f_0(x, \omega)$ with many jumps and local minima and the smoother behavior of the expected cost $\mathbb{E}f_0(x, \omega)$ which, however, retains a few local minima with cost values far away from the optimum. Two vertical thin lines indicate the range of the horizontal axis used in Figure 4.7.

Figure 4.6 further suggests that $\hat{F}(x)$ is more regular than $f_0(x, \omega)$, but this is only

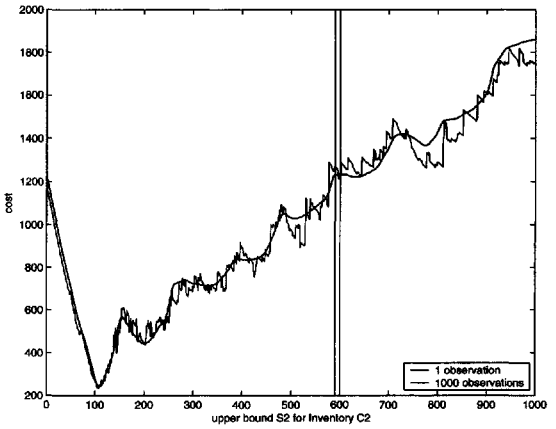


Figure 4.6. *Dependence of the cost function of supply chain example on S_2.*

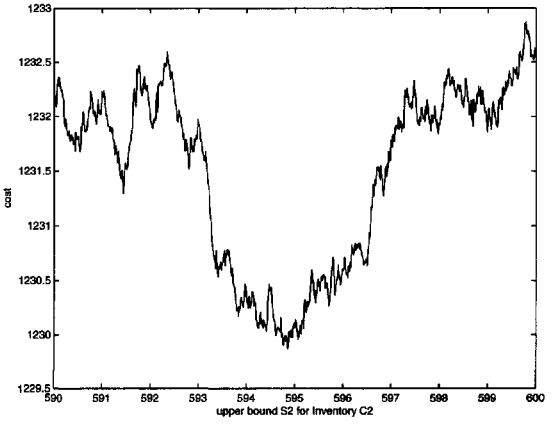


Figure 4.7. *Dependence of the cost function of supply chain example on S_2, detail.*

an illusion, as Figure 4.7 illustrates. This figure shows a small portion of the plot of $\hat{F}(x)$ from Figure 4.6 depicted there by the thick line. The greater detail of Figure 4.7 allows one to see that in reality $\hat{F}(x)$ behaves extremely irregularly with hundreds of jumps and local minima, where only one local maximum is visible in Figure 4.6. To better highlight this phenomenon the ratio between vertical and horizontal scales is made smaller in Figure 4.7 compared to Figure 4.6.

Summarizing, the sample objective function in this case is formed from the superposition of several patterns. There is a relatively well-behaved global pattern with a distinct global minimum, and there are one or more very irregular local patterns with multiple local minima. Increasing the number of samples does not help to eradicate the most irregular high-frequency pattern; it only reduces its magnitude and adds irregularities. This pattern is not present in $\mathbb{E}_\omega f_0(x, \omega)$, but it still can possess a considerable number of extremal points

of medium scale, as Figure 4.6 suggests. Potentially, this problem can be treated by genetic algorithms. However, genetic algorithms are not designed to use the distinct global pattern which is present in this case.

The SQG system contains several options specifically developed with the aim of filtering out local irregularities. One such option is stochastic smoothing, which consists of substitution of the original objective function $F(x) = \mathbb{E}_\omega f_0(x, \omega)$ by the smoothed function $F(x, r) = \mathbb{E}_y \mathbb{E}_\omega f_0(x + y, \omega)$ obtained by taking the expectation with respect to the artificially introduced random vector y with distribution depending on the smoothing parameter r. In the simplest case y is distributed uniformly over the hypercube of size r and center at zero. When $r \to 0$ the smoothed function tends to the original function $F(x)$, while for positive r the smoothing effect can be substantial. This is because for appropriate values of r the smoothing operation averages out the high frequency irregular component of the objective function and eliminates local minima created by this component. Stochastic smoothing does not add complexity to the computation of the stochastic gradient because it is equivalent to computation of the stochastic gradient at point $x^s + y^s$ instead of point x^s, where y^s is an observation of the random vector y. In the simplest case, one can take

$$\xi^s = f_{0x}(x^s + y^s, \omega^s),$$

where ω^s and y^s are independent observations of original and auxiliary random vectors.

However, smoothing alone may not be sufficient to eliminate all the local minima. For this reason it is important to combine smoothing with the possibility of moving to a different part of the feasible region which can be in the vicinity of another and possibly better local minimum. In the context of SQG this is achieved by adding large-scale low-probability disturbances to the stochastic gradient:

$$\xi^s = \xi^{1s} + \xi^{2s},$$

where ξ^{1s} is the stochastic gradient computed according to one of the rules described above and ξ^{2s} is the disturbance whose value almost always equals zero but which occasionally takes large values.

These approaches were combined for solving the supply chain optimization example from Figure 4.4. The first 400 iterations of a typical run are shown in Figure 4.8.

Forward finite differences were used to compute the stochastic gradients, and consequently nine computations of $f(x, \omega)$ were made at each iteration. All nine computations were made for the same fixed observation of the random parameters ω which changed from iteration to iteration. The average of these values is shown as the thin irregular curve in Figure 4.5. The jumps of this curve between iteration 280 and 340 are explained by outliers shown in Figure 4.5. The thick and more regular curve represents a moving average of these values. This moving average is used as an estimate of the current value of $\mathbb{E}_\omega f_0(x, \omega)$ to control the value of the stepsize.

To evaluate the progress of the optimization process relatively precise estimates of $\mathbb{E}_\omega f_0(x, \omega)$ were made at the starting point and after 400 iterations. (This amounts to evaluating $\hat{F}$ at x^0 and x^{400}, each of which requires 1000 function evaluations.) The starting point was located far away from the optimal solution with the 95% confidence interval for the value of $\mathbb{E}_\omega f_0(x^0, \omega)$ being 2333.1 ± 1.8. After 400 iterations the estimate became 299.2 ± 0.3, while the value of the moving average was 312.3, constituting a fairly tight

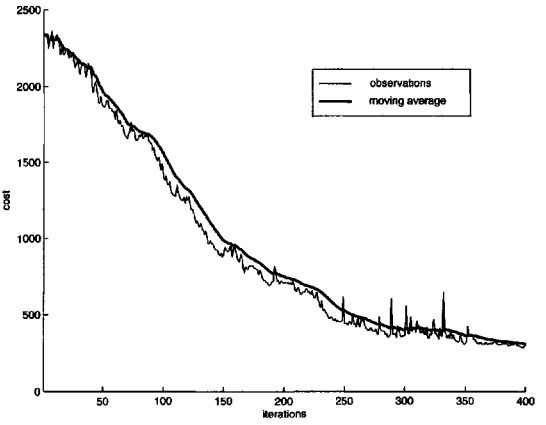


Figure 4.8. *Example of the optimization process.*

upper bound for the current value of the objective function. The decrease in the value of the objective function continued following the usual convergence pattern of SQG methods illustrated by Figure 4.8: relatively fast convergence to a vicinity of the optimal solution with further convergence proceeding at a slower rate. We conducted numerous experiments with this example from different starting points, and the best value of the objective function that we found was 246.3 ± 1.3. Taking this as the optimal value F_0^* and measuring the relative precision η^k obtained after k iterations in percent of the initial difference between the current and the optimal function values

$$\eta^k = \frac{F_0(x^k) - F_0^*}{F_0(x^0) - F_0^*} 100\%,$$

we obtain that in the case described in Figure 4.8, $\eta^{400} = 2.5\%$, which is a typical precision that can be expected from SQG methods after 200 to 400 iterations. Higher precision is not necessary in this case because the distribution of demand is seldom known with higher precision. Finally, we used 3600 evaluations of $f_0(x, \omega)$ for the purposes of optimization during the first 400 iterations, which seems small, considering the 2000 function evaluations needed for the estimates mentioned above.

Convergence in the space of decision variables follows the same pattern, although in the case where the function is very flat in the vicinity of the optimal solution, quite considerable oscillations can occur even when the stepsize becomes small.

4.6 Example 2: Pension fund management

Financial applications constitute another high-potential application area for SQG methods. Many asset and risk management problems in finance and insurance are multiperiod dynamic problems with multiple sources of uncertainty; see [5, 6, 39, 40]. The concepts of risk aversion are usually represented by nonlinear functions. In many cases use of dynamic

parametrized policies and optimization of simulation models are the preferred ways to solve the problem; see [38].

We present here a model of pension fund management solved by the SQG system. A pension fund collects contributions from participants, invests them in financial assets, and pays back the resulting amount of money when a contributor arrives at a pensionable age. The model is a dynamic multiperiod model with periods $t = 1, \ldots, T$ and time steps that vary from one quarter to one year. The model body is implemented as an Excel spreadsheet and contains the following components.

Contribution inflow. This describes the flow of contributions z_t into the pension fund. This component consists of code written in Visual Basic which implements the legal norms that govern the contributions to the pension fund and the process of entering and exit of new contributors. This process can be a source of uncertainty, although in a fund with many thousands of contributors it can be a relatively minor source. Since contributions are dependent on income dynamics, this component is connected with the macroeconomic component.

Macroeconomic and asset dynamics. This component describes the dynamics of asset prices and macroeconomic variables v_t, like inflation rate during the time horizon $t = 1, \ldots, T$. Assets considered here are the broad asset classes described by price indices for different types of markets. The total number of assets varies from 5 to 30. This is the major source of uncertainty in the model and was described by the recursive equation

$$v_{t+1} = v_t + \omega_t,$$

where ω_t is a vector of correlated random variables. The correlation may be present both within the period and between the periods. An important part of this component is the generation of observations/scenarios of ω_t. This was organized in a way similar to the supply chain management example from the previous section. Data describing properties of the distribution of ω_t were placed in the same Excel spreadsheet as the model body and were passed to SQG, which generated scenarios and put them back in the spreadsheet.

Liabilities. This component describes the cash outflow u_t generated by the obligations of the pension fund to contributors as well as the total value of the current liabilities U_t. It is governed by the normative framework and by an inflow stream generated by the contributions. This component together with the contribution inflow component can contain decision variables in the case in which the pension fund offers products with guarantees.

Cash flow and investment. This component describes the investment process of the fund and the evolution of its total value. The fund must invest the contribution cash flow into different assets from the set of investment opportunities I to meet its obligations and compete with other funds. The total value V_t of the fund assets consists of the sum of the values of individual assets y_t^i which change as follows:

$$y_{t+1}^i = \frac{v_{t+1}^i}{v_t^i} y_t^i - y_t^{i-} + y_t^{i+}, \quad i \in I,$$

where y_t^{i-} is the value of asset i that was sold during time period t and y_t^{i+} is the value of asset i bought during this period. The cash flow equation connects the investment process with the contributions and liabilities:

$$z_t + \sum_{i \in I} (1 - \alpha_i) y_t^{i-} = \sum_{i \in I} (1 + \alpha_i) y_t^{i+} + u_t,$$

where α_i is a coefficient that describes transaction costs. This component includes different constraints that are imposed on admissible asset allocation of a pension fund by regulation. Decision variables x_i define investment policies. In our context they are parameters of dynamic investment strategies. The simplest example of such a strategy is the fixed-mix strategy, which divides the total investment into fixed fractions x_i between assets and rebalances the portfolio to keep these fractions constant. In the absence of transaction costs these variables define the values of y_t^i from the following relation:

$$y_{t+1}^i = x_i \left(\sum_{i \in I} \frac{v_{t+1}^i}{v_t^i} y_t^i + z_t - u_t \right),$$

$$y_t^{i+} = \max \left\{ 0, y_{t+1}^i - \frac{v_{t+1}^i}{v_t^i} y_t^i \right\}, \qquad y_t^{i-} = \max \left\{ 0, \frac{v_{t+1}^i}{v_t^i} y_t^i - y_{t+1}^i \right\}$$

Other examples of parametrized dynamic portfolio selection strategies can be found in [39, 18].

The objective function of portfolio fund management is to satisfy the fund's obligations while maximizing financial performance. This can be expressed as a nonlinear function $\mathbb{E}_\omega f(V, U, \omega)$, where $V = (V_1, \ldots, V_T)$, $U = (U_1, \ldots, U_T)$, $\omega = (\omega_1, \ldots, \omega_T)$ defined on the fund's trajectories.

The SQG system was used for finding the optimal values of the parameters of parametrized dynamic investment strategies, including the case of products with fixed guarantees. Although the sample objective function is continuous here, unlike the case described in the previous section, it still may have multiple local minima. Therefore some of the techniques used in section 4.5 were used in the solution of this problem.

4.7 Summary

We described here the SQG software system for solving stochastic programming problems with stochastic quasi-gradient methods. These methods belong to the class of sampling methods, which are complementary to the methods based on deterministic equivalents. Their target field of applications includes nonlinear stochastic optimization problems, problems with continuous distributions or with a very large number of scenarios, dynamic stochastic optimization problems with parametrized decision strategies, and optimization of simulation models. Supply chain management, finance, and energy production contain many examples of such decision problems under uncertainty.

Bibliography

[1] F. ARCHETTI, A. A. GAIVORONSKI, AND F. STELLA, *Stochastic optimization on Bayesian nets*, European J. Oper. Res., 101 (1997), pp. 360–373.

[2] J. R. BIRGE, *An L-shaped method computer code for multistage stochastic linear programs*, in Numerical Techniques for Stochastic Optimization, Ser. Comput. Math. 10, Springer-Verlag, New York, 1988, pp. 255–266.

[3] J. R. BIRGE AND F. LOUVEAUX, *Introduction to Stochastic Programming*, Springer-Verlag, New York, 1997.

[4] C. G. CASSANDRAS, *Discrete Event Systems: Modeling and Performance Analysis*, Irwin Publishers, Boston, 1993.

[5] G. CONSIGLI AND M. DEMPSTER, *Dynamic stochastic programming for asset-liability management*, Ann. Oper. Res., 81 (1998), pp. 131–161.

[6] J. DUPACOVA, M. BERTOCCHI, AND V. MORIGGIA, *Postoptimality for scenario based financial planning models with an application to bond portfolio management*, in Worldwide Asset and Liability Management, W. Ziemba and J. Mulvey, eds., Cambridge University Press, Cambridge, UK, 1998, pp. 263–285.

[7] Y. ERMOLIEV, *Methods of Stochastic Programming*, Nauka, Moscow, 1976 (in Russian).

[8] Y. ERMOLIEV, *Stochastic quasigradient methods and their application to system optimization*, Stochastics, 9 (1983), pp. 1–36.

[9] Y. ERMOLIEV AND A. A. GAIVORONSKI, *Stochastic quasigradient methods for optimization of discrete event systems*, Ann. Oper. Res., 39 (1992), pp. 1–39.

[10] Y. ERMOLIEV AND R. J.-B. WETS, EDS., *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin, 1988.

[11] A. GAIVORONSKI, *Interactive Program SQG-PC for Solving Stochastic Programming Problems on IBM Compatibles: User Guide*, Technical Report WP-88-11, IIASA, Laxenburg, Austria, 1988.

[12] A. A. GAIVORONSKI, *On nonstationary stochastic optimization problems*, Cybernetics, 14 (1978), pp. 89–92.

[13] A. A. GAIVORONSKI, *Implementation of stochastic quasigradient methods*, in Numerical Techniques for Stochastic Optimization, Y. Ermoliev and R. J.-B. Wets, eds., Springer-Verlag, Berlin, 1988, pp. 313–352.

[14] A. A. GAIVORONSKI, *Convergence properties of backpropagation for neural nets via theory of stochastic gradient methods, Part* I, Optim. Methods Softw., 4 (1994), pp. 117–134.

[15] A. A. GAIVORONSKI AND P. E. DE LANGE, *An asset liability management model for casualty insurers: Complexity reduction vs. parametrized decision rules*, Ann. Oper. Res., 99 (2000), pp. 227–250.

[16] A. A. GAIVORONSKI AND G. PFLUG, *Value-at-risk in portfolio optimization: Properties and computational approach*, J. Risk, 7 (2004/05), pp. 1–31.

[17] A. A. GAIVORONSKI, L. Y. SHI, AND R. S. SREENIVAS, *Augmented infinitesimal perturbation analysis: An alternate explanation*, Discrete Event Dyn. Syst., 2 (1992), pp. 121–138.

[18] A. A. GAIVORONSKI AND F. STELLA, *On-line portfolio selection using stochastic programming. High-performance computing for financial planning*, J. Econom. Dynam. Control, 27 (2003), pp. 1013–1043.

[19] H. GASSMANN, *MSLIP: A computer code for the multistage stochastic linear programming problem*, Math. Program., 47 (1990), pp. 407–423.

[20] J. L. HIGLE AND S. SEN, *Stochastic decomposition: An algorithm for two-stage linear programs with recourse*, Math. Oper. Res., 16 (1991), pp. 650–669.

[21] Y. C. HO AND X. R. CAO, *Perturbation Analysis of Discrete Event Dynamic Systems*, Kluwer, Boston, 1991.

[22] G. INFANGER, *DECIS: A System for Solving Large-Scale Stochastic Programs*, Stanford University, Palo Alto, CA, 1997.

[23] P. KALL AND S. WALLACE, *Stochastic Programming*, John Wiley, New York, 1994.

[24] J. KIEFER AND J. WOLFOWITZ, *Stochastic approximation of a maximum of a regression function*, Ann. Math. Statist., 23 (1952), pp. 462–466.

[25] A. J. KING, *SP/OSL Version* 1.0 *Stochastic Programming Interface User's Guide*, Technical Report, Research Division, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1994.

[26] K. MARTI AND E. PLÖCHINGER, *Optimal step sizes in semi-stochastic approximation procedures*, Optimization, 21 (1990), pp. 123–153.

[27] *MATLAB* 6.5, The MathWorks, Inc., Natick, MA, 2002.

[28] J. MAYER, *Stochastic Linear Programming Algorithms: A Comparison Based on a Model Management System*, Gordon and Breach, Amsterdam, 1998.

[29] *NAG C Library, Mark* 6, The Numerical Algorithms Group, Oxford, UK, 2001.

[30] G. C. PFLUG, *Optimization of Stochastic Models: The Interface between Simulation and Optimization*, Kluwer, Boston, 1996.

[31] H. ROBBINS AND S. MONRO, *A stochastic approximation method*, Ann. Math. Statist., 22 (1951), pp. 400–407.

[32] R. RUBINSTEIN AND A. SHAPIRO, *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization*, John Wiley, New York, 1993.

[33] A. RUSZCZYŃSKI AND W. SYSKI, *A method of aggregate stochastic subgradients with on-line stepsize rules for convex stochastic programming problems*, Math. Program. Stud., 28 (1986), pp. 113–131.

[34] E. A. SILVER, D. F. PYKE, AND R. PETERSON, *Inventory Management and Production Planning and Scheduling*, John Wiley, New York, 1998.

[35] S. URYASEV, *A stochastic quasi-gradient algorithm with variable metric*, Ann. Oper. Res., 39 (1992), pp. 251–267.

[36] R. M. VAN SLYKE AND R. WETS, *L-shaped linear programs with applications to optimal control and stochastic programming*, SIAM J. Appl. Math., 17 (1969), pp. 638–663.

[37] R.-B. WETS, *Stochastic programming: Solution techniques and approximation schemes*, in Mathematical Programming: The State of the Art, A. Bachem, M. Groetshel, and B. Korte, eds., Springer-Verlag, Berlin, 1982.

[38] S. A. ZENIOS, *High-performance computing in finance: The last 10 years and the next*, Parallel Comput., 25 (1999), pp. 2149–2175.

[39] W. ZIEMBA AND J. MULVEY, EDS., *Worldwide Asset and Liability Management*, Cambridge University Press, Cambridge, UK, 1998.

[40] W. ZIEMBA, *The Stochastic Programming Approach to Asset-Liability and Wealth Management*, AIMR, Charlottesville, VA, 2003.

Chapter 5

Computational Grids for Stochastic Programming

Jeff Linderoth and Stephen J. Wright†*

5.1 Introduction

Parallel supercomputers continue to increase in power and in ability to solve very large and complex problems in computational science. For many users, however, a number of practical limitations are associated with these machines, including their high cost, the difficulty of obtaining access to them, and the difficulty of writing or procuring software tools that execute on them. In recent years, there has been a good deal of interest in alternative computing platforms known as *computational grids* [10], which are made up of large collections of geographically dispersed CPUs, storage, and visualization devices linked by local networks and the Internet. Of particular interest to the optimization community are computational grids that are made up of workstations, PCs and PC clusters, and supercomputer nodes, and which may be owned by a number of different individuals and institutions. Grids such as these grant access to compute cycles that would not otherwise be used by the owners of the machines of which they are composed, without interfering with the computing activities of the machine owners. As we discuss in the next section, computational grids have some obvious drawbacks that make them difficult to use. Nevertheless, the tremendous power and low cost of these platforms makes them appealing for large-scale computations of certain types, as has been demonstrated in the popular `SETI@home` project [34]. Recently, commercial ventures have arisen to build and harness computational grids [9].

Perhaps more than any other area of computational optimization, stochastic programming has made use of parallel computing to solve large problem instances; see [2, 3, 8, 11, 13, 18, 19, 20, 28, 31]. In this earlier work, different problem types and different algo-

*Department of Industrial and Systems Engineering, Lehigh University, 200 West Packer Avenue, Bethlehem, PA 18015-1582 (jtl3@lehigh.edu)

†Computer Sciences Department, 1210 West Dayton Street, University of Wisconsin, Madison, WI 53706 (swright@cs.wisc.edu)

rithms were matched to particular parallel platforms in different ways, with decomposition according to scenario a recurring theme in many approaches. In this chapter, we describe the characteristics of the type of grid outlined in the previous paragraph and point out in broad terms why its properties are a good match for stochastic programming problems and algorithms. We focus in particular on grids based on the Condor system [24] and on the MW runtime support library for supporting master-worker computations on these grids [14]. We then discuss our asynchronous decomposition algorithm for two-stage stochastic linear programs with recourse and describe in some detail its implementation on the grid, using Condor and MW. We present computational results obtained for some very large sampled instances of stochastic programs from the literature, including comparisons with results obtained for other software tools. Finally, we describe some future enhancements to grid infrastructure that will equip them to solve other classes of stochastic programs, including stochastic integer programs.

5.2 Computational grids

The term "grid computing" (synonymously "metacomputing") is generally used to describe parallel computations performed on a geographically distributed, heterogeneous computing platform. The particular variant of this concept of interest to us is a parallel platform made up of workstations, PCs and PC clusters, and supercomputers, owned by various individuals and entities and distributed across a campus or other organization, or across the world. Although such platforms are powerful and inexpensive, they are difficult to harness for productive use, for the following reasons:

- *Heterogeneity.* Resources may vary widely in their operational characteristics: memory, swap space, processor speed, operating system, installed software.
- *Poor communications properties.* Latencies (times to pass messages between processors) may be high, variable, and unpredictable.
- *Unreliability.* Resources may disappear without notice; for example, a workstation may be reclaimed by its owner, resulting in termination of any computations being performed by us on that machine.
- *Dynamic availability.* The pool of available processors grows and shrinks during the computation, according to the changing priority of our job, the claims of other users, and scheduling considerations.

In all these respects, our target platform differs from conventional multiprocessor platforms (such as IBM-SP or SGI Origin machines) and from Beowulf clusters. However, applications developed to run on computational grids often can be executed efficiently on these more conventional platforms.

In this section, we describe the Condor system that forms the basis of the computational grid we used in our project. We then discuss the suitability of stochastic optimization for solution on grids. Finally, we outline the software tool MW that enables the implementation of master-worker algorithms on grids.

5.2.1 Condor

The Condor system [7, 24] manages distributively owned collections ("pools") of processors spread across a campus or other organization. The owner of each machine in a Condor pool specifies the conditions under which jobs of other users are allowed to run on their machine. Most owners require Condor to terminate any job it is running on their machine whenever they start to use the machine themselves. Thus, Condor intrudes minimally on the owner's use of their machine, while putting to work for other users the computational cycles that would otherwise have been wasted. Users have little to lose by donating their machines to a pool and much to gain in access to the Condor system when they want to make use of the pool for their own purposes. Organization-wide Condor pools have been assembled at a number of universities and research centers (see Table 5.2 for the locations used in one of our runs).

When a user submits a job to Condor, the system finds an available processor in the pool with the right architecture and capabilities and starts executing the job on that machine. At periodic checkpoints, Condor saves the state of the job, so that if the current host becomes unavailable (for example, if the machine is reclaimed by its owner), the job can be migrated and restarted from the latest checkpoint on a different processor in the pool. Condor-managed processes can communicate with the workstation from which they were submitted through a Condor-enabled version of the popular message passing library Parallel Virtual Machines (PVM) [12] or by writing and reading shared files. The submitting workstation can submit more than one job to the Condor pool at a time—a fact which makes it possible to perform parallel computing on this platform.

Parallel algorithms with the following characteristics are in principle well suited for execution on the Condor system:

- *Algorithms that can be expressed in the master-worker framework.* The Condor setup supports this framework naturally by executing the master process on the submitting workstation and the worker processes on different hosts in the Condor pool. Workers communicate with the master process via the mechanisms mentioned above.

- *Algorithms that are compute-intensive rather than data-intensive.* Computations of the latter type need to communicate large quantities of data between processors and may be affected seriously if the communications properties of the platform are poor.

- *Algorithms with weak synchronicity requirements.* Algorithms that require strict coordination between the different parts of the overall computation may run inefficiently when the platform is highly heterogeneous or has poor latency properties or if the worker processors are constantly being reclaimed by their owners. In such algorithms, the entire computation can be suspended while waiting for a single slow or suspended worker processor to complete its task.

- *Algorithms in which the size of the computational task can be varied.* In these algorithms, task sizes can be adapted to the capabilities of the machines in the pool and of the communication networks.

- *Algorithms for problems that are large and significant enough to justify the programming and debugging effort.* Although significant advances have been made in

tools to facilitate programming for complex applications in this environment (see section 5.2.3), the effort required to implement algorithms on grid computing platforms is certainly not trivial at present.

Although the master-worker model may seem restrictive, we have found that a surprising range of algorithms in optimization can be expressed in terms of this model. For example, branch-and-bound algorithms can quite naturally be implemented as master-worker algorithms by using the master process to store global information and manage the branch-and-bound tree and using worker processes to solve individual nodes or subtrees of relaxed problems.

We are convinced that inexpensive, powerful platforms such as those managed by the Condor system can be put to highly effective use by many optimization researchers and practitioners. In an academic setting, this platform has already been used to solve large instances of the quadratic assignment problem [1], linear integer programming [6], and nonlinear integer programming [15], as well as the stochastic programming problems with recourse discussed in this report. In an industrial setting we believe that the use of idle time on office workstations and PCs to solve complex problems associated with scheduling and logistics is an appealing alternative to the purchase and operation of a dedicated computing server.

5.2.2 Stochastic programming on grids

The platform outlined above is well suited to solving stochastic programming problems of various types. In the next section, we discuss one particular problem—two-stage stochastic linear programming with recourse over a finite set of scenarios. More generally, however, we can note the following relevant properties of stochastic optimization problems:

- Stochastic problems are typically compute-intensive rather than data-intensive. The often huge set of scenarios usually can be expressed in a compact fashion (for example, by listing the possible values of the independent random variables in a problem and their associated probabilities). Problems with integer variables are especially compute-intensive.

- Stochastic programming algorithms often have weak synchronicity requirements. For instance, in sampling-based methods [26, 35], the different samples can be evaluated independently with no synchronization—an example of "embarrassingly parallel" or "pleasantly parallel" computation. In the L-shaped algorithm for stochastic programs with recourse, second-stage problems associated with the different scenarios can be solved independently, although synchronicity enters the algorithm in that the master processor waits for all scenarios to be evaluated before solving a new master problem. (We discuss in a later section how this requirement can be relaxed.)

- Stochastic programming algorithms can easily vary the size of their computational tasks. One can partition by scenarios, defining each task to be a cluster of scenarios, and vary the task size by varying the number of scenarios in the task. The fact that we often can predict the time required to perform each task with some accuracy makes it easier to schedule the application efficiently on a parallel platform. Load

balancing is more difficult in most other areas of optimization, for example, in integer programming, where it is difficult to predict the time required to process each subtree of the branch-and-bound tree [25].

- Stochastic programming algorithms can benefit from the computational power offered by a grid platform. For problems with continuous variables, the number of scenarios that we can handle grows roughly in proportion to the computing power available. The ability to solve problems with more scenarios, and to solve multiple sampled instances of the same problem, can significantly improve the quality of the computed solution.

5.2.3 MW: A tool for implementing master-worker algorithms on grids

MW [14] is a runtime support library that facilitates implementation of master-worker algorithms on computational grids. The application programming interface takes the form of a set of three C++ abstract classes, containing ten methods, that must be reimplemented by the applications programmer to fit their particular application. Users are of course free to define additional methods as needed for their algorithms. In section 5.3.3, we outline the reimplementation of these classes in the case of our asynchronous algorithms for two-stage stochastic linear programming with recourse. The three classes are as follows:

- *MWDriver*. The MWDriver class is responsible for initializing the algorithm and for performing the actions required by the application whenever one of the worker processors completes its assigned computational task. The base class contains additional functionality whose complexity is hidden from the user, including the handling of worker processes that join and leave the computation, assignment of tasks to appropriate workers, rescheduling of tasks when their host workers disappear without warning, and keeping track of performance data for the run.

- *MWTask*. The MWTask class is the abstraction of a single computational task. It holds both the data describing that task and the results obtained by executing the task. The user must reimplement methods for packing and unpacking the task data into simple data structures to be passed between master and workers, using communication primitives appropriate to the particular grid platform. (When using Condor's implementation of PVM, for instance, PVM communication primitives are used.) Similar routines for packing and unpacking the results of the task are also required.

- *MWWorker*. The MWWorker class is the core of the executable that runs on each worker. After initializing itself, using information passed from the master, the worker process sits in a loop, waiting for tasks to be sent to it. When a task is received, an MWWorker method executes the task and then invokes an MWTask method to pass results back to the master. The worker process then returns to its wait loop.

Besides its interface to the application, MW also has an abstract *infrastructure programming interface* (IPI) that allows it to be implemented on different grid platforms. To implement MW on their grid, a grid programmer must reimplement a few functions that facilitate communications between processors and management of computational resources.

To date, there are four concrete implementations of the MW IPI. Three of these implementations use Condor to manage the computational resource pool but differ in the manner in which they pass messages between master and workers; they use PVM, shared files, and UNIX sockets, respectively. The fourth implementation runs both master and worker as a single process and is used for debugging. (These implementations are described further by [14].)

5.3 Decomposition methods for stochastic programming on grids

We now discuss algorithms for two-stage stochastic linear programming with recourse and their implementation on a grid platform based on Condor and MW. A more complete discussion of these algorithms, their implementations, and computational results is given by [23].

Our target problem has the form

$$\min_x \ \mathcal{Q}(x) \stackrel{\text{def}}{=} c^T x + \sum_{i=1}^{N} p_i \mathcal{Q}_i(x) \ \text{ subject to } Ax = b, \ x \geq 0, \tag{5.1}$$

where each $\mathcal{Q}_i(x)$ is the value function for a second-stage linear program:

$$\mathcal{Q}_i(x) \stackrel{\text{def}}{=} \min_{y(\omega_i)} \ q(\omega_i)^T y(\omega_i) \ \text{ subject to} \tag{5.2a}$$

$$W y(\omega_i) = h(\omega_i) - T(\omega_i)x, \quad y(\omega_i) \geq 0. \tag{5.2b}$$

Each ω_i, $i = 1, 2, \ldots, N$, represents a scenario index, possibly obtained by sampling from a large (possibly infinite) set of scenarios. The function $\mathcal{Q}$ is convex and piecewise linear. We assume for purposes of this discussion that $\mathcal{Q}$ is defined for all $x \geq 0$ satisfying $Ax = b$; that is, our problem has relatively complete recourse. The algorithms use subgradient information about $\mathcal{Q}$ to construct a lower-bounding model function which becomes the basis for choosing a succession of iterates x^k, $k = 0, 1, 2, \ldots$, of the first-stage variables. These iterates converge to the solution set for the problem (5.1).

We denote the subdifferentials of $\mathcal{Q}$ and its component functions $\mathcal{Q}_i$ by $\partial\mathcal{Q}$ and $\partial\mathcal{Q}_i$, respectively. If $\pi(\omega_i)$ is a vector of Lagrange multipliers for the constraints $Wy(\omega_i) = h(\omega_i) - T(\omega_i)x$ in (5.2), then we have

$$-T(\omega_i)^T \pi(\omega_i) \in \partial\mathcal{Q}_i(x).$$

Provided that x lies in the domain of every $\mathcal{Q}_i$, it follows from (5.1) that

$$\partial\mathcal{Q}(x) = c + \sum_{i=1}^{N} p_i \partial\mathcal{Q}_i(x).$$

Therefore, we can both evaluate the function $\mathcal{Q}(x)$ and generate subgradient information by finding a primal-dual solution of each second-stage problem (5.2).

5.3.1 Parallel L-shaped algorithm

The first algorithm we implemented for solving (5.1) is a multicut version of the L-shaped algorithm of [37]; see [5]. It works by partitioning the N scenarios of (5.1) into T clusters $\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_T$ and by using function and subgradient information for each partial sum defined by

$$\mathcal{Q}_{[j]}(x) = \sum_{i \in \mathcal{N}_j} p_i \mathcal{Q}_i(x), \quad j = 1, 2, \ldots, T,$$

to define a lower-bounding piecewise-linear approximation $m_{[j]}$ to $\mathcal{Q}_{[j]}$. At the kth iteration, we solve the following problem to obtain a new iterate x^{k+1}:

$$\min_x m_k(x) \text{ subject to } Ax = b, \ x \geq 0, \tag{5.3}$$

where

$$m_k(x) \stackrel{\text{def}}{=} c^T x + \sum_{j=1}^{T} m_{[j]}^k(x).$$

Here, $m_{[j]}^k$ denotes the model function for cluster j after subgradient information from the first k iterates $x^0, x^1, \ldots, x^k$ has been incorporated. The problem (5.3) can be formulated as a linear program whose optimal value provides a lower bound on the solution of (5.1).

Even if we define a computational task to be the evaluation of scenarios in just one cluster, a maximum of T workers can be utilized at any one time. The master processor stores all the subgradient information needed to define the problem (5.3) and solves the resulting linear program to obtain a new iterate x^{k+1} when evaluation of the function and subgradient for (5.3) has been completed for $x = x^k$.

This basic approach has the disadvantage of being too synchronous for a grid platform. If one of the tasks is delayed or lost, the master process and the entire worker pool are left without useful work to do. Ultimately, Condor and MW will reschedule the task on another worker, but the delay may seriously affect the performance of the algorithm.

Fortunately, it is easy to modify the L-shaped approach to reduce its synchronicity. The master simply waits until some fraction $\sigma \in (0, 1]$ of the evaluation tasks at x^k are completed before re-solving the master problem (5.3) (using all the current information in the model) to obtain x^{k+1}. In this asynchronous variant, the algorithm is not seriously delayed by a few tardy tasks. For values of σ nearer zero, the list of tasks waiting to be processed at any given time may include tasks from a number of past iterates, and the number of tasks being processed at any given time may be significantly greater than T. Hence, the number of workers that can be put to work on the problem is potentially greater than in the basic algorithm. However, the algorithm typically requires more iterations to converge to a solution of (5.1) as we decrease σ. This phenomenon is not surprising, since we are evaluating x^{k+1} using less information about the function $\mathcal{Q}$ than is available at the corresponding iteration of the synchronous algorithm ($\sigma = 1$). Hence, there is a trade-off between total workload and improved asynchronicity or parallelism. In our tests, we tried values of the synchronicity parameter σ between .25 and .9, and found that values of σ in the range [.75, .9] appeared to yield the best wall clock times on a typical grid configuration.

5.3.2 Parallel trust-region algorithm

Both the practical performance of the L-shaped method and its theoretical properties leave something to be desired. The L-shaped method tends to take large steps on early iterations, when the model function m_k is only a crude approximation to $\mathcal{Q}$. Further, although one can devise heuristics for deleting uninteresting subgradient information from the model (to prevent the number of constraints in the equivalent linear program for (5.3) becoming excessively large), there is no theoretically reliable way to do so.

Regularization by means of a trust region can be used to improve the properties of the method. A trust-region algorithm adds the following basic features to the L-shaped approach:

- In solving for the new candidate iterate x, we add the trust-region constraint

$$\|x - x^k\|_\infty \leq \Delta^k, \tag{5.4}$$

  where Δ^k is the trust-region radius at iteration k, to problem (5.3). The resulting problem still can be formulated as a linear program.

- The point obtained by solving (5.3), (5.4) is not accepted as the new iterate unless it yields a "sufficient decrease" in the value of $\mathcal{Q}$ over the current point x^k. If it fails to do so, the trust-region radius may be reduced and a new subproblem of the form (5.3), (5.4) solved. Note that we do not necessarily reduce the trust region radius; even without changing the trust region, the additional function and subgradient information obtained at the failed candidate may enhance the model sufficiently to produce a successful step.

- After stepping to a new iterate, the algorithm may delete subgradient information generated earlier in the algorithm. In particular, subgradients that have been inactive for a number of successive iterations may be removed.

The sequence of iterates generated by this trust-region algorithm can be shown to converge to the solution set $\mathcal{S}$ of (5.1) if $\mathcal{S}$ is nonempty; see [23, Theorem 2]. The approach has the advantage that if started from a point x^0 close to $\mathcal{S}$ (a situation that frequently arises in our tests), the trust region ensures that the initial iterates stay close to this point. In fact, the algorithm does not step away from x^0 at all until the model m_k becomes good enough to generate a point with a significantly better value of $\mathcal{Q}$. The overall algorithm is related to those proposed by [30, 21] and [17, Chapter XV], who use quadratic penalty terms to restrict the distance from x^k to the next candidate iterate.

Parallel implementation of the trust-region algorithm on the grid is similar to implementation of the synchronous L-shaped method, and the main drawback is the same: a delay in the evaluation of one task can cause the workers and master to be left idle for long periods. However, devising an asynchronous variant of the trust-region approach that retains the appealing theoretical properties of the synchronous variant is more difficult than for the L-shaped method. Our asynchronous trust-region (ATR) method maintains an incumbent point x^l and a basket $\mathcal{B}$ containing a set of candidate points for which function and subgradient information is currently being calculated by the workers. When evaluation of one of the points in $\mathcal{B}$ is completed, a "sufficient decrease" test is applied to determine whether

this point should replace x^I as the incumbent. Whether this happens or not, the trust-region radius may be adjusted, and cuts may be deleted from the model. A trust-region subproblem similar to (5.3), (5.4) is solved to find a new candidate point. The model function used in the trust-region subproblem uses whatever subgradient information is currently available, including information from partially completed evaluations of points in the basket $\mathcal{B}$. The trust region is centered at the incumbent x^I, rather than the latest iterate x^k as in (5.4).

The basket is filled initially by solving the trust-region subproblem after a certain fraction $\sigma \in (0, 1]$ of the evaluation tasks for the first few iterates is completed. However, during most of the execution, the basket is in steady state, with each completed evaluation resulting in a point leaving the basket and being replaced by a new candidate obtained by solving a trust-region subproblem.

Techniques for adjusting the trust-region radius and managing the subgradient information in the model are closely related to those used in the synchronous case. The theoretical convergence results are also similar. Under the assumption that all evaluation tasks complete in finite time, the sequence of incumbents x^I converges to the optimal set $\mathcal{S}$ (see [23, Theorem 4]).

The ATR algorithm clearly has greater potential for parallel implementation than its synchronous trust-region cousin. A larger number of tasks are available for evaluation at any given time. It also is less synchronous in that a delay in processing one task holds up the evaluation of one point in the basket $\mathcal{B}$ but not the entire algorithm. Not unless nearly all points in $\mathcal{B}$ are blocked in this way do we expect serious degradation in parallel performance. On the other hand, as in the asynchronous L-shaped approach, ATR typically performs more total computation than the synchronous trust-region algorithm; the total number of iterates (that is, the number of times the trust-region subproblem is solved) grows with basket size. Once again, there is a trade-off between total work, on one hand, and improved parallelism and asynchronicity, on the other.

5.3.3 MW implementations

In this section, we elaborate on our discussion of the MW runtime support library in section 5.2.3. We describe how each of the key methods in MW is implemented in the case of the ATR application of section 5.3.2.

MWDriver.

- `get_userinfo`. This function, which is executed at the start of ATR, reads a command file to set parameters such as convergence tolerances, number of scenarios, number of partial sums to be evaluated in each task, maximum number of worker processors to be requested, and initial trust-region radius. It calls the routines that read and store the problem data files, and reads the initial point, if one is supplied by the user. This routine also sets important algorithmic parameters. In particular, it decides how many of the clusters defined in (5.3) to place in one computational task, to be assigned subsequently to a worker. Each task is chosen large enough to ensure that it requires at least 5 seconds to execute on the fastest processor in the pool. (In our largest examples, considerably larger tasks were used.)

- `setup_initial_tasks`. This populates the task pool with the tasks for evaluating the objective at the initial point x^0.

- `pack_worker_init_data`. This sends the information in the input files for the stochastic programming problem to each new worker as it joins the pool. When the worker subsequently receives a task, it uses the original input data to generate the particular second-stage data for its assigned set of scenarios.

- `act_on_completed_task`. When a worker completes its task, this routine adds the partial sums $\mathcal{Q}_{[j]}$ evaluated by this task to the objective function and places its subgradient information in a buffer. When this task is the one that completes the evaluation of the objective $\mathcal{Q}(x)$ at some point x, the method decides whether to accept x as the new incumbent, performs any necessary adjustments to the trust-region radius, adds cuts from the current buffer into the model, possibly deletes obsolete cuts, solves the trust-region subproblem to generate a new candidate point, and adds the evaluation tasks for the new candidate to the list of tasks.

MWTask. In our applications, each task evaluates the partial sum $\mathcal{Q}_{[j]}(x)$ and a subgradient for a certain number of clusters. The task is described compactly by a range of scenario indices for each of its clusters and by the value of the first-stage variables x. The results consist of the value of $\mathcal{Q}_{[j]}$ and its subgradient for each cluster making up the task. The functions in this class simply pack and unpack the data and results into data structures suitable for transmission between master and worker.

MWWorker. The `execute_task` method of this class formulates the second-stage linear programs in the cluster for the given task, using the task definition information received from the master. It then calls the linear programming solver SOPLEX [38] to solve these problems and uses their dual solutions to calculate the subgradients.

5.3.4 Computational results

We now present a selection of computational results obtained with the solvers described above. For a fuller picture, in particular the dependence of the results on various parameter choices, see [23]. For extensive sampling studies performed with these solvers, see [22].

Our first test problem is known as SSN, a telecommunications design application described in [33]. This problem has 89 first-stage variables and a single constraint (a budget constraint), while the second-stage problems have 706 unknowns, 175 constraints, and a constraint matrix with 2284 nonzeros. This problem contains 86 independent random variables and a total of 10^{70} scenarios. It has a reputation of being difficult, in that algorithms typically require many iterations and, even when variance reduction techniques are applied, good quality upper and lower bounds on the optimal objective value are hard to calculate [16, 26]. Computational experience with this problem has been reported with the serial Regularized Decomposition (RD) code of [32]. Mak, Morton, and Wood [26] report solving 30 sampled instances with 1000 scenarios each using RD on an IBM RS/6000 3ct (a processor similar to an RS/6000 590) with 512 MB of memory, in an average of 81 minutes

Table 5.1. *SSN trial with best parameter combinations, $N = 10{,}000$ scenarios, algorithms ALS, TR, and ATR.*

run	points evaluated	size of $\mathcal{B}$	# tasks	# clusters	max. processors allowed	av. processors	parallel efficiency	max. # cuts in model	master problem solve time (min)	wall clock time (min)
ALS	282	-	50	50	100	26	.81	5562	24	254
TR	46	-	25	100	25	22	.43	4032	1	56
TR	43	-	50	100	50	33	.48	3922	1	35
ATR	96	3	25	100	50	37	.55	6682	2	47
ATR	100	3	50	100	100	41	.64	8083	2	30
ATR	171	6	25	100	87	56	.80	7653	5	37
ATR	170	6	50	100	175	44	.90	9143	5	36

per instance. In 1996, the authors of the RD code reported solution times of 64 minutes for an instance with 500 scenarios, and 163 minutes for an instance with 1000 scenarios, and on a single processor of a Cray 6400 SuperServer, which is roughly equivalent in speed to a SparcStation 5. More recently, we obtained an execution time for RD of 220 minutes on a Sun UltraSparc-2 with 128MB of memory for an instance with 5000 scenarios. (This version of RD has an upper limit of 5000 on the number of scenarios.)

Table 5.1 shows results obtained from a sampled instance of SSN with $N = 10{,}000$ scenarios, which can be formulated as a linear program with approximately 1.75×10^6 constraints and 7.06×10^6 variables. The table shows results from the asynchronous version of the L-shaped method (ALS), the trust-region method (TR), and the ATR method with various choices of basket size ($|\mathcal{B}|$) and varying numbers of tasks and clusters. The second column tabulates the number of points at which the function $\mathcal{Q}$ was evaluated, and the last column shows the total wall clock time for the jobs. The number of workers requested from the pool, the average number of workers used during the computation, and the parallel efficiency (the proportion of time for which the owned workers were kept busy) are also shown. The final columns show the maximum number of cuts (subgradients) stored in the master-problem model during the computation and the total time spent solving problem (5.3), (5.4) on the master processor.

The table shows that the ALS code is generally slower than the trust-region variants, which all perform fairly well for the choices of parameters shown here. In general, the performance depends somewhat on conditions on the Condor pool at the time the job is run. The priority assigned to the job affects the number of workers it can obtain, and the overall run time can be seriously affected if workers are suspended repeatedly. Poor performance of the synchronous trust-region algorithm has been observed on some runs, when two or three workers are significantly slower than the others in the pool.

To demonstrate the scale of problem that can be solved on a computational grid plat-

Table 5.2. *Machines available for storm problem, with $N = 10^7$ scenarios.*

Number	Type	Location
184	Intel/Linux	Argonne
254	Intel/Linux	New Mexico
36	Intel/Linux	NCSA
265	Intel/Linux	Wisconsin
88	Intel/Solaris	Wisconsin
239	Sun/Solaris	Wisconsin
124	Intel/Linux	Georgia Tech
90	Intel/Solaris	Georgia Tech
13	Sun/Solaris	Georgia Tech
9	Intel/Linux	Columbia U.
10	Sun/Solaris	Columbia U.
33	Intel/Linux	Italy (INFN)
1345		**TOTAL**

form, we solved large instances of a cargo flight scheduling problem known as "storm." This problem was described by Mulvey and Ruszczyński [27], who also reported considerable computational testing with the algorithm they proposed in their paper, along with several other codes and an earlier implementation of the regularized decomposition algorithm of [30]. In experiments performed in 1992 and 1993, the latter was found to perform best; a 1000-scenario instance was solved in about 6 hours of CPU time on a SUN SparcStation 2. (The other codes were not able to handle more than 200 scenarios in a reasonable amount of time.) Recently, we tested the current implementation of RD on a Sun UltraSparc-2 and solved a 1000-scenario instance in about 20 minutes (45 iterations) and a 5000-scenario instance in about 260 minutes (33 iterations).

Linderoth and Wright [23] report solving a sampled instance of storm with $N = 250{,}000$ in 116 minutes using ATR. The average size of the worker pool during the run (drawn from the Condor pool at Wisconsin) was 106. In our largest run, we solved an instance of storm with $N = 10^7$ scenarios, for which the equivalent linear program has approximately 10^9 constraints and 10^{10} variables. A starting point was calculated from a smaller sampled instance. For this run, we used ATR with a basket size of 4 and executed on a pool of more than 1300 workers from seven different institutions running three different operating systems (see Table 5.2). The job ran for a total of almost 32 hours; the number of machines in use over the course of the run is graphed in Figure 5.1. The master process crashed after 8 hours of execution, but Condor's checkpointing mechanism allowed it to be restarted 2 hours later and then run an additional 24 hours to completion.

The run is profiled in Table 5.3. An average of 433 workers were present during the run, and 556 workers were used altogether. A total of 40,837 tasks were generated during the run, representing 3.99×10^8 second-stage linear programs. (At this rate, an average 3472 second-stage linear programs were being solved per second during the run.) The average time to solve a task was 774 seconds. The total cumulative CPU time spent by the worker

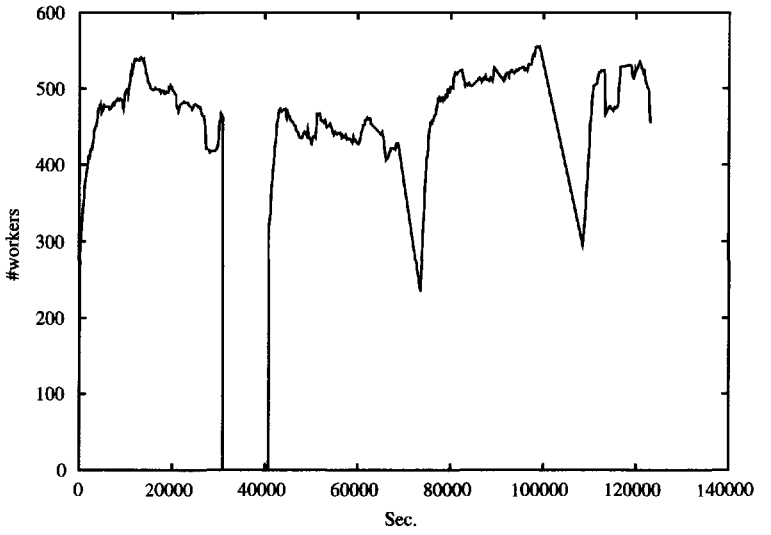


Figure 5.1. *Machines in use during execution of storm problem with $N = 10^7$ scenarios.*

Table 5.3. *Storm problem, with $N = 10^7$ scenarios.*

run	points evaluated	size of $\lvert\mathcal{B}\rvert$	# tasks	# clusters	max. processors allowed	av. processors	parallel efficiency	max. # cuts in model	master problem solve time (hr)	wall clock time (hr)
ATR	38	4	1024	1024	800	433	.668	39647	1.9	31.9

pool was 9014 hours, or slightly more than one year.

5.4 Future directions

Our goal has been to introduce the stochastic programming community to a platform that seems well suited to the needs of algorithms for solving large practical instances of problems in the area. In this section we outline some future plans for algorithms and software for other stochastic problems and relevant plans for the underlying grid environment.

We plan to improve the ATR code further by improving its algorithmic features, possibly by using a Euclidean-norm trust region (which necessitates solving a quadratic program at each iteration) and by improving the management of the subgradient buffer. Other improvements will come from the use of performance modeling techniques to better understand the behavior of the ATR code in dynamic, heterogeneous computational environments. At

present, our understanding of the effect on performance of different choices of the algorithmic parameters (such as basket size and numbers of chunks and tasks) is limited, especially when the pool contains machines of very different speeds. We are working with experts in performance modeling [29] to understand these issues better. Ultimately, we hope to use performance models within the ATR code, enabling it to adapt its behavior to changing pool conditions during a run. Adaptations that may be made include changing the basket size, changing the number of tasks or clusters, removing an unpromising point from the basket before its evaluations are complete, and assigning different amounts of work to different workers.

Approaches that require solution of a number of independent sample-average approximations are ideal for implementation on a computational grid. The jobs can be run independently, possibly from different masters to avoid interference with each other. Batch approaches such as this are useful in computing estimates of the upper and lower bounds on the objective value. See [26] for verification experiments that rely on this kind of computation. Linderoth, Shapiro, and Wright [22] describe a large set of empirical experiments carried out with ATR.

Many of our planned adaptations to the algorithms will require enhancements to the capabilities of MW. Future versions of MW will include improved performance information that can be exploited by the performance modeling techniques described above, such as better measurement of task execution times, communication times, times required to obtain new workers, and worker suspension rates. Improved logging tools will enable us to improve performance and eliminate sources of inefficiency in the algorithms. Other planned features for MW will allow the application code greater control over its worker pool, including the ability to remove a worker, to interrupt or terminate execution of tasks on specific workers, and to dynamically change the number of workers requested during the computation. These enhancements to MW capabilities will not only benefit our stochastic programming application but also be useful in parallel algorithms for other compute-intensive numerical applications.

Longer-term goals involve stochastic integer programming. The complexity of parallel algorithms for this problem and their high computational demands will require much larger worker pools and new ways to manage these pools. We anticipate extending MW to support a hierarchical master-worker framework in which a second layer of master processors will direct computations on subtrees of the branching tree. Stochastic integer programming is recognized as a difficult class of problems (see [4]), and relatively little research on algorithms has been performed to date, especially not on general-purpose algorithms (see the bibliography maintained by [36]). We believe that computational grids of the type discussed in this chapter open the door to solution of interesting problems in this area, and we hope that this possibility will spur new work on algorithm development and implementation.

Acknowledgments

We thank Alex Shapiro for interesting discussions and advice over the course of this project. We also thank other members of the metaNEOS, Condor, and POEMS teams for their ongoing work on the grid platform described here and for their continuing inspiration and advice.

Bibliography

[1] K. Anstreicher, N. Brixius, J.-P. Goux, and J. T. Linderoth, *Solving large quadratic assignment problems on computational grids*, Math. Program., 91 (2002), pp. 563–588.

[2] K. A. Ariyawansa and D. D. Hudson, *Performance of a benchmark parallel implementation of the Van Slyke and Wets algorithm for two-stage stochastic programs on the Sequent/Balance*, Concurrency Practice Experience, 3 (1991), pp. 109–128.

[3] J. R. Birge, C. J. Donohue, D. F. Holmes, and O. G. Svintsitski, *A parallel implementation of the nested decomposition algorithm for multistage stochastic linear programs*, Math. Program., 75 (1996), pp. 327–352.

[4] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer Ser. Oper. Res., Springer-Verlag, New York, 1997.

[5] J. R. Birge and F. V. Louveaux, *A multicut algorithm for two-stage linear programs*, European J. Oper. Res., 34 (1988), pp. 384–392.

[6] Q. Chen, M. C. Ferris, and J. T. Linderoth, *Fatcop* 2.0: *Advanced features in an opportunistic mixed integer programming solver*, Ann. Oper. Res., 103 (2001), pp. 17–32.

[7] *Condor*, 2002; available online from http://www.cs.wisc.edu/condor.

[8] G. Dantzig, J. Ho, and G. Infanger, *Solving Stochastic Linear Programs on a Hypercube Multicomputer*, Technical Report SOL 91-10, Department of Operations Research, Stanford University, Stanford, CA, 1991.

[9] *Entropia*, 2001; available online from http://www.entropia.com/.

[10] I. Foster and C. Kesselman, eds., *The Grid: Blueprint for a New Computing Infrastructure*, Morgan-Kaufmann, San Francisco, 1998.

[11] E. Fragniere, J. Gondzio, and J.-P. Vial, *Building and solving large-scale stochastic programs on an affordable distributed computing system*, Ann. Oper. Res., 99 (2000), pp. 167–187.

[12] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam, *PVM: Parallel Virtual Machine*, MIT Press, Cambridge, MA, 1994.

[13] J. Gondzio and R. Kouwenberg, *High performace computing for asset liability management*, Oper. Res., 49 (1991), pp. 879–891.

[14] J.-P. Goux, S. Kulkarni, J. T. Linderoth, and M. E. Yoder, *Master-worker: An enabling framework for applications on the computational grid*, Cluster Comput., 4 (2001), pp. 63–70.

[15] J.-P. Goux and S. Leyffer, *Solving Large MINLPs on Computational Grids*, Numerical Analysis Report NA/200, Mathematics Department, University of Dundee, Dundee, Scotland, 2001.

[16] J. L. HIGLE, *Variance reduction and objective function evaluation in stochastic linear programs*, INFORMS J. Comput., 10 (1998), pp. 236–247.

[17] J. HIRIART-URRUTY AND C. LEMARÉCHAL, *Convex Analysis and Minimization Algorithms* II, Comp. Stud. Math., Springer-Verlag, New York, 1993.

[18] E. R. JESSUP, D. YANG, AND S. A. ZENIOS, *Parallel factorization of structured matrices arising in stochastic programming*, SIAM J. Optim., 4 (1994), pp. 833–846.

[19] A. J. KING, *SP/OSL V1.0, Stochastic Programming Interface Library User's Guide*, Technical Report, Research Division, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1994.

[20] A. J. KING AND S. E. WRIGHT, *A Flexible-Partition, Nested L-Shaped Method for Linear Programming*, Technical Report, Research Division, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1999.

[21] K. KIWIEL, *Proximity control in bundle methods for convex nondifferentiable minimization*, Math. Program., 46 (1990), pp. 105–122.

[22] J. T. LINDEROTH, A. SHAPIRO, AND S. J. WRIGHT, *The Empirical Behavior of Sampling Methods for Stochastic Programming*, Optimization Technical Report 02-01, Computer Sciences Department, University of Wisconsin-Madison, Madison, WI, 2002

[23] J. T. LINDEROTH AND S. J. WRIGHT, *Decomposition algorithms for stochastic programming on a computational grid*, Comput. Optim. Appl., 24 (2003), pp. 207–250.

[24] M. LIVNY, J. BASNEY, R. RAMAN, AND T. TANNENBAUM, *Mechanisms for high throughput computing*, SPEEDUP, 11 (1997); also available online from http://www.cs.wisc.edu/condor/doc/htc_mech.ps.

[25] R. LÜLING AND B. MONIEN, *Load balancing for distributed branch and bound algorithms*, in Proceedings of the International Parallel Processing Symposium, Beverly Hills, CA, 1992, pp. 543–549.

[26] W. K. MAK, D. P. MORTON, AND R. K. WOOD, *Monte Carlo bounding techniques for determining solution quality in stochastic programs*, Oper. Res. Lett., 24 (1999), pp. 47–56.

[27] J. M. MULVEY AND A. RUSZCZYŃSKI, *A new scenario decomposition method for large scale stochastic optimization*, Oper. Res., 43 (1995), pp. 477–490.

[28] S. S. NIELSEN AND S. A. ZENIOS, *Scalable parallel Benders decomposition for stochastic linear programming*, Parallel Comput., 23 (1997), pp. 1069–1089.

[29] *POEMS*, 2002; available online from http://www.cs.utexas.edu/users/poems/.

[30] A. RUSZCZYŃSKI, *A regularized decomposition method for minimizing a sum of polyhedral functions*, Math. Program., 35 (1986), pp. 309–333.

[31] A. RUSZCZYŃSKI, *Parallel decomposition of multistage stochastic programming problems*, Math. Program., 58 (1993), pp. 201–228.

[32] A. RUSZCZYŃSKI AND A. ŚWIETANOWSKI, *On the Regularized Decomposition Method for Two-Stage Stochastic Linear Problems*, Working Paper WP-96-014, IIASA, Laxenburg, Austria, 1996.

[33] S. SEN, R. D. DOVERSPIKE, AND S. COSARES, *Network planning with random demand*, Telecommunications Syst., 3 (1994), pp. 11–30.

[34] *SETI@home*, 2001; available online from http://setiathome.ssl.berkeley.edu/.

[35] A. SHAPIRO, *Stochastic Programming by Monte Carlo Simulation Methods*, Technical Report, School of ISYE, Georgia Institute of Technology, Atlanta, GA, 1999.

[36] M. H. VAN DER VLERK, *Stochastic Integer Programming Bibliography*, 1996-2002; available online from http://mally.eco.rug.nl/biblio/SIP.HTML.

[37] R. M. VAN SLYKE AND R. WETS, *L-shaped linear programs with applications to control and stochastic programming*, SIAM J. Appl. Math., 17 (1969), pp. 638–663.

[38] R. WUNDERLING, *Paralleler und Objektorientierter Simplex-Algorithmus*, Ph.D. thesis, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 1996; also available online from http://www.zib.de/Optimization/Software/Soplex/soplex.php.

This page intentionally left blank

Chapter 6

Building and Solving Stochastic Linear Programming Models with SLP-IOR

Peter Kall and János Mayer**

6.1 Introduction

This chapter describes the capabilities and the usage of SLP-IOR, our interactive model management system for stochastic linear programming (SLP). The main features of SLP-IOR are the following: the system is intended to support the entire life cycle of a model, including model formulation, analysis of the model instance, solving it, and analyzing the solution. A main design characteristic is keeping connection to an algebraic modeling system; we have chosen GAMS [3, 4]. This approach has the following advantages: on the one hand, the powerful general-purpose solvers connected to GAMS are available for solving deterministic equivalents of SLP problems; on the other hand, deterministic linear programs (LPs) formulated in the algebraic modeling language of GAMS can be imported into SLP-IOR to develop stochastic variants of these. However, the use of GAMS is optional; with the exception of the above-mentioned GAMS-related features, SLP-IOR can be fully utilized without having access to GAMS.

From the functional point of view, SLP-IOR integrates three main groups of facilities. The core part serves for dealing with SLP model instances. The workbench component supports working with test problem batteries and performing test runs with them. The third component consists of a solver library organized as a solver description database and a collection of executables of solvers. Besides some general-purpose LP solvers, the solver library contains several solvers specialized to the various SLP model classes.

The system runs under Windows 32. SLP-IOR itself has been developed in Borland Delphi 6 in an object-oriented style. The majority of our own solvers have been developed in Fortran using Compaq Visual Fortran 6.1.

*Institute for Operations Research, University of Zurich, CH-8044 Zurich, Switzerland (kall@ior.unizh.ch, mayer@ior.unizh.ch)

SLP-IOR is available free of charge for academic purposes. File transfer of the system and the user's guide can be arranged following e-mail contact with the authors.

For the model management aspects of SLP in general and their implementation in SLP-IOR in particular, see Kall and Mayer [17, 18, 19, 20, 21]. Alternative modeling systems for SLP are SPInE [32, 38], the optimization system and library SP/OSL of IBM [26], and STOCHGEN [6].

For extensions of algebraic modeling languages concerning SLP models see [12] and [11] as well as [32] and [38]. For linking SLP solvers to algebraic modeling languages the system SETSTOCH [5] has been developed.

This chapter is organized as follows. In the next section we describe the scope of SLP-IOR from the modeling point of view. Section 6.3 is devoted to the model formulation phase and summarizes the facilities which are available for setting up a model instance. Section 6.4 gives an overview on the facilities which support the model manipulation and analysis phase. Section 6.5 describes the scope of SLP-IOR from the point of view of current solver availability and the scope of those solvers. Section 6.7 gives practical guidelines for using SLP-IOR to model and solve SLP problems. In this section we also discuss the present limitations of the system. The final section provides an outlook on further developments for SLP-IOR which are planned in the short run.

6.2 The scope of SLP-IOR

In this section we give a summary of the type of models which can be built and manipulated by using SLP-IOR. The following types of SLP problems are included:

- two-stage recourse problems with the subclasses fixed recourse, complete recourse, and simple recourse,
- two-stage simple integer recourse problems,
- two-stage multiple simple recourse problems,
- jointly chance-constrained problems (SLP problems with a joint probabilistic constraint), and
- SLP problems with separate chance constraints (probabilistic constraints).

Subclasses of recourse models are listed (and handled) as separate objects. This is an important issue, because for a subclass frequently much more powerful solvers are available than for the general case; see Kall and Mayer [22, 23].

Multiple simple recourse models are like simple recourse models except that the objective in the second stage is a piecewise linear convex function. See Ziemba [40] and Haneveld [27].

In addition to the model types listed above, multistage recourse problems with scenarios are also implemented.

The random entries of the various arrays in the SLP problems are modeled via affine linear relations of random variables; see [25]. We illustrate this point by considering just

the recourse constraint of a two-stage recourse problem. In [1, section 3.1] this restriction has the form

$$T(\omega)x + Wy = h(\omega), \tag{6.1}$$

and the primary random vector is $\xi(\omega)^{\mathrm{T}} = (h(\omega)^{\mathrm{T}}, T_1(\omega), \ldots, T_{m_2}(\omega))$, with $T_i(\omega)$ being the ith row of the technology matrix $T(\omega)$. This means that the primary random variables, the distribution of which should be specified by the user, are the random entries of the arrays themselves. In SLP-IOR the following more general form has been implemented (see [25, section 3.1]): we consider random variables as separate entities, and the connection to the random array entries in the model is established via affine linear relations,

$$\begin{aligned} T(\omega) &= T^0 + \sum_{k=1}^{r} T^k \xi_k(\omega), \\ h(\omega) &= h^0 + \sum_{k=1}^{r} h^k \xi_k(\omega), \end{aligned} \tag{6.2}$$

where the user has to specify the probability distribution of the r-dimensional random vector $\xi(\omega)$.

The direct entry case is clearly a special case of our approach. To see this let us assume that the association $T_{11}(\omega) = \xi_1(\omega)$ is to be established. The following setting accomplishes it: $T^0_{11} = 0$, $T^1_{11} = 1$, $T^1_{ij} = 0$ for all $(i, j) \neq (1, 1)$, $T^k_{11} = 0$ for all $k > 1$, $h^1 = 0$, where T^k_{ij} refers to the (i, j) entry of the matrix T^k.

The facility based on the affine linear relations (6.2) additionally allows for modeling situations where the random entries can be expressed by affine sums in terms of many fewer basic random variables ("factors").

For the random variables the following probability distributions are available (see, e.g., [8]):

- *Univariate discrete distributions:* empirical, uniform, binomial, hypergeometric, geometric, negative binomial, and Poisson distributions;
- *Univariate continuous distributions*: uniform, normal, exponential, gamma, beta, Cauchy, Weibull, chi-squared, Fisher's F, student's t, extreme value, logistic, log-normal, Pareto, power function, and triangular distributions;
- *Multivariate discrete distributions*: empirical and uniform distributions;
- *Multivariate continuous distributions*: uniform and normal distributions.

By an empirical distribution we mean a finite discrete distribution specified by its realization tableau and corresponding probabilities.

6.3 Setting up a model instance

New model instances can be set up in a menu-driven fashion. The user has to specify the type of the model, the dimensions, and the stochastic parts. By the latter we mean those arrays which contain stochastic entries.

Having done this, the probability distribution of the random vector ξ is to be specified. The components of ξ can be partitioned into mutually stochastically independent groups. The probability distributions of the groups can then be specified in turn.

As a final step the random vector is to be mapped onto the stochastic entries of the arrays. In the most general form, the terms in the affine sums (6.2) can be edited in turn. For the special case when there is a one-to-one correspondence between the random array entries and the components of ξ, a shortcut facility is provided; i.e., in this case the user does not even have to know that SLP-IOR establishes the connection internally via the affine sums (6.2).

After having set up the model structure and having defined dimensions, the entries of the various arrays (e.g., technology matrices and recourse matrices, right-hand-side vectors, objective vectors, realization tableaus, correlation matrices) can be entered via a matrix editor. This may become a cumbersome task in higher dimensions; therefore SLP-IOR provides facilities for exporting and importing whole blocks of data.

Building blocks are provided on two levels. On the top level they consist of three blocks of data corresponding to the underlying LP, to the distribution of ξ, and to the affine relations (6.2), respectively. By the underlying LP we mean the deterministic LP which arises via replacing the random entries by their expected values. On the second level building blocks correspond to the various arrays in the model. Building blocks can be saved and retrieved via files.

Arrays or parts of them are frequently primarily available in various other systems, e.g., MS Excel. SLP-IOR provides a facility for exporting and importing the arrays of the model instance via the Windows Clipboard.

Previously built model instances can be loaded from file; SLP-IOR can save and load models using an internal data format.

A model instance can also be set up by importing it either in the SMPS format or in a GAMS model file format.

The SMPS data format [2] is a widely accepted data format for recourse problems. Ongoing work addresses extensions; see [10]. For an overview see Gassmann's article in this volume (Chapter 2). A model instance in SMPS format consists of three ASCII files. The CORE file contains the underlying LP, the TIME file describes the decision stage structure, and the STOCH file specifies the random entries and their probability distribution. SLP-IOR can export and import model instances in SMPS format.

If the user has access to GAMS, deterministic LP models formulated in the GAMS modeling language and saved as a GAMS file can be imported into SLP-IOR for the sake of creating stochastic variants of them. After importing the model it is represented as a deterministic LP within SLP-IOR. Subsequently this LP has to be transformed into the chosen SLP model type (by utilizing the transform facility; see the next section). Finally, data have to be reorganized and missing data must be provided, according to the chosen SLP model type.

6.4 Model manipulation and analysis

A model instance can be transformed into an instance of another (S)LP model type or into a deterministic equivalent, provided that the latter exists; for missing data default values are

added. The resulting model instance has to be tailored afterward by specifying additional data, if needed. The transform facility of SLP-IOR serves, e.g., for the following purposes: the user may wish to formulate both a recourse model and a chance-constrained model on the same basic data set, or a deterministic equivalent can be created for the purpose of exporting it to an external solver. This facility also supports the deterministic optimizer who wishes to formulate SLP variants of her or his problem, e.g., by importing an LP model instance formulated in the GAMS language and transforming it subsequently into the desired SLP model type.

In section 6.2 we listed the probability distributions that can be chosen in SLP-IOR. On the solver side, however, available solvers can deal with only a subset of those distributions; see section 6.5. For recourse models, e.g., most solvers can deal with only finite discrete distributions. To overcome this difficulty, SLP-IOR provides a facility for discretizing the probability distribution of the model instance, thus resulting in an approximate model which can subsequently be solved.

SLP-IOR contains analysis facilities for analyzing a model instance and/or a solution. These facilities are available only for two-stage recourse problems and consist of the following items:

- computing the expected value of perfect information (EVPI) and the value of stochastic solution (VSS) (see [1, Chapter 4]);
- checking the complete recourse property and the (perhaps hidden) simple recourse structure;
- computing the recourse objective for a given first stage vector x;
- provided that all relations in the second stage are inequalities, the reliability of the first-stage solution (RFS) can be computed for fixed x. Replacing the equality in (6.1) by the inequality $\geq$, the definition is

$$\text{RFS} = \mathbf{P}\{\omega \mid T(\omega)x \geq h(\omega)\}.$$

This can be interpreted as the probability of the event that no recourse actions will be needed; see [21].

There are two options for carrying out the above computations. For finite discrete distributions the computations can be performed exactly. For continuous distributions and discrete distributions with a prohibitively large number of realizations, the computations can be carried out on a sampling basis.

6.5 The solver library of SLP-IOR

SLP problems are frequently quite hard to solve numerically; see, e.g., [1, 25, 31, 34]. Therefore it is important to include solvers which utilize the special structure originating in model type or probability distribution. For this reason we have connected several solvers to SLP-IOR. This includes solvers developed by ourselves and solvers which have been kindly provided to us by the authors. We would like to express our sincere thanks to these persons.

Below we list the solvers which are presently connected to SLP-IOR. For keeping the list of references below a reasonable bound, we provide direct references to only some of them; for the rest we indicate where the reference can be found. The years indicated in parentheses correspond to solver release dates.

- *Solvers for the deterministic equivalent LP of recourse problems with a finite discrete distribution:*
  - *General purpose LP solvers:*
    - *HiPlex*, simplex method with the Phase-I method of [29] (see also [30]), implemented by the author (1994);
    - *HOPDM*, primal-dual interior-point method of [13], implemented by the author (1996);
    - *Minos*, simplex method, Murtagh and Saunders (1995) (see [31]);
    - *OB*1, several interior-point methods, Marsten et al. (1989) (see [22]);
    - *XMP*, simplex method, Marsten (1986) (see [31]).
  - *Solvers utilizing the structure of the LP:*
    - *BPMPD*, augmented system interior-point method of [33], implemented by the author (1996);
    - *MSLiP*, nested Benders decomposition method [9], implemented by the author (1992);
    - *QDECOM*, regularized decomposition algorithm of [35], implemented by the author (1985);
    - *SHOR*2, decomposition scheme based on r-algorithm of [36], implemented by Shor and Likhovid (1998).
- *Solvers for complete recourse aiming at the original problem (without building the equivalent LP):*
  - *DAPPROX*, successive discrete approximation method. For using discrete approximation as a solution algorithm see [16]; for generalizations of the Edmundson–Madansky upper bound, based on one-dimensional interval partitions, and for the multivariate extension of this inequality in the independent case, see [15]; for the successive discrete approximation method for complete recourse including warm-start (which is crucial considering efficiency) see [24]; for discrete approximation methods in general see also [25]; the solver has been implemented by Kall and Mayer (2001);
  - *SDECOM*, stochastic decomposition method of [14], implemented by Kall and Mayer (1995).
- *Solvers for simple (continuous) recourse:*
  - *SHOR*1, decomposition scheme based on r-algorithm of [36], implemented by Shor and Likhovid (1997);

- *SRAPPROX*, successive discrete approximation method (including warm start) of [24] (see also [25]); Huang, Ziemba, and Ben-Tal [15] propose a different successive discrete approximation algorithm for simple recourse; they utilize the Edmundson–Madansky upper bound; in the Kall–Stoyan algorithm, however, a tighter upper bound is used. The method has been implemented by Kall and Mayer (1994).

- *Solver for simple integer recourse:*
  - *SIRD2SCR*, convex hull method of Klein Haneveld, Stougie, and Van der Vlerk [28], implemented by Mayer and van der Vlerk (1994).

- *Solver for multiple simple recourse:*
  - *MScr2Scr*, transformation method of Van der Vlerk [39], implemented by Mayer and van der Vlerk (2001).

- *Solvers for jointly chance-constrained problems:*
  - *PCSPIOR*, supporting hyperplane method of Szántai (see [25, 31, 34]); implemented by Mayer (1995);
  - *PROBALL*, central cutting plane method of Mayer (see [25, 31]); implemented by Mayer (2001);
  - *PROCON*, reduced gradient method of Mayer (see [25, 31, 34]); implemented by Mayer (1992).

For computing the multinormal probability distribution function and its gradient, two subroutine packages have been utilized: NORSET of [7] and PCSPNOR3 of [37]; for the algorithms implemented in the latter package see also [34].

Tables 6.1 and 6.2 show the main capabilities of the solvers presently connected to SLP-IOR. Table 6.1 summarizes the main solver characteristics. For separate chance constraints where only the right-hand side is stochastic, the problem can be transformed into an equivalent deterministic LP problem (see, e.g., [25]). Therefore any of the general-purpose LP solvers can be chosen. For this reason we have not indicated this model class in Table 6.1. Table 6.2 is a cross-reference table for solvers versus models, including the separately chance-constrained case.

For users who have a GAMS license, some of the GAMS general-purpose LP solvers (BDMLP, CONOPT, MINOS5, OSL, ZOOM) are included in the solver description database; therefore they can be used directly. The other GAMS solvers can be activated by using the solver connection utility of SLP-IOR, which will be discussed in section 6.6.2. Two SLP solvers are available with GAMS: SP/OSL and DECIS. Building a connection to SP/OSL [26] is in progress. Further information concerning the GAMS solvers is available on the GAMS website (www.gams.com).

The following abbreviations are used in Tables 6.1 and 6.2:

- *Model type*: *LP*, *GR*, *FR*, *CR*, *SR*, *SIR*, *MSR*, *JC*, *SC* mean deterministic LP, general recourse, fixed recourse, complete recourse, simple continuous recourse, simple integer recourse, multiple simple recourse, joint chance constraints, and separate chance constraints, respectively.

Table 6.1. *Main solver characteristics: model type, stochastic parts, probability distributions, version or date, availability.*

	Models	St. Parts	Distr.	Version	Availab.
BPMPD	LP, GR	W, T, h, q	dd	V2.1, 1996	SLP-IOR
DAPPROX	CR	T, h	i, dd, cd	2001	SLP-IOR
HiPlex	LP, GR	W, T, h, q	dd	V1.01, 1994	Author
HOPDM	LP, GR	W, T, h, q	dd	V2.3, 1996	SLP-IOR
MINOS	LP, GR	W, T, h, q	dd	V5.4, 1995	Commercial
MScr2Scr	MSR	h	dd	2001	SLP-IOR
MSLiP	GR	W, T, h, q	dd	V8.2, 1992	SLP-IOR
OB1	LP, GR	W, T, h, q	dd	ROB1, 1989	Authors
PCSPIOR	JC	h	nd	1995	SLP-IOR
PROBALL	JC	h	nd	2001	SLP-IOR
PROCON	JC	h	nd	1992	SLP-IOR
QDECOM	FR	T, h, q	dd	1985	SLP-IOR
SDECOM	CR	T, h	i, dd, cd	1995	SLP-IOR
SHOR1	SR	h	dd	1997	SLP-IOR
SHOR2	CR	h	dd	1998	SLP-IOR
SIRD2SCR	SIR	h	dd	1994	SLP-IOR
SRAPPROX	SR	h	dd, cd	1994	SLP-IOR
XMP	LP, GR	W, T, h, q	dd	1986	Commercial

- *Stochastic parts*: W, T, h, q denote the recourse matrix, technology matrix, right-hand side, and objective, respectively.
- *Distribution*: *i* denotes stochastic independence, *dd* means finite discrete distribution, *cd* stands for the normal, uniform, and exponential continuous univariate distributions, *nd* means nondegenerate multivariate normal distribution, and *all* designates all univariate distributions.
- *Availability*: *SLP-IOR* means that the solver is distributed along with SLP-IOR. *Author(s)* means that the user needs a separate license from the author(s).

6.6 Workbench facilities for testing SLP solvers

SLP-IOR encompasses an integrated workbench with facilities for dealing with test problem batteries and solvers.

6.6.1 Dealing with test problem batteries

For testing solvers, test problem batteries are needed. They either can be imported in SMPS format or can be randomly generated.

Table 6.2. *Solvers versus models. For DAPPROX and SDECOM, an additional requirement is the stochastic independence of the random variables.*

	LP -	GR dd	FR dd	CR dd	CR cd	SR dd	SR cd	SIR dd	MSR dd	JC nd	SC all
BPMPD	*	*	*	*	-	*	-	-	-	-	*
DAPPROX	-	-	-	*	*	*	*	-	-	-	-
HiPlex	*	*	*	*	-	*	-	-	-	-	*
HOPDM	*	*	*	*	-	*	-	-	-	-	*
MINOS	*	*	*	*	-	*	-	-	-	-	*
MScr2Scr	-	-	-	-	-	-	-	-	*	-	-
MSLiP	-	*	*	*	-	*	-	-	-	-	-
OB1	*	*	*	*	-	*	-	-	-	-	*
PROBALL	-	-	-	-	-	-	-	-	-	*	-
PROCON	-	-	-	-	-	-	-	-	-	*	-
PCSPIOR	-	-	-	-	-	-	-	-	-	*	-
QDECOM	-	-	*	*	-	*	-	-	-	-	-
SDECOM	-	-	-	*	*	*	*	-	-	-	-
SHOR1	-	-	-	-	-	*	-	-	-	-	-
SHOR2	-	-	-	*	-	*	-	-	-	-	-
SIRD2SCR	-	-	-	-	-	-	-	*	-	-	-
SRAPPROX	-	-	-	-	-	*	*	-	-	-	-
XMP	*	*	*	*	-	*	-	-	-	-	*

The test problem generator GENSLP of Kall and Keller (see [31]) has been further developed, and it is now integrated into SLP-IOR. It serves for randomly generating three kinds of test problem batteries: batteries consisting of deterministic LP problems, batteries containing recourse problems with guaranteed existence of an optimal solution, and test problem batteries with jointly chance-constrained problems having a known solution.

Another facility serves for generating test problem batteries as follows: a single SLP problem can be chosen and the test problem battery computed by imposing random perturbations on a selected array of this model instance.

Test problem batteries can be handled as a whole. The following operations can be performed on each element of the battery in turn:

- discretizing the probability distribution;
- endowing the test problems with a normal distribution;
- injecting a fixed distribution;
- selection of a set of solvers; each element in the battery can be solved in turn without further user interaction. The computational summary tables are presented as LaTeX tableaus.

6.6.2 Connecting a solver to SLP-IOR

For serving as a workbench, an important issue is that the user should be able to connect her or his solver to the system. SLP-IOR is an open system in this respect.

The solver library is organized as follows. The solver descriptions are kept in a solver description database, whereas the executables of the solvers are placed in a separate directory. The solver description database contains all data concerning the currently connected solvers, such as solver capabilities concerning model type and structure, maximal dimensions, run-time parameters, termination codes, and input/output formats. For connecting a new solver, the following steps must be carried out:

- The above-mentioned database must be updated by the user. This can be done by entering the data concerning the new solver in a menu-driven fashion.
- The input/output part of the solver must be updated for receiving problem data and for returning the solution.
  - *Communicating problem data from SLP-IOR to the solver*: If the solver accepts problem data in MPS/SMPS format, then there is no need to change the source of the solver; it is sufficient to declare this fact in the solver database session. If this is not the case, then the solver input routine must be changed to read data according to the format used in communicating data to our own solvers. This is accomplished via a transparent ASCII data file.
  - *Communicating results from the solver to SLP-IOR*: The solver is supposed to write two ASCII files. The first is a one-line file containing termination code, number of iterations, solution time, and objective value. The second file contains just the solution vector.
- The executable of the solver must be put into the solver's subdirectory.

The solver's own summary file, log file, and error file can be viewed within SLP-IOR after solver termination. The newly connected solver is handled by SLP-IOR in all respects in the same way as the solvers connected to it by ourselves.

6.7 Recommendations and limitations

6.7.1 Some guidelines

In this section we present some recommendations, based on our computational experience, concerning the building and solving of SLP models with SLP-IOR. In the discussion we consider only those solvers which are part of the SLP-IOR distribution; see Table 6.1.

From the modeling point of view, we would like to stress the usefulness of the affine relations (6.2); whenever appropriate, this technique for modeling the random entries should be utilized, especially for recourse problems. The main reason is that it can lead to a significant reduction in the number of random variables (the dimension of the random vector ξ). As a side effect, it is frequently the case that the components of ξ are stochastically independent, whereas the random array entries themselves are highly stochastically dependent. This

technique makes no difference for solvers aiming at the solution of the deterministic equivalent LP, like BPMPD, HOPDM, MSLiP, or QDECOM, because the equivalent LP relies on the joint realizations of the random entries themselves. Reducing the number of random variables and having a random vector ξ with stochastically independent entries opens the way, however, for utilizing the successive approximation technique, i.e., DAPPROX. This solver can be used for solving recourse problems with huge numbers of joint realizations, far beyond the capabilities of the solvers intended for the equivalent LP.

An important point is that for recourse problems, solvers utilizing special structure should be used whenever possible. On the one hand, the computational time for the solution can be significantly reduced this way. On the other hand, larger problems (with respect to the number of random variables and/or number of realizations in the discrete case) may still be within the scope of the solver.

All computational results quoted in the rest of this section are drawn from our own experience.

The most important special structure is simple recourse; SLP-IOR provides a tool for exploring whether the SLP problem has this structure. Simple recourse models can be solved with a large number of random variables having an astronomical number of joint realizations. According to our experience, simple recourse models with dimensions $A(300 \times 500)$, $W(300 \times 600)$ with 300 random variables and 5^{300} joint realizations can be solved on a 660 MHz IBM PC (with 256 MB RAM) using SRAPPROX in approximately 30 seconds. Models with multiple simple recourse behave similarly with respect to computing times as simple recourse models.

Another important subclass of recourse models is the class of models with (relatively) complete recourse. SLP-IOR provides a tool for checking whether the model has the complete recourse property. The two main algorithmic approaches, specialized to this type of models, are successive discrete approximation, implemented in DAPPROX, and stochastic decomposition, implemented in SDECOM. DAPPROX is well suited to problems with independent random variables if the number of random variables is less than 12. For the discrete case, the number of joint realizations plays a minor role. In our experience, complete recourse models with dimensions $A(10 \times 20)$, $W(5 \times 10)$, with five discretely distributed independent random variables having approximately 700,000 joint realizations, can be solved with DAPPROX on the same PC as mentioned above in 6 seconds on average, provided that only the right-hand side is stochastic. If the technology matrix is also stochastic, the computation time increases to approximately 170 seconds. SDECOM relies on sampling, and it provides a solution which is optimal in the statistical sense. In our present implementation, the stochastic independence of the random variables is presupposed. The computing times for the above-mentioned problem size are approximately 7 seconds and 17 seconds, respectively. Concerning the number of random variables, there is no inherent upper bound for SDECOM; however, we have not yet tested this solver for a larger number of random variables.

For two-stage complete recourse problems with stochastically independent random variables having continuous distributions, we have direct solvers, provided that the distribution is either uniform, normal, or exponential. These solvers are DAPPROX, SDECOM, and SRAPPROX, the latter for simple recourse problems only. For the other continuous distributions, approximate solutions can be computed by discretizing the distribution and by subsequently solving the approximate problem.

For jointly chance-constrained problems, we have the solvers PCSPIOR, PROBALL, and PROCON, for the case when only the right-hand side is stochastic and has a non-degenerate multivariate normal distribution. At present we consider PROBALL our best solver. Typical computation times with PROBALL for models with $A(48 \times 46)$ and four random variables in the chance (probabilistic) constraint are approximately 0.2 seconds. For models with $A(80 \times 160)$ and 20 random variables the computing time amounts to approximately 160 seconds, and finally for models with $A(50 \times 120)$ and 30 random variables the computing time is approximately 20 minutes.

6.7.2 Current limitations

We have to distinguish two kinds of limitations. The first concerns the core part of SLP-IOR, i.e., the handling of SLP model instances. The second concerns the solver library.

Considering the core part, the limitations are primarily imposed by the user's machine capacity. SLP-IOR deals with arrays by employing sparse matrix techniques. This implies that (within machine capacity) SLP problems can be built and handled without hardwired limitations in the code. It may happen that for a model instance there is no appropriate solver in the solver library of SLP-IOR. In this case the SLP problem data can be exported in (S)MPS format to an external solver, or the user can connect her or his favorite solver to SLP-IOR.

Considering the solver availability in the solver library, Tables 6.1 and 6.2 summarized the present status. Here we point out some of the limitations.

For random recourse problems, only the general-purpose LP solvers are available via solving the equivalent LP. For fixed recourse with finite discrete distributions, MSLiP and QDECOM are available in addition; these solvers can also handle stochastic objectives. In the case of complete recourse problems, DAPPROX and SDECOM are also available, with the limitation that the random variables should be stochastically independent. For DAPPROX the number of random variables should not be too high; on a 660 MHz IBM PC with 256 MB RAM the limit is approximately 11.

For jointly chance-constrained problems, solvers are available only for the case when only the right-hand side is stochastic and has a nondegenerate multivariate normal distribution. For separate chance constraints, solvers are provided only for the case when only the right-hand side is stochastic; there are no limitations concerning the probability distributions in this case.

6.8 Further development

The following are features which have been added to SLP-IOR since the writing of this chapter:

- *Multistage recourse problems with finite discrete distribution, i.e., with the scenario approach.* The extension involves dealing with scenario trees and building of deterministic equivalent LPs for the solver phase. For scenario generation (discretization), we have added some heuristic procedures.

- *SLP problems with integrated chance constraints and discretely distributed random variables.* This also involves the development of a solver.

Bibliography

[1] J. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer-Verlag, New York, 1997.

[2] J. R. Birge, M. A. H. Dempster, H. I. Gassmann, E. Gunn, A. J. King, and S. W. Wallace, *A Standard Input Format for Multiperiod Stochastic Linear Programs*, Working Paper WP-87-118, IIASA, Laxenburg, Austria, 1987.

[3] A. Brooke, D. Kendrick, and A. Meeraus, *GAMS. A User's Guide, Release* 2.25, Boyd and Fraser/The Scientific Press, Danvers, MA, 1992.

[4] A. Brooke, D. Kendrick, A. Meeraus, and R. Raman, *GAMS. A User's Guide*, Technical Report, GAMS Development Corporation, Washington DC, 1998; also available online from http://www.gams.com/docs.

[5] C. Condevaux-Lanloy and E. Fragnière, *SETSTOCH: A Tool for Multistage Stochastic Programming with Recourse*, Logilab Technical Report, Department of Management Studies, University of Geneva, Geneva, Switzerland, 1998.

[6] G. Consigli and M. Dempster, *Dynamic stochastic programming for asset-liability management*, Ann. Oper. Res., 81 (1998), pp. 131–161.

[7] I. Deák *Subroutines for computing normal probabilities of sets—Computer experiments*, Ann. Oper. Res., 100 (2000), pp. 103–122.

[8] M. Evans, N. Hastings, and B. Peacock, *Statistical Distributions*, John Wiley, New York, 1993.

[9] H. Gassmann, *MSLiP: A computer code for the multistage stochastic linear programming problem*, Math. Program., 47 (1990), pp. 407–423.

[10] H. Gassmann and E. Schweitzer, *A comprehensive input format for stochastic linear programs*, Ann. Oper. Res., 104 (2001), pp. 89–125.

[11] H. I. Gassmann, *Modelling support for stochastic programs*, Ann. Oper. Res., 82 (1998), pp. 107–137.

[12] H. I. Gassmann and A. M. Ireland, *On the formulation of stochastic linear programs using algebraic modeling languages*, Ann. Oper. Res., 64 (1996), pp. 83–112.

[13] J. Gondzio, *HOPDM (version* 2.12)*—A fast LP solver based on a primal-dual interior point method*, Eur. J. Oper. Res., 85 (1995), pp. 221–225.

[14] J. L. Higle and S. Sen, *Stochastic Decomposition. A Statistical Method for Large Scale Stochastic Linear Programming*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[15] C. C. HUANG, W. T. ZIEMBA, AND A. BEN-TAL, *Bounds on the expectation of a convex function of a random variable: With applications to stochastic programming*, Oper. Res., 25 (1977), pp. 315–325.

[16] P. KALL, *Approximations to stochastic programs with complete fixed recourse*, Numer. Math., 22 (1974), pp. 333–339.

[17] P. KALL AND J. MAYER, *A model management system for stochastic linear programming*, in System Modelling and Optimization, P. Kall, ed., Springer-Verlag, New York, 1992, pp. 580–587.

[18] P. KALL AND J. MAYER, *SLP-IOR: A model management system for stochastic linear programming: System design*, in Optimization-Based Computer-Aided Modelling and Design, A. Beulens and H.-J. Sebastian, eds., Springer-Verlag, New York, 1992, pp. 139–157.

[19] P. KALL AND J. MAYER, *SLP-IOR: A model management system for stochastic linear programming*, in Statistical Methods for Decision Processes, G. Hellwig, P. Kall, and P. Abel, eds., Daimler Benz AG, Stuttgart-Möhringen, 1994, pp. 54–63.

[20] P. KALL AND J. MAYER, *Computer support for modeling in stochastic linear programming*, in Stochastic Programming: Numerical Techniques and Engineering Applications, K. Marti and P. Kall, eds., Springer-Verlag, New York, 1995, pp. 54–70.

[21] P. KALL AND J. MAYER, *SLP-IOR: An interactive model management system for stochastic linear programs*, Math. Program., 75 (1996), 221–240.

[22] P. KALL AND J. MAYER, *On solving stochastic linear programming problems*, in *Stochastic Programming Methods and Technical Applications*, K. Marti and P. Kall, eds., Springer-Verlag, New York, 1998, pp. 329–344.

[23] P. KALL AND J. MAYER, *On testing SLP codes with SLP-IOR*, in New Trends in Mathematical Programming, F. Giannessi, T. Rapcsák, and S. Komlósi, eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998, pp. 115–135.

[24] P. KALL AND D. STOYAN, *Solving stochastic programming problems with recourse including error bounds*, Math. Oper. Statist. Ser. Opt., 13 (1982), pp. 431–447.

[25] P. KALL AND S. W. WALLACE, *Stochastic Programming*, John Wiley, New York, 1994.

[26] A. KING, *SP/OSL Version* 1.0 *Stochastic Programming Interface User's Guide*, Technical Report, IBM Research Division, Thomas J. Watson Research Center, Yorktown Heights, NY, 1994.

[27] W. K. KLEIN HANEVELD, *Duality in Stochastic Linear and Dynamic Programming*, Lecture Notes in Econ. and Math. Systems 274, Springer-Verlag, New York, 1986.

[28] W. K. KLEIN HANEVELD, L. STOUGIE, AND M. H. VAN DER VLERK, *An algorithm for the construction of convex hulls in simple integer recourse programming*, Ann. Oper. Res., 64 (1996), pp. 67–81.

[29] I. MAROS, *A general Phase-I method in linear programming*, Eur. J. Oper. Res., 23 (1986), pp. 64–77.

[30] I. MAROS AND G. MITRA, *Strategies for creating advanced bases for large-scale linear programming problems*, INFORMS J. Comput., 10 (1998), pp. 248–260.

[31] J. MAYER, *Stochastic Linear Programming Algorithms: A Comparison Based on a Model Management System*, Gordon and Breach, New York, 1998.

[32] E. MESSINA AND G. MITRA, *Modelling and analysis of multistage stochastic programming problems: A software environment*, Eur. J. Oper. Res., 101 (1997), pp. 343–359.

[33] CS. MÉSZÁROS, *The augmented system variants of IPM's in two stage stochastic programming*, Eur. J. Oper. Res., 101 (1997), pp. 317–327.

[34] A. PRÉKOPA, *Stochastic Programming*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.

[35] A. RUSZCZYŃSKI, *A regularized decomposition method for minimizing a sum of polyhedral functions*, Math. Program., 35 (1986), pp. 309–333.

[36] N. SHOR, T. BARDADYM, N. ZHURBENKO, A. LIKHOVID, AND P. STETSYUK, *The use of nonsmooth optimization methods in stochastic programming problems*, Kibernet. Sistem. Anal., 5 (1999), pp. 33–47 (in Russian); English translation: Cybernet. System Anal., 35 (1999), pp. 708–720.

[37] T. SZÁNTAI, *Calculation of the Multivariate Probability Distribution Function Values and Their Gradient Vectors*, Working Paper WP-87-82, IIASA, Laxenburg, Austria, 1987.

[38] P. VALENTE, G. MITRA, C. POOJARI, AND T. KYRIAKIS, *Software Tools for Stochastic Programming: A Stochastic Programming Integrated Environment (SPInE)*, Technical Report TR/10/01/, Department of Mathematical Sciences, Brunel University, Uxbridge, Middlesex, UK, 2001.

[39] M. H. VAN DER VLERK, *On Multiple Simple Recourse Models*, Stochastic Programming E-Print Series 2002-7, Department of Econometrics and Operations Research, University of Groningen, Groningen, The Netherlands, 2002; also available online from http://dochost.rz.hu-berlin.de/speps/.

[40] W. T. ZIEMBA, *Computational algorithms for convex stochastic programmes with simple recourse*, Oper. Res., 18 (1970), pp. 414–431.

This page intentionally left blank

Chapter 7

Stochastic Programming from Modeling Languages

Emmanuel Fragnière and Jacek Gondzio†*

7.1 Introduction

The majority of deterministic mathematical programming problems have a compact formulation in terms of algebraic equations. Therefore they can easily take advantage of the facilities offered by algebraic modeling languages (AMLs). These tools allow expressing models by using convenient mathematical notation (algebraic equations) and translate the models into a form understandable by the solvers for mathematical programs.

AMLs provide facility for the management of a mathematical model and its data, and access different general-purpose solvers. The use of AMLs simplifies the process of building the prototype model and in some cases makes it possible to create and maintain even the production version of the model.

As presented in other chapters of this book, stochastic programming is needed when exogenous parameters of the mathematical programming problem are random. Dealing with stochasticities in planning is not an easy task. In a standard scenario-by-scenario analysis, the system is optimized for each scenario separately. Varying the scenario hypotheses, we can observe the different optimal responses of the system and delineate the "strong trends" of the future. Indeed, this scenario-by-scenario approach implicitly assumes perfect foresight. The method provides a first-stage decision, which is valid only for the scenario under consideration. Having as many decisions as there are scenarios leaves the decision maker without a clear recommendation. In stochastic programming the whole set of scenarios is combined into an event tree, which describes the unfolding of uncertainties over the period

*School of Management, University of Bath, Bath BA2 7AY, England (mnsef@bath.ac.uk). The research of this author was supported by Fonds National de la Recherche Scientifique Suisse grant #1213-058892.99/1.

†Department of Mathematics and Statistics, University of Edinburgh, King's Buildings, Edinburgh, EH9 3JZ, Scotland (J.Gondzio@ed.ac.uk). The research of this author was supported by Engineering and Physical Sciences Research Council of UK grant GR/M68169.

of planning. The model takes into account the uncertainties characterizing the scenarios through stochastic programming techniques. This adaptive plan is much closer, in spirit, to the way that decision makers have to deal with an uncertain future in real life.

Most of the difficulties in modeling uncertainty through stochastic programming originate from the lack of an agreed standard of its representation. Indeed, stochastic programming problems usually involve dynamic aspects of decision making which combined with uncertainty inevitably lead to a complicated model. To make the problem tractable, uncertainty is usually expressed in terms of an approximate discrete distribution. However, the need for accuracy in modeling inevitably leads to an explosion of dimension in the size of the corresponding mathematical program. This imposes additional limits on the ways of modeling stochastic programming problems and further complicates the management of such models. In consequence there still does not exist a *standard* way of modeling stochastic programming problems in AMLs. However, AML developers are working on them and have already come up with a number of possible extensions.

In this chapter we address the difficulties of modeling stochastic programs and discuss in detail different approaches developed so far to deal with this problem.

The chapter is organized as follows. In section 7.2 we briefly explain the important role played by AMLs in the development of optimization-based models. In section 7.3 we present different formulations of stochastic programs. In section 7.4 we discuss specific issues related to an automatic generation of stochastic programs that result in difficulties with standardization of their generation by AMLs. In section 7.5 we discuss the techniques of stochastic programming available to AMLs, and in section 7.6 we comment on the crucial issues of communication between the solver and the AML. Section 7.7 contains concluding remarks.

7.2 AMLs

AMLs enable decision models to be formulated with an algebraic notation. They use a generic model description in the form of a data file. The models developed with AMLs can be easily modified. The user builds the model and provides the AML with the appropriate data. The AML translates the model into a form that is understandable to a solver and invokes the appropriate solver. In this setting, the solver is seen as a black box. The optimization code may query the AML about any additional information on the problem. For example, a nonlinear optimization code may ask for the function values as well as the first and the second derivatives at a given point. Once the solution of the mathematical program is found, it is returned to the AML and the results are reported to the user.

AML enables a modeler to express the problem in an index-based mathematical form with abstract entities: sets, indices, parameters, variables, and constraints. The key notion in the AML is the ability to group conceptually similar entities into a set. Once the entities are grouped in a given set, they can be referenced by indices to the elements of this set. This leads to a problem formulation that is very close to the formulation using algebraic notation. For instance, the mathematical operation $\sum_{i \in I} X_i$ is represented by the expression `SUM(I, X(I))` in the GAMS modeling language. The role of the AML is to expand the compact problem formulation (problem structure and data) into the problem instance, which is ready to be solved by an appropriate optimization code. This operation is realized within the AML

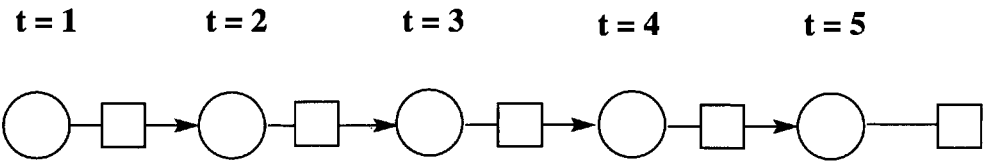


Figure 7.1. *Deterministic Invendeman model.*

by replicating every entity over the different elements of the set. This is often referred to as a *set-indexing* ability of the AML. The user of an AML can define generic expressions that are indexed over several sets. Set-indexing in such cases involves compound sets.

There exist many AMLs or, more generally, optimization modeling languages [14]. AMLs such as GAMS [5], AMPL [13], and AIMMS [3] are routinely used by the mathematical programming community.

To illustrate the use of such modeling tools, we first present the algebraic formulation of a multiperiod inventory model with deterministic demands. A full description of this model, called Invendeman, can be found in [24, Chapter 10]. The model has the form of a simple optimization problem:

$$\begin{aligned} \max \quad & \sum_{t=1}^{T}((p_t - 2)x_t^- - (p_t + 2)x_t^+ - hI_t) \\ \text{s.t.} \quad & x_t^- - x_t^+ + I_t - I_{t-1} = -d_t, \\ & I_t \leq \bar{I}, \\ & x_t^-, x_t^+, I_t \geq 0. \end{aligned} \tag{7.1}$$

The objective function is net profit. There are three generic variables (inventory, quantity bought, quantity sold) and one generic constraint (inventory balance), all indexed over time. The model is generated for five periods, as shown in Figure 7.1. The variables used in the model have the following meanings:

- t is the time period, $t = 1, 2, \ldots, T$;
- x_t^+ is the quantity bought in period t;
- x_t^- is the quantity sold in period t;
- 2 is the unit transactions cost which has to be paid each time a purchase or sale is made;
- p_t is the market price at time t; a seller gets $p_t - 2$; a buyer pays $p_t + 2$;
- d_t is the demand of the firm for the commodity at time t;
- I_t is the stock of inventory held at time $t = 0, 1, \ldots, T$;
- $\bar{I}$ is the fixed warehouse capacity; and
- h is the unit holding cost for inventory.

We present below an extract of the corresponding model written with the GAMS [5] modeling language (the full model along with the data appears in Appendix 7.A.1):

```
OBJECTIVE..    PROFIT =E= SUM(INDEX,(P(INDEX)-2.0)*XMINUS(INDEX)
               -(P(INDEX)+2.0)*XPLUS(INDEX)-H*I(INDEX-1));
INVBAL(T-1).. XMINUS(T)-XPLUS(T)+I(T)-I(T-1) =E= -D(T);
```

The formulation in GAMS is very close to the original algebraic formulation. In general terms, AMLs provide declarative statements (as opposed to programming languages, which contain procedural statements such as `loops` or `if-then-else` commands). This means that the code in the case of an AML can be seen as a declaration of the properties of the optimization problem.

The modeling language takes as an input the algebraic formulation of the model and a set of data. Next all operations are automated. The modeling language generates a mathematical program, also called an instance of the problem. In a particular case of the deterministic multiperiod inventory model, the generated instance can be seen as a unique scenario. Later in this chapter we shall extend this model to take uncertainty into account. This will necessitate considering several scenarios.

7.3 Different formulations of stochastic programming problems

A multistage stochastic program with recourse is a multiperiod mathematical program where parameters are assumed to be uncertain along the time path. The term *recourse* means that the decision variables adapt to the different outcomes of random parameters at each time period. A natural formulation of the stochastic programming problem relies on recursion [2] to describe dynamics of the modeled process. Several different formulations of stochastic programs have been discussed in detail already in Part 1. Therefore we omit recursive formulations and only briefly mention event trees and the deterministic equivalent formulation and an alternative formulation with nonanticipativity constraints, two forms which are most often used for modeling stochastic programs with AMLs.

7.3.1 Event tree and the deterministic equivalent formulation

In a planning approach the evolution of uncertainties can be described as an alternation of decisions and random realizations. In its simplest form the discrete stochastic process can be represented as an event tree describing the unfolding of the uncertainty over the period of planning (see Figure 7.2). A path, from the root to a leaf of the event tree, represents a scenario. Each scenario has a given probability. At each node of the event tree, a set of constraints and an economic function, which involves variables specific to that node and its predecessor nodes, are defined. For instance, node 1 of the tree presented in Figure 7.2 corresponds to the first stage, and associated decisions are identical for scenarios 1, 2, 3, and 4. At stage 2, decisions of scenarios 1 and 2 are identical. In the same way decisions of scenarios 3 and 4 are identical.

To formulate the deterministic equivalent of the multistage stochastic programming problem we first need to enumerate all nodes of the event tree (see Figure 7.2). We use a

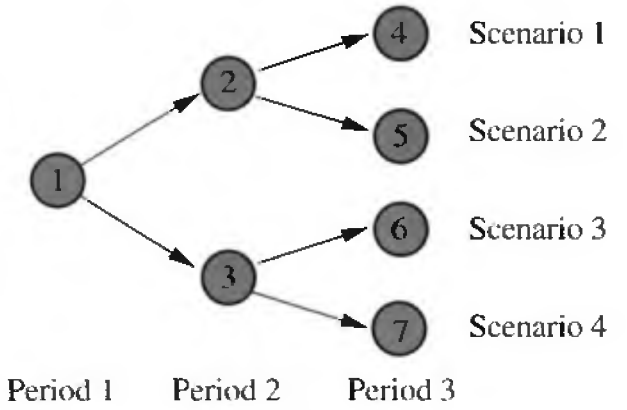


Figure 7.2. *A simple event tree.*

breadth-first search order; i.e., we start from a root node corresponding to the initial stage and end with leaf nodes corresponding to the last stage. Let $t = 1, 2, \ldots, T$ denote the stage, and l_t be the index of a node at stage t. Thus the root node has index $l_1 = 1$, and the stage 2 nodes start from index 2. Let L_t denote the last node at stage t. Hence the nodes that belong to stage $t > 1$ have indices $l_t = L_{t-1} + 1, L_{t-1} + 2, \ldots, L_t$. To capture dynamics in the model we use $a(l_t)$ to denote the direct ancestor of node l_t. Clearly, the ancestor of l_t is a node that belongs to stage $t - 1$ (e.g., node 2 in Figure 7.2 is the ancestor of nodes 4 and 5). All decision variables x are indexed only by the node number in the event tree: we use a superscript l_t. The stage the variable belongs to is therefore defined implicitly. The main constraint that describes system dynamics is

$$T^{l_t} x^{a(l_t)} + W^t x^{l_t} = h^{l_t}, \qquad l_t = L_{t-1} + 1, L_{t-1} + 2, \ldots, L_t, \tag{7.2}$$

where T^{l_t} is the technology matrix that varies with the node in the event tree and W^t is the recourse matrix that varies only with time but does not depend, in our example, on the realization within the same stage. One could impose different conditions on matrices T and W and use indexing with time t or both time and uncertainty l_t.

The deterministic equivalent formulation of the multistage problem is

$$\begin{aligned}
\min\ & c^T x^1 + \sum_{l_2=2}^{L_2} p^{l_2} (q^{l_2})^T x^{l_2} + \sum_{l_3=L_2+1}^{L_3} p^{l_3} (q^{l_3})^T x^{l_3} + \cdots + \sum_{l_T=L_{T-1}+1}^{L_T} p^{l_T} (q^{l_T})^T x^{l_T} \\
\text{s.t. } & A x^1 = b, \\
& T^{l_2} x^1 + W^2 x^{l_2} = h^{l_2}, \quad l_2 = 2, \ldots, L_2, \\
& T^{l_3} x^{a(l_3)} + W^3 x^{l_3} = h^{l_3}, \quad l_3 = L_2 + 1, \ldots, L_3, \\
& \qquad \ddots \qquad\quad \vdots \\
& T^{l_T} x^{a(l_T)} + W^T x^{l_T} = h^{l_T}, \quad l_T = L_{T-1} + 1, \ldots, L_T, \\
& x^{l_t} \geq 0, \qquad\qquad l_t = 1, \ldots, L_T.
\end{aligned} \tag{7.3}$$

The numbers of children of each node in the event tree may differ, as they depend on a probability distribution of the appropriate stochastic process. If the depth-first search or-

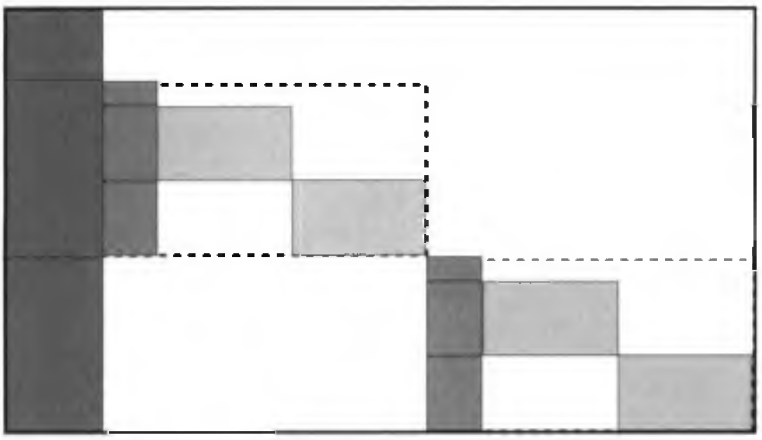


Figure 7.3. *The constraint matrix associated with the event tree.*

dering of the nodes in the event tree is maintained during the generation of the mathematical program, the corresponding constraint matrix displays a nested dual block-angular structure. Links between the nested dual block-angular structure and the algebraic formulation of the original model can be easily established.

In our example (see Figure 7.2), node 1 is the root node, nodes 2 and 3 belong to stage 2 ($l_2 = 2, 3$), and nodes 4 to 7 belong to stage 3 ($l_3 = 4, \ldots, 7$). The deterministic equivalent formulation of the problem is presented in (7.4). Let us observe that by shifting x^3 just after x^5 and shifting the third constraint after the fifth one, we immediately retrieve the structure presented in Figure 7.3. It is worth noting that this reordering operation means changing the breadth-first search order of nodes in Figure 7.2 (1, 2, 3, 4, 5, 6, 7) to the depth-first search order 1, 2, 4, 5, 3, 6, 7.

$$
\begin{array}{ll}
\min & c^T x^1 + p^2(q^2)^T x^2 + p^3(q^3)^T x^3 + p^4(q^4)^T x^4 + p^5(q^5)^T x^5 + p^6(q^6)^T x^6 + p^7(q^7)^T x^7 \\
\text{s.t.} & \\
 & \begin{array}{lllllll}
Ax^1 & & & & & & = b, \\
T^2x^1 + W^2x^2 & & & & & & = h^2, \\
T^3x^1 & + W^2x^3 & & & & & = h^3, \\
T^4x^2 & & + W^3x^4 & & & & = h^4, \\
T^5x^2 & & & + W^3x^5 & & & = h^5, \\
 & T^6x^3 & & & + W^3x^6 & & = h^6, \\
 & T^7x^3 & & & & + W^3x^7 & = h^7,
\end{array} \\
 & x^1 \geq 0, \quad x^2 \geq 0, \quad x^3 \geq 0, \quad x^4 \geq 0, \quad x^5 \geq 0, \quad x^6 \geq 0, \quad x^7 \geq 0.
\end{array}
\tag{7.4}
$$

The probabilities in the objective function of problem (7.4) are not scenario probabilities but (partial) path probabilities: p^n is the probability (at the start) that a path goes through node n. Clearly, (7.4) represents a structured linear program. Its structure should be exploited in the solution algorithm. Unfortunately, if the model is written with an AML, the structure, easily identifiable in the algebraic formulation, is usually lost when the corresponding mathematical program is sent to the solver. Each AML uses its own algorithm to generate an equivalent mathematical program, which scrambles the structure.

7.3.2 Formulation with nonanticipativity constraints

Another way to write the deterministic equivalent consists in creating independent copies of variables corresponding to every ancestor in the tree for every child of this node. We replicate the variable $x^{a(l_t)}$ in (7.2) and create copies $x_{t-1}^{l_t}$ for each l_t corresponding to a child node of $a(l_t)$ in the event tree. We slightly change the notation at this point and explicitly add the stage subscript to each variable. Namely, with a given node l_t at stage t we associate two variables: an appropriate decision variable $x_t^{l_t}$ (at stage t) and a copy of the decision variable at the ancestor node corresponding to this particular child, $x_{t-1}^{l_t}$. For example, the variable x^3 representing the state corresponding to node 3 in stage 2 in Figure 7.2 would have two copies x_2^6 and x_2^7. Hence the last two constraints in (7.4) can be replaced with the two constraints

$$\begin{array}{rcrcrcl} T^6 x_2^6 & & + & W^3 x_3^6 & & = & h^6, \\ & T^7 x_2^7 & & & + \; W^3 x_3^7 & = & h^7, \end{array}$$

each with an independent set of variables. In the case of the example in Figure 7.2, for node $a(l_t) = 3$ we would have to add a constraint

$$x_2^6 = x_2^7.$$

Such a constraint is called a *nonanticipativity* or *locking* constraint.

The complete set of nonanticipativity constraints for problem (7.4) may thus be written as

$$\begin{aligned} x_1^4 &= x_1^5, \\ x_1^4 &= x_1^6, \\ x_1^4 &= x_1^7, \\ x_2^4 &= x_2^5, \\ x_2^6 &= x_2^7. \end{aligned} \tag{7.5}$$

There are other ways of representing the nonanticipativity constraints (the cyclical form is also frequently used).

7.4 Stochastic programs in AMLs

The presence of two different sets associated with time and uncertainty dimensions in stochastic programs creates a difficulty for an AML. The uncertainty (or scenario) needs to be indexed over time, and AMLs normally do not provide such a facility. Consequently, none of the formulations of stochastic programs presented in section 7.3 can be easily modeled in AMLs.

In this section, we make the assumption that probability distributions are discrete and that problems contain multiple stages or periods. Consequently, the problem can be represented in the form of an event tree. This event tree is made of scenarios. It is quite usual to relate variables of a given node with those that correspond to the ancestor node in the previous stage. For example, any constraint that describes dynamics of the system

would have such a form. However, the constraints which establish the link between the parent-child pair of nodes are particularly difficult to generate from the AML. The difficulty originates from the lack of standard description of the event tree or, more precisely, the lack of a tree-structured indexing system in AMLs.

When an AML generates the model, it performs extensive searches throughout the event tree. Therefore how the event tree is described becomes crucial. Trees are obviously used in many computer science applications. There exist many different ways of describing and coding trees, and event trees used in stochastic programming could take advantage of these developments. Unfortunately, such techniques are not usually available from AMLs. The difficulty lies in the type of indexing system required to describe an event tree.

Trees like the one presented in Figure 7.2 are symmetric (every node except the leaves has the same number of children). Tricks exist such as the one used by [16] to exploit the contiguity property to represent the symmetric tree and to easily retrieve the ancestor or the children of a given node (cf. [1, pp. 774–776] for details). The idea is to use the breadth-first ordering of nodes in the event tree. Consider, for example, a regular (symmetric) tree with d children at every node. Then the predecessor of node i is the node $a(i) = \lceil \frac{i-1}{d} \rceil$, where the ceiling function $\lceil . \rceil$ rounds up the argument to the next integer. The successors of node i are nodes $id - d + 2, id - d + 3, \ldots, id + 1$. Unfortunately, this addressing scheme cannot be generalized to unsymmetric event trees.

In many stochastic problems the discrete approximations of continuous distributions of random variables have various densities in different branches of the tree. Moreover, many models use trees that are automatically generated approximations of the stochastic process. These factors may lead to choosing highly unsymmetric event trees. Hence the restriction that only symmetric trees are modeled is unacceptable. The lack of efficient tree-structured indexing in algebraic formulations remains the main difficulty when AMLs are applied to generate stochastic programs. Although this could certainly be overcome at the cost of embedding some cumbersome generation schemes in AMLs, the major developers of AMLs hesitate before coding a devoted syntax to deal with stochastic programs in their modeling tools.

Modeling stochastic programs through AMLs is still in an early phase, but several attempts have been made to standardize this process. The following brief literature review gives a nonexhaustive list of attempts made in this direction.

Gassmann and Ireland [17, 18] note that stochastic programming type modeling could greatly benefit from the implicit declaration of scenarios, via the declaration of random parameters. Buchanan, McKinnon, and Skondras [6] propose extensions to AML that allow recursive definition of stochastic dynamic problems and facilitate the link with sampling techniques. Leuba and Morton [21] produce a complete SMPS format, i.e., the *core*, *time*, and *stoch* files (cf. Chapter 2 in this volume) directly from GAMS. Condevaux-Lanloy, Fragnière, and King [8] extend the structure exploiting tool [15] to permit the formulation of the SMPS format from the AML. In their approach the time-related information is retrieved from the core model handled by the AML, and the uncertainty information is loaded directly into the specialized SMPS-based solver outside the AML. Entriken [10] uses object-oriented programming techniques within the optimization modeling language to facilitate the management of stochastic programming models.

However, and in a general manner, we note the lack of standardization of modeling stochastic programs in AMLs. This has at least two reasons. First, there is not yet a widely

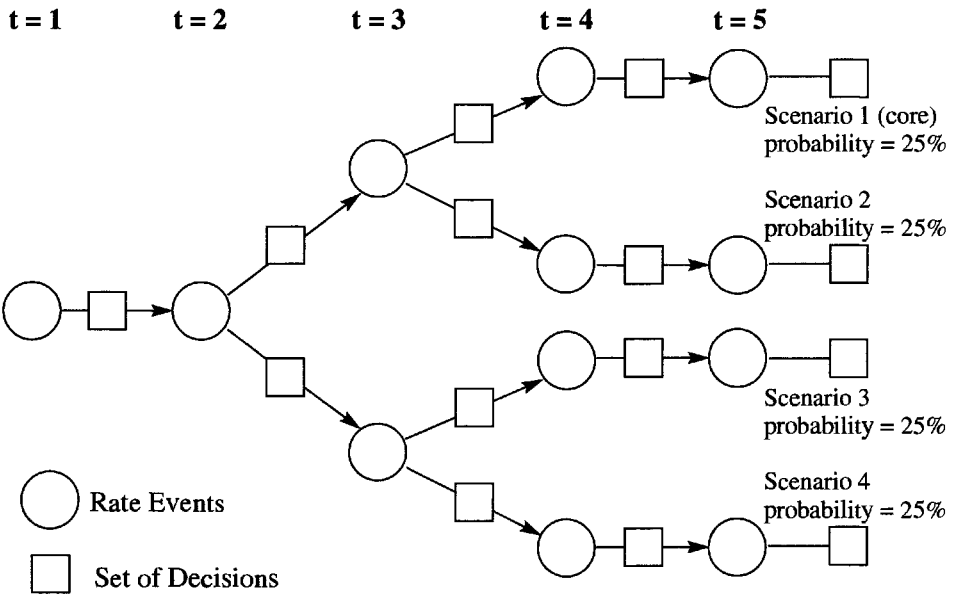


Figure 7.4. *Stochastic Invendeman model.*

accepted syntax for a description of stochastic programs. Second, there is not yet a compact and flexible format in which AMLs could send the stochastic program to the specialized solver.

Below we illustrate the difficulties of tree-structured indexing in more detail when we use locking constraints to extend the deterministic inventory management problem to take uncertainty into account. The locking constraints have to be indexed over the event tree. This is done by hand in the model discussed. Next, we present the proposition made by the AMPL developers to model event trees.

7.4.1 Stochastic extension of multiperiod inventory model

In this section we present an extension of the multiperiod inventory problem that takes uncertainty into account. We use explicit locking constraints in the GAMS model presented in section 7.2. Such an approach can be used to model both symmetric and unsymmetric event trees. However, it is rather tedious to implement.

Consider again the inventory problem (7.1). Suppose we introduce uncertainties in the future values of the demand parameters, as represented by the event tree of Figure 7.4, and we model the problem as a stochastic program with recourse. This means that some decisions (activity levels) will be made after the information about the true value has been obtained. However, some decisions have to be taken immediately. These immediate decisions should take into account the expected cost of the recourse.

The stochastic model written in GAMS includes a new index `S`, which stands for *scenarios*. Now the inventory balance constraint is generated for both the time and the

scenario dimensions. As we have four scenarios, h, l, m, and a, GAMS would generate four independent problems, each associated with a different set of data as indicated below.

```
TABLE
D(T,S)          market demand
              h     l     m     a
        0     0     0     0     0
        1     0     0     0     0
        2   200   200   150   150
        3   300   250   250   200
        4   400   400   400   400
        5     0     0     0     0
```

Let us observe that all generic constraints and variables include the index S. The objective in the stochastic inventory management problem is the expected value of the profit over all possible scenarios:

```
OBJECTIVE.. PROFIT=E=SUM((INDEX,S),PROB(S)*((P(INDEX)-2)*XMINUS(INDEX,S)
          -(P(INDEX)+2)*XPLUS(INDEX,S)-H*I(INDEX-1,S)));
INVBAL(T-1,S).. XMINUS(T,S)-XPLUS(T,S)+I(T,S)-I(T-1,S) =E= -D(T,S);
```

As mentioned in section 7.1, a possible way to deal with uncertainty is to define and analyze several independent scenarios. The model is then solved independently for each scenario and the optimal solutions are gathered and compared with each other. This approach does not provide the unique first-stage solution. Instead, it provides answers to "What-if" questions. The approach we present below extends it to ensure that the first-stage decisions are identical for all scenarios.

We can achieve this by explicitly forcing the first-stage decisions in all four otherwise independent scenarios to be identical. In the GAMS model we add the three constraints AFI0SS1, AFI0SS2, and AFI0SS3 that fix the initial inventory to be identical in all four scenarios h, l, m, and a. For example, the line

```
AFI0SS1..        I("0","h")=E=I("0","l");
```

forces I_0 in scenarios h and l to be the same, and the line

```
AFXM0SS1..       XMINUS("0","h")=E=XMINUS("0","l");
```

forces x_0^- in scenarios h and l to be identical. Several similar constraints are added for the variables at stage 1. At stage 2, however, only pairs of scenarios (h, l) and (m, a) are linked together. The number of necessary locking constraints is smaller than those in stages 0 and 1. All the constraints called AF* in the GAMS model are nonanticipativity constraints, where the last digit indicates the period in which the constraint is to be found.

The approach presented here requires explicit formulation of all nonanticipativity constraints and significantly increases the number of constraints in the model, as can be seen in the complete GAMS model given in Appendix 7.A.2. An important advantage of this approach is that it can be used for both symmetric and unsymmetric event trees. However, the approach is inefficient and prone to errors if a large number of nonanticipativity constraints has to be added. Using logical operators in set indexing would have allowed writing fewer locking constraints and leaving their generation to the AML.

The extensive formulation presented in this section illustrates clearly that the size of the simple model dramatically increases when the problem is transformed into a stochastic program. This type of stochastic programming formulation is therefore not tractable for large-scale problems.

7.4.2 The AMPL proposal

The developers of AMPL have proposed extensions to the syntax of their modeling language to allow a description of event trees. We reproduce their proposal below, following [11]. The modeling has been split into two steps. The first step consists of the definition of scenarios:

- `scenario` *scen-name*`;` Create a new *current* scenario. Inherit all set and parameter data from *parent* that was previously current. Incorporate subsequent data changes in *child* scenario only.
- `scenario` *scen-name* `{ indexing };` Create an indexed collection of scenarios.
- `scenario` *scen-name* `weight` *expr*`;` Associate a probability or other weight. *expr* denotes any arithmetic expression in current sets and parameters scenarios.

The second step adds scenarios referencing:

- `scenario` *scen-ref*`;` Make the indicated `scenario` current. *scen-ref* denotes either single *scen-name* or indexed *scen-name[object-ref]*.
- `scenario` *scen-name* `parent` *scen-ref*`;` Create new scenario having indicated parent, overriding the default (implicitly build a tree of scenarios).
- `nscens, scenname[`*expr*`], scen[`*expr*`]`. Extension of AMPLs generic names to scenario references (loop over all scenarios in the tree).

This approach has not been implemented yet. When implemented it would allow building stochastic programming models of small to medium sizes. However, the proposal does not give a clue as to how a large unsymmetric event tree can be modeled within AMPL.

More generally, the developers of AMLs do not want to commit their languages to a specific syntax of event tree description. This syntax is closely related to a standard in which problems are described in the AMLs and the format in which they are passed to the specialized stochastic programming solvers.

7.5 Stochastic programming solution techniques available to AMLs

At the writing of this chapter, the only option available in AMLs is to generate the full deterministic equivalent. The only alternative left is thus to use the general-purpose solvers that by default would use a direct solution method to tackle the problem. This approach is quite efficient as long as the problem is of small to medium size and can be generated

within memory limits. The need for accurate modeling of stochastic processes inevitably leads to a size explosion in the model. Even if the user is satisfied with the accuracy of the generated problem, and the general-purpose solver can solve this problem efficiently, there is a danger that the generation of the problem significantly contributes to the overall solution process. It is not unusual, for example, that model generation by an AML takes more time than the subsequent solution of the problem. Gondzio and Kouwenberg [19] have generated a medium-scale stochastic model by the GAMS modeling language and a specialized generation program [20]. The latter was 815 times faster.

Over the years many specialized techniques have been developed for stochastic programming. They usually exploit special structures of the problem. Many of these techniques rely on some variant of Benders decomposition [25]. The decomposition approach breaks the very large problem into smaller manageable optimization problems. This has several advantages. First, the peak memory requirement (needed to generate and then to read the deterministic equivalent problem) can be avoided. Additionally, the problem can be passed to the solver in pieces that are suitable for the decomposition approach. Therefore, as has been observed by [16], within the same memory limits decomposition-based solvers can deal with problems that are at least an order of magnitude larger than those solvable by a direct approach.

An alternative would consist of implementing simple decomposition techniques directly within AMLs. This approach is routinely used in certain economical applications: the decomposition loops are programmed in GAMS, for example, in the context of nonlinear stochastic programming problems [7]. Indeed, the presence of procedural statements such as `if-then-else` and `do-while` provided by most AMLs makes it possible to implement simple optimization algorithms. The interested reader can consult the library of examples of algorithms implemented through AMPL which includes Benders decomposition [12]. The article by Gassmann and Gay in this volume shows how to implement a nested Benders algorithm within the AMPL control language. The authors conclude that such an approach cannot be generalized because the AML-based decomposition algorithm depends on the syntax used by a particular model and is not reusable in a different model. Moreover, the AML is not necessarily the best environment to implement complicated optimization algorithms needed to solve stochastic programming problems efficiently.

Although several efficient algorithms have been proposed for stochastic programming, the limitations discussed so far prevent access to many of these techniques from AMLs. Indeed, the research in stochastic programming provides evidence that very large problems can be generated and solved. The research results on solution techniques are very much ahead of current links to solvers available in AMLs.

Below we recall some of the results that indicate currently achievable limits of solvable problems. We underline that all the solution techniques use parallel computing. Yang and Zenios [26] solved test problems with up to 2.6 million constraints and 18.2 million variables. They used a parallel direct interior-point method. Gondzio and Kouwenberg [19] solved a financial planning problem with 7 decision stages and a total of 5 million scenarios at the planning horizon, the linear program consisting of 12.5 million constraints and 25 million variables. They used an interior-point-based variant of Benders decomposition run on a 16-processor parallel machine. Blomvall and Lindberg [4] solved a problem with 10 stages and 1.9 million scenarios, resulting in a separable convex program with 119 million constraints and 67 million variables. They used a direct interior-point method with

a specialized Riccati-based solver for computing Newton directions and ran it on a Beowulf cluster of 32 PCs. Linderoth and Wright [22] solved a problem with 10 million scenarios, the linear program having 985 million constraints and 12,600 million variables (see also Chapter 5 of this volume). They used a variant of Benders decomposition and ran it on a grid of 1345 workstations.

To conclude, there is a need to improve the links between the AMLs and the solvers. Attempts have already been made in this direction. For example, Fragnière, Gondzio, and Vial [16] have used GAMS to generate a 1 million scenario problem, a linear program with 1,111,112 constraints and 2,555,556 variables. The problem was solved by a specialized parallel interior-point-based decomposition algorithm running on a cluster of 10 Linux PCs. The solver was accessed directly from the AML. Still the problem was passed to the solver in a deterministic equivalent form. This approach clearly demonstrates the need for improving the link between the AML and the specialized solver to avoid the bottleneck generation of the deterministic equivalent. We address this issue in the next section.

7.6 Communication between solver and AML

Every AML has a set of specialized routines to communicate with the solver. Usually, the whole problem is passed at once to the solver in the form of a text or binary file. This implies that sufficient memory has to be available to store the complete mathematical program. Typically AMLs generate the stochastic programming problem in the deterministic equivalent form and call a general-purpose optimization code to solve it. The size of the deterministic equivalent problem is proportional to the number of nodes in the event tree. Therefore, the AML may require a vast amount of memory to store it. As has already been mentioned, the real bottleneck is often not the memory requirement but the time of the problem generation.

At least some of the earlier mentioned drawbacks of the problem generation by AMLs could be avoided if the SMPS format were used. Moreover, any efficient solution method for stochastic programming is built on the exploitation of the special structure of the problem, and the complete structure information is available from the SMPS format. At the time of writing this chapter, AMLs cannot generate stochastic problems in SMPS format. However, several attempts to overcome this difficulty have already been made [6, 8, 23]. The problem has been treated in different ways. One of them consists of developing the extensions of existing AMLs dedicated to stochastic optimization.

Although s-Magic [6] does not produce SMPS format from the problem, it uses the recursive definition and communicates with the solver using a specialized memory-efficient description of the problem. The problem is represented in a compact format close in spirit to SMPS. Stochastic extension [8] of the structure exploiting tool [15] uses the AML to generate the deterministic part of the model in the form of the *core* and *time* files in the SMPS format. The information of uncertainty is produced outside the AML and communicated directly to the specialized solver. SPInE (cf. [23] and Chapter 8 in this volume) is a closed modeling system that generates the SMPS format of the stochastic programming problem and has access to built-in specialized optimization tools—the Benders decomposition. Direct solution of the deterministic equivalent form of the problem is also available as an option.

7.7 Conclusions

Stochastic programming is a promising technology for handling planning problems in uncertain environments. At least this has been said since the publication of the seminal paper on linear programming under uncertainty [9]. Unfortunately, due to modeling difficulties this technology has not yet reached the wide audience it deserves. To facilitate incorporating uncertainty in the planning models, user-friendly modeling systems are needed that can access the stochastic programming technology. Widely used AMLs are candidates to close this gap.

In this chapter we underlined the difficulties in the use of uncertainty in the modeling of real-life problems. We began our exposé with a discussion of an inventory problem. Deterministic formulations of this problem in the AML and in mathematical terms are very similar to each other. Then we explained the modification to include a stochastic dimension (uncertain demands) in the problem. Although the problem remains simple, it illustrates all the difficulties of including stochastic programs in modeling systems. We elaborated on different approaches that allow writing stochastic programs directly in AMLs. We ended the chapter with a discussion of stochastic programming solution techniques accessible from modeling systems. These systems certainly need further development to reach industry standard. We expect that this progress will be made in the next few years and the integrated modeling system for stochastic programming will enable the modelers to popularize the stochastic programming technology through relevant applications.

7.A Appendix

7.A.1 The GAMS model of deterministic inventory problem

```
SETS
  T     time periods   /0,1,2,3,4,5/
  INDEX(T)             /  1,2,3,4,5/
  OPENING(T)           /0           /
  TERMINAL(T)          /          5/;
PARAMETERS
  P(INDEX)    market price
                /1   75
                 2   65
                 3   89
                 4   77
                 5   80/
  D(T)        market demand
                /0    0
                 1    0
                 2  200
                 3  300
                 4  400
                 5    0/;
VARIABLES
  PROFIT
```

```
POSITIVE VARIABLES
  XMINUS(T)    quantity sold at time T
  XPLUS(T)     quantity bought at time T
  I(T)         inventory held at time T;
EQUATIONS
  OBJECTIVE    calculating net profit
  INVBAL(T)    inventory balance at time T;
OBJECTIVE..    PROFIT =E= SUM(INDEX,(P(INDEX)-2.0)*XMINUS(INDEX)
               -(P(INDEX)+2.0)*XPLUS(INDEX)-I(INDEX-1));
INVBAL(T-1).. XMINUS(T)-XPLUS(T)+I(T)-I(T-1)  =E=  -D(T);
I.UP(T)         = 500;
I.FX(OPENING)   = 300;
I.FX(TERMINAL) = 300;
MODEL INVENDEMAN/ALL/;
SOLVE INVENDEMAN USING LP MAXIMIZING PROFIT;
```

7.A.2 The GAMS model of stochastic inventory problem

```
SETS
  S     scenarios      /h,l,m,a/
  T     time periods   /0,1,2,3,4,5/
  INDEX(T)             /  1,2,3,4,5/
  OPENING(T)           /0          /
  TERMINAL(T)          /          5/;
PARAMETERS
  P(INDEX)    market price
               /1   75
                2   65
                3   89
                4   77
                5   80/
  PROB(S)     scenario probabilities
               /h   0.25
                l   0.25
                m   0.25
                a   0.25/
TABLE
  D(T,S)      market demand
                      h     l     m     a
                0     0     0     0     0
                1     0     0     0     0
                2   200   200   150   150
                3   300   250   250   200
                4   400   400   400   400
                5     0     0     0     0
VARIABLES
  PROFIT
POSITIVE VARIABLES
  XMINUS(T,S)    quantity sold at time T
```

```
  XPLUS(T,S)     quantity bought at time T
  I(T,S)         inventory held at time T;
EQUATIONS
  OBJECTIVE      calculating net profit
  INVBAL(T,S)    inventory balance at time T
  AFI0SS1
  AFI0SS2
  AFI0SS3
  AFXM0SS1
  AFXM0SS2
  AFXM0SS3
  AFXP0SS1
  AFXP0SS2
  AFXP0SS3
  AFI1SS1
  AFI1SS2
  AFI1SS3
  AFXM1SS1
  AFXM1SS2
  AFXM1SS3
  AFXP1SS1
  AFXP1SS2
  AFXP1SS3
  AFI2SS1
  AFI2SS2
  AFXM2SS1
  AFXM2SS2
  AFXP2SS1
  AFXP2SS2 ;
OBJECTIVE..    PROFIT =E= SUM((INDEX,S),PROB(S)*((P(INDEX)-2.0)*XMINUS(INDE
             -(P(INDEX)+2.0)*XPLUS(INDEX,S)-H*I(INDEX-1,S)));
INVBAL(T-1,S).. XMINUS(T,S)-XPLUS(T,S)+I(T,S)-I(T-1,S) =E= -D(T,S);
AFI0SS1..          I("0","h") =E= I("0","1");
AFI0SS2..          I("0","m") =E= I("0","a");
AFI0SS3..          I("0","m") =E= I("0","1");
AFXM0SS1..         XMINUS("0","h") =E= XMINUS("0","1");
AFXM0SS2..         XMINUS("0","m") =E= XMINUS("0","a");
AFXM0SS3..         XMINUS("0","m") =E= XMINUS("0","1");
AFXP0SS1..         XPLUS("0","h")  =E= XPLUS("0","1");
AFXP0SS2..         XPLUS("0","m")  =E= XPLUS("0","a");
AFXP0SS3..         XPLUS("0","m")  =E= XPLUS("0","1");
AFI1SS1..          I("1","h") =E= I("1","1");
AFI1SS2..          I("1","m") =E= I("1","a");
AFI1SS3..          I("1","m") =E= I("1","1");
AFXM1SS1..         XMINUS("1","h") =E= XMINUS("1","1");
AFXM1SS2..         XMINUS("1","m") =E= XMINUS("1","a");
AFXM1SS3..         XMINUS("1","m") =E= XMINUS("1","1");
AFXP1SS1..         XPLUS("1","h")  =E= XPLUS("1","1");
AFXP1SS2..         XPLUS("1","m")  =E= XPLUS("1","a");
AFXP1SS3..         XPLUS("1","m")  =E= XPLUS("1","1");
```

```
AFI2SS1..          I("2","h") =E= I("2","1");
AFI2SS2..          I("2","m") =E= I("2","a");
AFXM2SS1..         XMINUS("2","h") =E= XMINUS("2","1");
AFXM2SS2..         XMINUS("2","m") =E= XMINUS("2","a");
AFXP2SS1..         XPLUS("2","h")  =E= XPLUS("2","1");
AFXP2SS2..         XPLUS("2","m")  =E= XPLUS("2","a");
I.UP(T,S)        = 500;
I.FX(OPENING,S)  = 300;
I.FX(TERMINAL,S) = 300;
MODEL INVENDEMAN/ALL/;
SOLVE INVENDEMAN USING LP MAXIMIZING PROFIT;
```

Acknowledgments

We are grateful to Gus Gassmann for constructive comments resulting in an improved presentation.

Bibliography

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows*, Prentice-Hall, New York, 1993.

[2] J. Birge, M. Dempster, H. Gassmann, E. Gunn, A. King, and S. Wallace, *A standard input format for multiperiod stochastic linear programs*, Math. Program. Soc. Comm. Algorithms Newsletter, 17 (1987), pp. 1–19.

[3] J. Bisschop and R. Entriken, *AIMMS: The Modeling System*, Paragon Decision Technology, 1993.

[4] J. Blomvall and P. O. Lindberg, *A Riccati-based primal interior point solver for multistage stochastic programming—extensions*, Optim. Methods Softw., 17 (2000), pp. 383–407.

[5] A. Brooke, D. Kendrick, and A. Meeraus, *GAMS: A User's Guide*, The Scientific Press, Redwood City, CA, 1992.

[6] C. Buchanan, K. McKinnon, and G. Skondras, *The recursive definition of stochastic linear programming problems within an algebraic modeling language*, Ann. Oper. Res., 104 (2001), pp. 15–32.

[7] D. Chang and E. Fragnière, *SPLITDAT and DECOMP: Two New GAMS I/O Subroutines to Handle Mathematical Programming Problems with an Automated Decomposition Procedure*, Department of Operations Research, Stanford University, Stanford, CA, 1996.

[8] C. Condevaux-Lanloy, E. Fragnière, and A. J. King, SISP: Simplified interface for stochastic programming: Establishing a hard link between mathematical programming modeling languages and SMPS codes. Optim. Methods Softw., 17 (2002), pp. 423–443.

[9] G. B. DANTZIG, *Linear programming under uncertainty*, Management Sci., 1 (1995), pp. 197–206.

[10] R. ENTRIKEN, *Language constructs for modeling stochastic linear programs*, Ann. Oper. Res., 104 (2001), pp. 49–66.

[11] R. FOURER AND D. GAY, *Proposals for Stochastic Programming in the AMPL Modeling Language*, International Symposium on Mathematical Programming, Lausanne, Switzerland, 1997.

[12] R. FOURER AND D. GAY, IMPLEMENTING ALGORITHMS THROUGH AMPL SCRIPTS (LOOPING AND TESTING 2), 1999; available online from http://www.ampl.com/cm/cs/what/ampl/NEW/LOOP2/index.html.

[13] R. FOURER, D. GAY, AND B. W. KERNIGHAN, *AMPL: A Modeling Language for Mathematical Programming*, The Scientific Press, San Francisco, 1993.

[14] E. FRAGNIÈRE AND J. GONDZIO, *Optimization modeling languages*, in Handbook of Applied Optimization, P. Pardalos and M. Resende, eds., Oxford University Press, New York, 2002, pp. 993–1007.

[15] E. FRAGNIÈRE, J. GONDZIO, R. SARKISSIAN, AND J.-P. VIAL, *Structure exploiting tool in algebraic modeling languages*, Management Sci., 46 (2000), pp. 1145–1158.

[16] E. FRAGNIÈRE, J. GONDZIO, AND J.-P. VIAL, *Building and solving large-scale stochastic programs on an affordable distributed computing system*, Ann. Oper. Res., 99 (2000), pp. 167–187.

[17] H. GASSMANN AND A. IRELAND, *Scenario formulation in an algebraic modelling language*, Ann. Oper. Res., 59 (1995), pp. 45–75.

[18] H. GASSMANN AND A. IRELAND, *On the automatic formulation of stochastic linear programs*, Ann. Oper. Res., 64 (1996), pp. 83–112.

[19] J. GONDZIO AND R. KOUWENBERG, *High performance computing for asset liability management*, Oper. Res., 49 (2001), pp. 879–891.

[20] R. KOUWENBERG, *LEQGEN: A C-Tool for Generating Linear and Quadratic Programs, User's Manual*, Econometric Institute, Erasmus University, Rotterdam, The Netherlands, 1999.

[21] A. LEUBA AND D. MORTON, *Generating Stochastic Linear Programs in S-MPS Format with GAMS*, INFORMS Conference, Atlanta, 1999.

[22] J. LINDEROTH AND S. J. WRIGHT, *Decomposition algorithms for stochastic programming on a computational grid*, Comput. Optim. Appl., 24 (2003), pp. 207–250.

[23] E. MESSINA AND G. MITRA, *Modelling and analysis of multistage stochastic programming problems: A software environment*, Eur. J. Oper. Res., 101 (1997), pp. 343–359.

[24] G. THOMPSON, *Computational Economics*, Scientific Press, New York, 1992.

[25] R. M. VAN SLYKE AND R. WETS, *L-shaped linear programs with applications to optimal control and stochastic programming*, SIAM J. Appl. Math., 17 (1969), pp. 638–663.

[26] D. YANG AND S. A. ZENIOS, *A scalable parallel interior point algorithm for stochastic linear programming and robust optimization*, Comput. Optim. Appl., 7 (1997), pp. 143–158.

This page intentionally left blank

Chapter 8

A Stochastic Programming Integrated Environment

P. Valente, G. Mitra,* and C. A. Poojari**

8.1 Introduction and background

Advances in hardware, software techniques, and solution methods have made stochastic programming (SP) a viable optimization tool. In the field of linear programming (LP) and integer programming (IP), algebraic modeling languages (AMLs) are well established as aids to prototyping and have led to considerable gains in modeling productivity [7]. Unfortunately, there are not many modeling systems and AMLs which support the creation and investigation of SP models. In this chapter, we address some of the challenges related to the investigation of SP models using the existing software tools and introduce SPInE (stochastic programming integrated environment). The original design of SPInE [15] has been revised and updated. Our design objective is to create a flexible software system based on AMLs which also embeds several solution algorithms. Recently, the AMPL and the MPL modeling systems have been extended to include SPInE's functionalities [22, 23]. The chapter is organized as follows. Section 8.2 introduces the classes of SP models which are supported by our system. In this section we also introduce an example, which is used throughout the chapter to illustrate the features of SPInE. Section 8.3 focuses on SAMPL and SMPL, which are extensions of the AMPL and MPL modeling languages, respectively. In section 8.4 we discuss the rationale underlying the parameter passing interface which connects the special purpose scenario generators to the SPInE system. In section 8.5, we give an overview of the solution algorithms implemented in SPInE and consider the performance and scale-up properties of these algorithms. In section 8.6 we describe the software architecture of SPInE: the illustrative example given in section 8.2 is used to explain the investigation of SP models with the SPInE system. Our aim is to make SPInE

*Centre for the Analysis of Risk and Optimisation Modelling Applications (CARISMA), Brunel University, West London, UK (masrppv@brunel.ac.uk, mastggm@brunel.ac.uk, mapgcap@brunel.ac.uk).

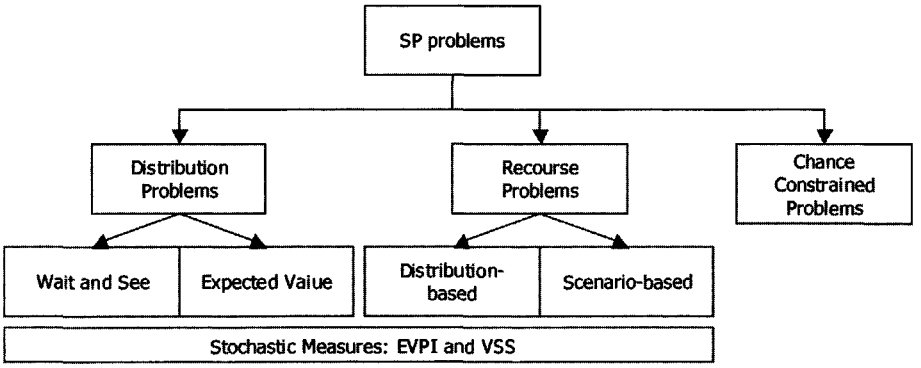


Figure 8.1. *Taxonomy of SP problems.*

widely available to the industrial and academic research community.

8.2 SP problems supported by SPInE

We follow the classification of SP problems introduced by [11]. We make a small extension of their categorization by adding the expected value models as a subclass of the distribution problems, leading to a working taxonomy shown in Figure 8.1.

8.2.1 Illustrative example: An asset and liability management model

We use an asset and liability management (ALM) multistage SP model with downside risk constraints [13] to illustrate the capabilities of the SPInE system. We first state the problem and subsequently formulate the corresponding algebraic SP model with recourse.

The ALM problem: Algebraic formulation

An investor faces the problem of creating a portfolio allocating assets out of a universe of assets. Each asset is characterized by a price, which is the only random variable. The possible future prices are represented by an event tree. The goal of the investor is to maximize the portfolio wealth at the end of the time horizon T. He needs to take into account future obligations (liabilities). Asset buying and selling decisions are made, and each trade has an associated transaction cost. The deviation of the portfolio value from a predefined target is taken as the measure of the risk. In each time stage the investor can decide the amount of assets to buy, sell, and hold in the portfolio. We implement this problem as a multistage SP with recourse, using a split-variable deterministic equivalent representation. The resulting model is set out below.

Sets and indices:

T	denotes the number of time periods in the time horizon,
Assets	is the set of assets in our universe, where | Assets | $= I$,
Scenarios	is the set of scenarios, where | Scenarios | $= Sc$,
$t = 1..T$	denote time periods,
$i = 1..I$	denote different assets,
$s = 1..Sc$	indicate alternative scenarios.

Parameters:

price_{its}	the price of asset i in period t, for scenario s ($i \in$ Assets, $t = 1..T$, $s \in$ Scenarios),
p_s	the weight (probability) associated with scenario s (where $s \in$ Scenarios),
$L_t \geq 0$	the expected liability at time period t ($t = 1..T$),
$F_t \geq 0$	the funding available in time period t ($t = 1..T$),
$A_t > 0$	the predefined target for time period t ($t = 1..T$),
$H0_i \geq 0$	the initial composition of the portfolio ($i \in$ Assets),
$R \geq 0$	the maximum deviation from the target accepted by the investor (in fraction),
$g \geq 0$	the transaction cost rate.

Decision variables:

H_{its}	the amount of assets of type i held in time period t under scenario s ($i \in$ Assets, $t = 1..T$, $s \in$ Scenarios),
B_{its}	the amount of assets of type i bought in time period t under scenario s ($i \in$ Assets, $t = 1..T$, $s \in$ Scenarios),
S_{its}	the amount of assets of type i sold in time period t under scenario s ($i \in$ Assets, $t = 1..T$, $s \in$ Scenarios).

Objective function: Maximize the expected value of the final portfolio wealth:

$$\max \sum_{s=1}^{Sc} p_s \sum_{i=1}^{I} \text{price}_{iTs} H_{iTs}. \tag{8.1}$$

Subject to: Asset holding constraints:

$$H_{its} = H0_i + B_{its} - S_{its}, \quad t = 1, i = 1..I, s = 1..Sc, \tag{8.2}$$

$$H_{its} = H_{it-1s} + B_{its} - S_{its}, \quad t > 1, i = 1..I, s = 1..Sc. \tag{8.3}$$

Fund Balance constraints:

$$(1-g)\sum_{i=1}^{I} \text{price}_{its} S_{its} - L_t + F_t = (1+g)\sum_{i=1}^{I} \text{price}_{its} B_{its}, \quad t = 1..T, s = 1..Sc. \tag{8.4}$$

Downside risk constraint:

$$A_t - \sum_{i=1}^{I} \text{price}_{its} H_{its} \leq A_t R, \quad t = 2..T, s = 1..Sc. \tag{8.5}$$

Table 8.1. *Formulation of the ALM model in AMPL.*

```
# model ALM deterministic equivalent (split-variables)

set I   := 1..23;       # asset type
set T   := 1..4;        # time stages
set Sc  := 1..64;       # scenarios

param g := 0.025;       # Transactions cost ratio
param R := 0.2;         # Risk level;
param L{T};             # Liabilities;
param H0{I};            # Initial portfolio;
param F{T};             # Funding
param A{T};             # Targets
param P{Sc};            # scenario probabilities
param price{T,I,Sc};    # asset prices

var S{T,I,Sc} >=0;
var H{T,I,Sc} >=0;
var B{T,I,Sc} >=0;

maximize wealth : sum{s in Sc, i in I} P[s]* H[4,i,s]* price[4,i,s];

subject to

###### ASSET HOLDING CONSTRAINTS #######
assetholding1{i in I, s in Sc}:          H[1,i,s]=H0[i]+B[1,i,s]-S[1,i,s];
assetholding2{i in I, t in 2..4,s in Sc}: H[t,i,s]=H[t-1,i,s]+B[t,i,s]-S[t,i,s];

###### FUND BALANCE CONSTRAINTS #######
fundbalance{t in T,s in Sc}:    sum {i in I} B[t,i,s]*price[t,i,s]*(1+g) -
                                sum {i in I} S[t,i,s]*price[t,i,s]*(1-g) =
                                F[t]-L[t];

###### DOWNSIDE RISK CONSTRAINT ######
zeta{ t in 2..4, s in Sc}:      A[t]- sum {i in I} H[t,i,s]*price[t,i,s]<= R*A[t];

###### NON ANTICIPATIVITY CONSTRAINT ######
nah11{i in I,s in 2..64}:       H[1,i,s]= H[1,i,s-1];
nah21{i in I,s in 2..8}:        H[2,i,s]= H[2,i,s-1];
nah22{i in I,s in 10..16}:      H[2,i,s]= H[2,i,s-1];
...
nah28{i in I,s in 58..64}:      H[2,i,s]= H[2,i,s-1];
nah31{i in I,s in 2..2}:        H[3,i,s]= H[3,i,s-1];
nah32{i in I,s in 4..4}:        H[3,i,s]= H[3,i,s-1];
...
nah316{i in I,s in 64..64}:     H[3,i,s]= H[3,i,s-1];

#### same for variables S and B ####
...
```

8.2.2 Illustrative model formulated in AMPL and MPL

The implementation of this model in the established modeling language AMPL [9] is set out in Table 8.1. An equivalent formulation in MPL [14] can be found in [24]. The values of the deterministic parameters, as well as the scenario data, are read from a database. In AMPL, the directives for the database connections are separated from the model definition. The AML systems translate these declarative formulations into the matrix of the deterministic equivalent model.

An examination of the nonanticipativity constraints indicates that this explicit form

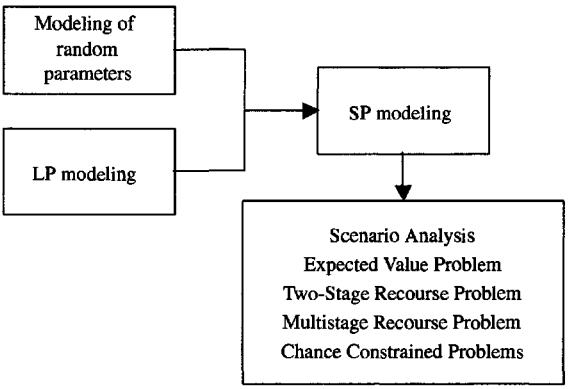


Figure 8.2. *The combined paradigm.*

of representation is not immediately natural: it can be laborious and error-prone. Also, the important requirement of separating the data from the model [12] is lost in the above representation. This can be avoided using alternative but equally laborious model formulations (see, for instance, [10]). Finally, the matrices generated for these deterministic equivalent models contain redundancies and are often too large to be handled by both solvers and modeling systems. This strongly makes the case for extending the AMLs in terms of syntax and structured matrix generation capabilities which together support the SP modeling steps.

8.3 Extending AMLs for SP

Our illustrative example introduced in section 8.2 highlights the difficulties of using existing AMLs to formulate SP. This is mainly due to the lack of constructs for the definition of the randomness of the model coefficients and for the declaration of the scenario tree structure. An examination of other investigators' works reveals that some extensions have been proposed to overcome these limitations but have not yet been deployed. We have designed and adopted a direct approach, whereby we provide extensions to AMLs to formulate SP recourse problems and chance-constrained problems with natural and concise constructs [23]. This approach allows us to extend the syntax of AMPL and MPL into what we call SAMPL and SMPL, respectively.

8.3.1 SAMPL and SMPL: An introduction

A stochastic programming model can be considered as a linear programming model extended and refined by the introduction of uncertainty (see Figure 8.2). More precisely, the underlying LP optimization model is extended by taking into account the probability distribution of some of the LP model coefficients which are random variables. Such distributions are provided by models of randomness (implemented in scenario generators), which are specific to the particular optimization problems under investigation.

As we consider the taxonomy and the classes of stochastic models introduced in

Figure 8.1, it becomes immediately obvious that AMLs are neither specifically designed nor well suited to construct these classes of models. In fact, the strong coupling between the model structure and the data structure which arises in the models of randomness makes it very difficult to separate model definition from data definition. If we consider discrete probability distributions of the random parameters, then it is always possible to define a deterministic equivalent model for SP problems with recourse. This approach suffers from a number of drawbacks. The difficulties of working with the deterministic equivalent representation are summarized below:

1. If a split-variable representation is used, the nonanticipativity constraints have to be explicitly declared to reflect the underlying model structure induced by the scenario tree. This leads to an unnecessary replication of decision variables and constraints. Although the modern solvers will remove these replicated variables, the model representation may contain high levels of redundancy.

2. In general, the size of the deterministic equivalent depends on the number of scenarios and stages considered in the event tree. For instance, if the number of realizations per stage is constant, the size of the deterministic equivalent increases exponentially with the number of stages. This often leads to unmanageable models when the size of the scenario tree increases.

3. The inherent staircase structure of any SP model is normally lost when modeling systems calculate the matrix of the deterministic equivalent. Additional processing may be required to recapture and exploit this structure in the solution algorithms.

Ideally, a modeling language for SP problems should include a set of constructs which allow the modeler to capture the effects induced by the uncertainty on the underlying model structure as well as provide a compact representation of the model instance. We have developed a generic approach in which we first define the underlying deterministic model and then introduce the stochastic information in respect of the random parameters. Broadly speaking this can be viewed as a stochastic extension of AML constructs.

The underlying deterministic model

In an SP problem where the event tree does not include "coffin" or "trap" states [11], it is always possible to identify an underlying deterministic model (also called the core model). This model captures the logical structure of the problem as well as the dynamical relations within decision variables, their bounds, and the objective function. The definition of the underlying deterministic model makes use of the standard constructs provided by the existing modeling languages (AMPL or MPL).

Declaration of the random structure

On implementing the underlying deterministic problem, the next step is to embed the information related to the model of randomness which characterizes the problem. We extend the language syntax to capture such stochastic information. The necessary items of information are as follows:

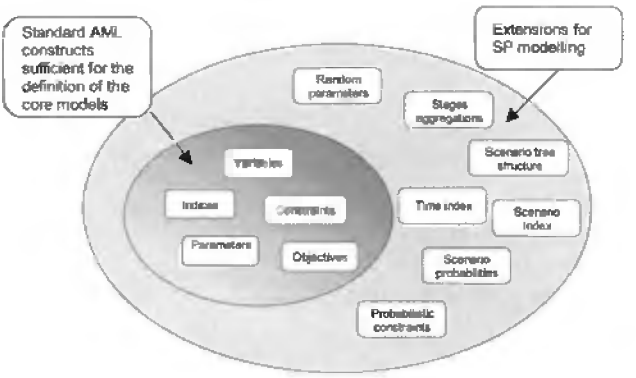


Figure 8.3. *Extended language constructs.*

Time dimension: defines the set used to describe the temporal horizon in the underlying model. This needs to be uniquely identified.

Stages: decision stages are defined in terms of a partition of the time horizon.

Scenario dimension: the set used to identify the scenarios needs to be uniquely identified, because the realizations of the random parameters in scenario-based problems are defined using this index.

Scenario tree: in scenario-based problems, it represents the structure of the event tree.

Scenario probabilities: the (discrete) probabilities associated with the scenarios.

Random parameters: defines and marks the random parameters of the problem in scenario-based problems.

Probabilistic constraints: probabilistic constraints in chance-constrained problems need to be explicitly declared.

Figure 8.3 shows how the basic constructs of a modeling language for linear programming are extended to capture the stochastic information. The design of the new constructs is adapted to be consistent with the grammar of the underlying modeling language. We have successfully applied this approach to the AMPL and MPL languages. See [23] for a detailed description of the syntax. The combined SAMPL and SMPL parser is the core of the modeling system embedded into SPInE. This system also generates data model instances in standard SMPS format and in a stochastic intermediate representation (SIR).

8.3.2 Illustrative model formulated in SAMPL and SMPL

Following the approach outlined in the previous section, we need to specify the stochastic information relating to the ALM problem. Table 8.2 contains the declarations in SAMPL and SMPL of the components which together comprise the required stochastic information.

Table 8.2. *Definition of the stochastic information of the ALM model.*

Description	SMPL definition	SAMPL definition
Time: The time index of this problem is represented by t.	`TIME T:=1..4;`	`set T:=1...4;`
Scenarios: The asset prices are given in the form of an event tree. We use the index s to identify the scenarios within set *Sc*.	`SCENARIO s:=1..64;`	`scenario set Sc:=1...64;`
Probabilities: We consider scenarios with equal probability.	`PROBABILITIES P[s]:= 1/count(s);`	`probability param P{Sc}:= 1/card(Sc);`
Stages: In this model there is a 1-to-1 relation between time periods and stages. In SMPL we use the *ONE_TO_ONE* keyword to specify this. In SAMPL, we use the suffix *stage* to assign each variable to a stage which is equal to the period t.	`STAGES PARTITION ONE_TO_ONE;`	`Suffix stage LOCAL; var B{t in T,I,Sc}>=0, suffix stage t;`
Tree: The scenario tree used in this model is symmetric. The number of branches at each node varies in different time stages (8, 4, 2), but is constant for nodes within a given stage.	`TREE MULTIBRANCH (8,4,2);`	`tree TR:= multibranch {8,4,2};`
Random data: The only random parameter is the price of the assets over time. This parameter is scenario-dependent and is therefore indexed over the scenario index.	`RANDOM DATA price[i,t,s] =DATABASE (tbl_PricesSP, return);`	`random param price{T,I,Sc};`

The complete model formulation in SAMPL is set out in Table 8.3. The equivalent formulation in SMPL can be found in [24]. An SP presented in SAMPL or SMPL can be separated into two parts. Part 1 contains the declaration of the underlying core LP using "standard" AML statements. Part 2 contains some structural details covering the stochastic aspects of the model. This includes the definition of the scenario tree structure, the partitioning of variables and constraints into stages, and an implicit reference to a scenario generator which provides random data parameter values to instantiate the SP model.

We observe the following:

1. The explicit definition of the nonanticipativity constraints has been eliminated.

2. The separation of data definition from model definition, which is one of the main advantages of the use of AMLs, is preserved. Although in this example the tree

Table 8.3. *Formulation of the ALM model in SAMPL.*

```
# ALM model, sampl version
### STOCHASTIC FRAMEWORK ################
suffix stage LOCAL;
scenarioset Sc := 1..64;
probability param P{Sc} := 1/card(Sc);
random param price{T,I,Sc} ;
tree TR:= multibranch{8,4,2};

### MODEL FORMULATION ###################
set T  := 1..4;          #time horizon
set I  := 1..23;         #asset type

param g := 0.025;        # Transactions cost rate
param R := 0.2;          # Risk level;
param L{T};              # Liabilities;
param H0{I};             # Initial portfolio;
param F{T};              # Funding
param A{T};              # Targets

var S{t in T,I,Sc} >=0,  suffix stage t;
var H{t in T,I,Sc} >=0,  suffix stage t;
var B{t in T,I,Sc} >=0,  suffix stage t;

maximize wealth : sum{s in Sc}P[s]*(sum{i in I} H[4,i,s]*price[4,i,s]);

subject to

###### ASSET HOLDING CONSTRAINTS ########
assetholding1{i in I, s in Sc}:                  H[1,i,s]=H0[i]+B[1,i,s]-S[1,i,s];
assetholding2{i in I, t in 2..4, s in Sc}:       H[t,i,s]=H[t-1,i,s]+B[t,i,s]-S[t,i,s];

###### FUND BALANCE CONSTRAINTS #########
fundbalance{t in T, s in Sc}:          sum {i in I} B[t,i,s]*price[t,i,s]*(1+g) -
                                       sum {i in I} S[t,i,s]*price[t,i,s]*(1-g) =
                                       F[t]-L[t];

###### DOWNSIDE RISK CONSTRAINT #########
zeta{t in 2..4, s in Sc}:              A[t]-sum {i in I} H[t,i,s]*price[t,i,s] <=R*A[t];
```

structure is explicitly stated using the MULTIBRANCH keyword, it can be also imported from the scenario generator as any other data table, thus retaining the data separation.

3. Asymmetric tree structures can also be defined using the SAMPL and SMPL constructs [23].

4. If the model contains multiple random parameters and their event trees are different, it is necessary to first construct a combined tree and then to declare the resulting structure in the tree section.

8.4 Scenario generation

Scenario generators capture the randomness properties of a particular application's domain. Typically, a consumer product supply chain model and an energy distribution model both

require scenarios of forecast demand, but the factors which influence the demands and the forecasting models may be very different. Again in finance applications the asset prices under consideration may be generated using different models of credit risk, interest rate risk, or other considerations. Therefore, in designing SPInE, one of our main goals was to develop an appropriate parameter passing interface which enables the system to connect diverse special-purpose scenario generators.

8.4.1 Integrating scenario generators in SPInE

An event tree within SP serves two purposes:

1. define the model of randomness (scenario generation) and
2. specify the algebraic structure of the decision variables and constraints.

A scenario generator φ captures in a procedural form a domain-specific model of randomness. In general, φ is a function of historical information, an event tree structure, and some other specification parameters. We can thus separate the main groups of parameters as

H:	history,
τ:	event tree structure,
θ:	remaining parameters.

For $\omega \in \Omega$, considering the vectors $\xi(\omega)$ which denote the realizations of the uncertain parameters for a given event ω, we further define the set of all scenarios as Ξ, such that

$$\bigcup_{\omega \in \Omega} \xi(\omega) = \Xi. \tag{8.6}$$

The set of scenarios Ξ is then seen as the collection of scenarios which are output by the generation procedure:

$$\varphi(H, \tau, \theta) \Rightarrow \Xi. \tag{8.7}$$

In the algebraic form of the SP model we also need to specify the "variable and constraint" tree structure, which we label as τ'. Thus using the extended AML, we provide a specification of τ' in the SP model through the tree declaration. For consistency, of course, we need the two trees to be congruent. In other words, we need to ensure that the event tree structure τ used by the special-purpose scenario generator is compatible with the τ' specified in the SP model [23]. The requirement for scenario generator parameter passing and tree consistency conditions are illustrated in Figure 8.4. This figure illustrates how a special-purpose scenario generator is connected to SPInE. The two trees τ and τ' are compared for consistency. The scenario generator then creates the set of scenarios and the associated probabilities $p(\omega)$.

The data interface for the presentation of the scenarios to the modeling system is based on open database connectivity (ODBC) connections. This allows the scenario generator to store the output in virtually any type of database (including text files). The flexible interface with scenario generators and the ability to create in-sample scenarios for SP model optimization and out-of-sample scenarios for simulation (see section 8.6.3) make the connection to external generators a valuable feature of SPInE. A library of scenario generators for diverse application domains is under development [24].

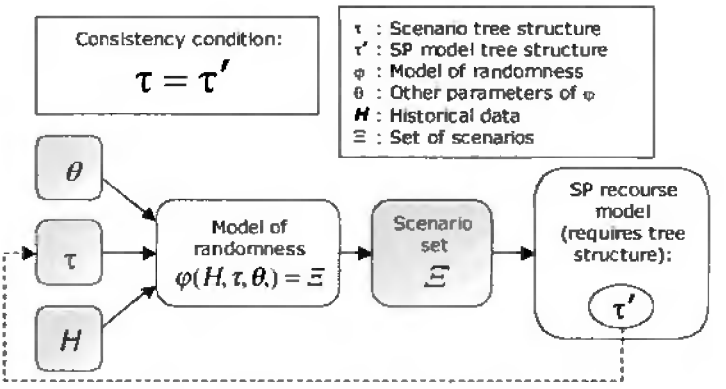


Figure 8.4. *Scenario generation parameters and tree consistency.*

8.5 Stochastic solver in SPInE

The proposal and adoption of the stochastic mathematical programming standard (SMPS) [3], as well as the library of models prepared to this specification and maintained by researchers (see [1] and [4]), has made it easier to develop solvers and evaluate their performance. The solver developed and integrated in SPInE has a clearly set-out coupling with the modeling system and can accept model data instances either in SIR or in standard SMPS format. As a result the interface is extremely flexible, and the solver, which incorporates a number of established and a few innovative algorithms, may be used as a stand-alone tool.

8.5.1 The solver system in SPInE

The solver system which we have embedded in SPInE is designed to variously process the family of SP models, which include the following:

1. expected value problem,
2. wait-and-see problems,
3. here-and-now problem (recourse models).

Hence, the solver also can compute EVPI and VSS for a given model. SP models capture the two important aspects of decision making, namely, time and uncertainty, but their computational realizations suffer from the curse of dimensionality. This in turn requires that the data structure of the solver must be efficient to capture the dynamic evolution of uncertain parameters and scale-up (of the model size through scenarios and stages) to process real-world models.

Algorithms

A number of solution algorithms have been implemented and tested within SPInE. Currently, we use FortMP [8] as the main LP/MIP solver engine, but the design allows us to replace

Table 8.4. *Solution algorithms of the solver system.*

SP Model	Algorithm	Comments
Two-stage linear SP	DEQ explicit	Implemented and tested.
	DEQ implicit	Implemented and tested.
	Benders' decomp.	Implemented and tested.
	Stochastic decomp.	Currently being implemented.
	Benders' importance sampling	Currently being implemented.
Integer two-stage SP	Lagrangian relaxation	This algorithm is described in our working paper [20] and also in our supply chain paper [18].
	Lagrangian relaxation and importance sampling	To be implemented.
Multistage SP	Universe	Implemented and tested.
	Nested Benders'	Implemented and tested.
	Nested Benders' and importance sampling	To be implemented.
	EVPI-based importance sampling	To be implemented. This is based on the work done in [6].

it by any other powerful solver engine, such as CPLEX. Table 8.4 sets out the SP model and solver algorithm combinations which have been tested or planned for inclusion within SPInE.

Control

The solution algorithms to be deployed are chosen by control parameters specified via a parameter file. The full details of these parameters and the parameter file can be found in [22]; examples of setting these controls are shown in section 8.6.2. The solver can be also deployed to process the expected value problem and the wait-and-see problems for all scenario instances. Furthermore, there exist control switches to calculate the measures EVPI and VSS.

8.5.2 Quality assurance/benchmark and scale-up properties

The embedded SP solver module has been extensively tested for a wide range of quality assurance (QA) test problems which have been collected from a number of sources. A summary of these QA models is given in [24]; a paper describing the algorithm and computational performance of the SP solver engine is under preparation [20]. The following five algorithms make up the most important part of the SP solver engine:

Table 8.5. *Models summary.*

Model	Rows	Columns	Nonzeroes
Pgp2	9	20	40
FXM	330	457	2589
Pltexp	270	732	1491
Storm	713	1380	4037
Phone	24	93	207

1. deterministic equivalent split-variable (DEQ explicit): model processed by IPM;
2. deterministic equivalent compact variable (DEQ implicit): model processed by SSX;
3. nested Benders' decomposition: the master and subproblems processed by SSX;
4. wait-and-see: the individual problems processed by SSX;
5. expected value: the expected-value LP problem processed by SSX.

The computational platform used for the benchmarks is based on a Pentium III, 1 GHz with 512 Mbyte RAM and FortMP solver compiled using Digital Fortran running under Win NT. Table 8.5 displays the summary of a subset of test models taken from this QA set.

The relative performances of the three here-and-now algorithms (Benders' decomposition, DEQ implicit, and DEQ explicit) and the performances and solutions of the expected value and wait-and-see problems are shown in Table 8.6. We also include the EVPI and VSS values. In these tables and figures, we use the following legend:

NB:	Nested Benders' decomposition
DI:	DEQ implicit
DE:	DEQ explicit
WS:	Wait-and-see
EV:	Expected value
NEM:	Not enough memory

To study the scale-up properties of these algorithms we investigated the STORM model. STORM is a two-period freight-scheduling problem described in [17]; the problem is held at the University of Michigan and provided by Adam Berger. This model was investigated for progressively larger sizes, from 1 to 1000 scenarios. The corresponding processing times (in seconds) are set out in Table 8.7 and are also plotted separately in Figure 8.5 to a log (time) versus linear (number of scenarios) scale.

Benders' decomposition extends well for parallel implementation [2, 16, 25, 26]. Our solver, FortSP [20], is available on the NEOS [5] and OSP-CRAFT domains (www.neos.mcs.anl.gov/neos and www.osp-craft.com, respectively). Our earlier experience of solving large SP models on a parallel platform using client-server architecture is described in [18].

Table 8.6. *Computational results.*

Description		SP models			Stochastic	Metric
Name	**Stage/ Scenario**	**HN (time DI/DE/NB)**	**WS (time)**	**EV (time)**	**EVPI**	**VSS**
Pgp2	2/8	508.975 (1s/1s/1s)	449.844 (1s)	431.407 (1s)	59.13	∞
Fxm	2/16	18417.1 (24s/47s/46s)	18417.1 (4s)	18416.8 (1s)	0.06	0.06
Pltexp	3/36	-13.968 (28s/59s/2s)	-13.9135 (1s)	-14.2801 (1s)	0.0166	∞
Storm	2/8	15535210 (20s/25s/12s)	15488580 (3s)	15459240 (1s)	46637.65	∞
	2/27	15508969 (272s/335s/36s)	15476546 (3s)	15459253 (1s)	32422	∞
	2/125	15512048 (7756s/12228/176s)	15476584 (36s)	15459219 (1s)	52829	∞
	2/1000	15802505 (NEM/NEM/1141s)	15766791 (297s)	15750516 (1s)	35714	∞
Phone	2/32768	36.9 (NEM/NEM/2244s)	36.9 (117s)	36.9 (1s)	0	∞

Table 8.7. *Performance time (in seconds) for algorithms on the STORM model.*

Scenarios	DI	DE	WS	NB
1	1	1	1	3
10	28	40	3	25
25	205	320	8	54
50	887	1108	17	92
100	4012	5014	33	199
200	18152	22685	67	372
550	∞	∞	183	881
1000			297	1141

8.6 SPInE

We have revised the design of the first SPInE prototype [15], adding several features, such as support of the SAMPL and SMPL extended languages, the development of a new solver for SP multistage recourse problems, and integration with scenario generators. In section 8.6.1 we provide an overview of the software architecture. In section 8.6.2 we describe the SPInE menu options and controls by means of a few example screen shots taken from the investigation of the ALM model illustrated earlier in the chapter. In section 8.6.3 we show

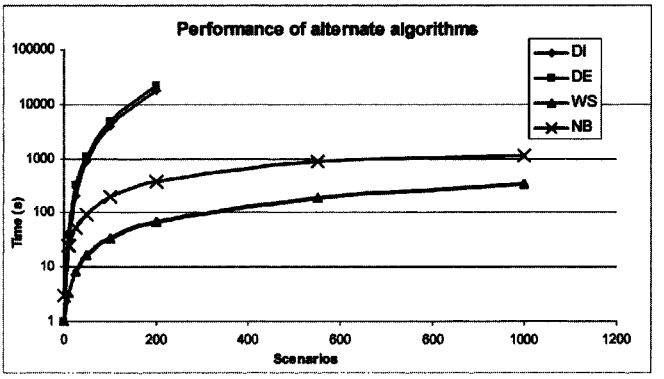


Figure 8.5. *Performance of the algorithms on the STORM model.*

how the system can be used, through the spreadsheet data exchange feature, to compute value at risk for the first-stage decisions.

8.6.1 Software architecture: An overview

The new SPInE environment integrates a number of subsystems which are managed by a control system. The subsystems as such are software components which may be used to create embedded applications.

The diagram in Figure 8.6 illustrates the architecture of SPInE and the interaction of the modules which together make up the software environment. SPInE is divided into four main subsystems, namely, scenario generation, modeling, solver, results analysis, and the overarching control module.

Scenario generation

SPInE is designed to interface with scenario generators which supply the scenario data in ODBC databases or text files. An important aspect of the scenario generation interface is to establish the consistency between the SP model tree τ' and the data path tree τ underlying the scenario generation (see section 8.4).

Modeling subsystem

The modeling subsystem is designed to support the language extensions SAMPL and SMPL introduced in section 8.3. The software module called stochastic program generator (SPG) combines two separate parsers and a matrix generator. SPG processes together the algebraic models and the scenario data set to create an instance of the model in either SMPS format or in SIR. The SPG module makes use of an underlying modeling engine, specifically, OptiMax or MPL for the support of SMPL models and a comparable AMPL-based COM object, developed by our research group, for models prepared in SAMPL. The modeling system interacts directly with the scenario generators for the stochastic data and connects

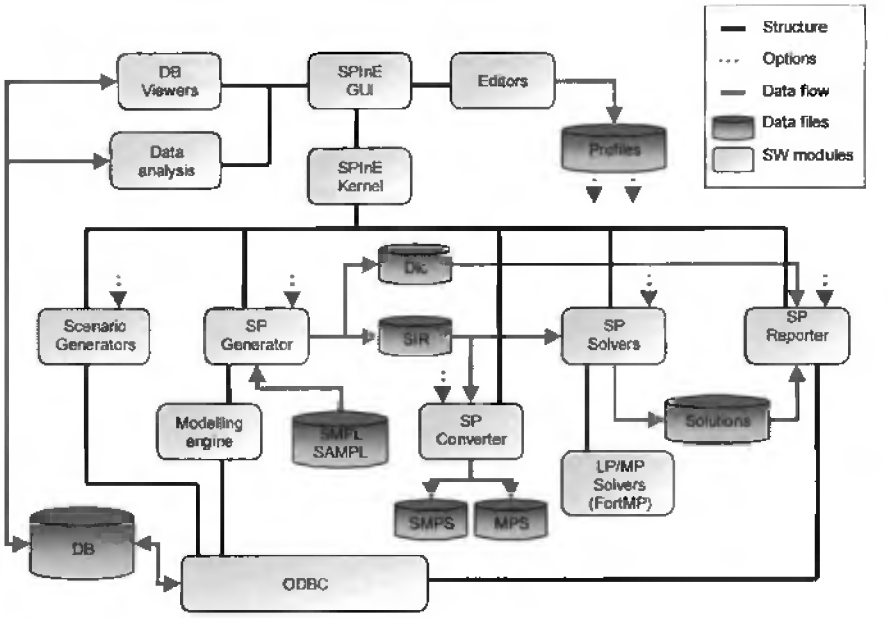


Figure 8.6. *Software architecture of SPInE.*

to the database systems which maintain the deterministic data relating to the core model of the SP problem.

Solver subsystem

Given a stochastic programming problem with recourse, the stochastic solver embedded in SPInE provides the solution to three related classes of models:

1. here-and-now,
2. scenario analysis (wait-and-see), and
3. expected value.

The solver is also able to report the stochastic measures EVPI and VSS. For a detailed description of the solver's features, see section 8.5. Other solvers which accept SMPS input can easily be connected to the system.

Results analysis

A critical phase in the development of stochastic models is the analysis of the solutions. The integration with database systems enables the exploitation of data manipulation languages (DML), which usually accompany the DBMS for the development of customized viewers and advanced data analysis tools. The SP reporter (SPR) module of SPInE allows the user to export solution vectors using standard ODBC or using text files. The volume of the

solution results produced by the stochastic solver can be very large. The investigator might be interested only in a subset of the solutions (e.g., the first-stage strategic decisions). SPR provides filtering functionality which is used to transfer only the relevant decision data to the DBMS.

Control module and graphical user interface

Each module in SPInE can be run as an independent application through script files. A control module including a graphical user interface (GUI) has been developed and can be used to investigate SP problems. The GUI makes use of standard Windows objects to display and control hierarchical structures. The main subsystems of SPInE, namely, the SP instances generator SPG, the stochastic solver SPS, and the solution reporter SPR, have also been wrapped in a dynamic link library (DLL), which enables the rapid development of embedded applications. This DLL has been successfully integrated with the MPL modeling system, leading to the MPL/SPInE environment [22], which combines MPL's GUI with the SPInE stochastic programming capabilities.

8.6.2 Using SPInE: Commands and controls

A sequence of control commands and dialog boxes are annotated below to illustrate a simple use of SPInE. After activating SPInE, the main window appears, as shown in Figure 8.7. The menu bar and the alternative commands are also displayed.

The main menu items are described as follows:

1. Options: this menu item is used to specify generator and solver settings;
2. Run: this item enables the user to first parse the model, generate the SP instance, solve the model, and finally export the results;
3. Model: this menu item allows the user to view the scenario tree and to edit controls manually.

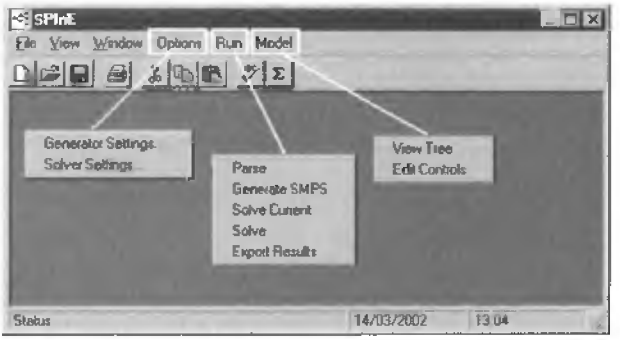


Figure 8.7. *SPInE's menu commands.*

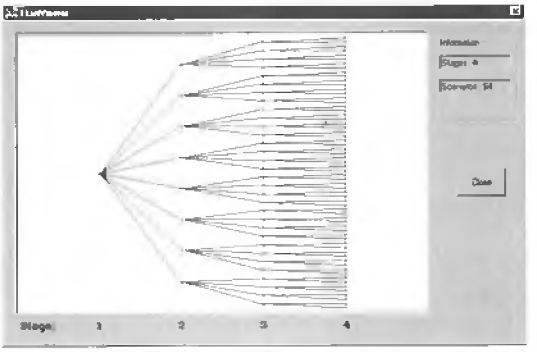


Figure 8.8. *View of the scenario tree.*

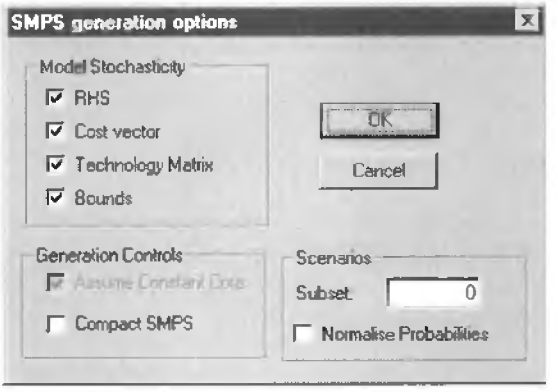


Figure 8.9. *SMPS generation controls.*

For the example ALM model, the View Tree command after Parse leads to the display shown in Figure 8.8.

After the model has been parsed, the SMPS generation is controlled by the options shown in the dialog box of Figure 8.9. If the user knows a priori which parts of the model's matrix contain random coefficients, the generation can be speeded up by instructing the system to skip the processing of the remaining (constant) parts.

The solver execution on the different related models (here-and-now, expected value, wait-and-see) is controlled by the options shown in the dialog box set out in Figure 8.10.

See [22] for a comprehensive guide to SPInE's settings and usage.

8.6.3 Value at risk computation

SPInE can be also used to undertake more advanced investigation of an SP model. In many applications, quantification of the risk associated with a decision is becoming an important modeling issue. In general, value at risk (VaR) as a metric for computing risk has become widely accepted, particularly by the finance community [21]. SPInE can be used to interact

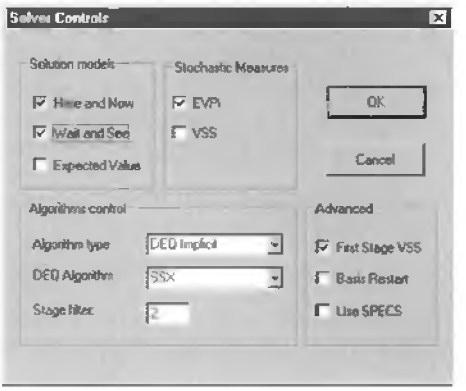


Figure 8.10. *SP solver controls.*

with Excel spreadsheets and can produce VaR metrics for any given first-stage decision using either in-sample or out-of-sample scenarios. For a given model, we can easily compute and compare the VaR for the optimum first-stage decisions x^*_{HN} given by the here-and-now solution and the optimum first-stage decisions x^*_{EV} given by the expected value LP solution. These solution vectors are imported into Excel (see Figure 8.11) and are supplied as fixed values to a scenario analysis model which is used to simulate their performance.

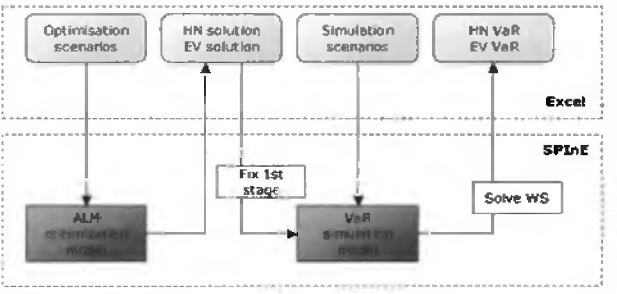


Figure 8.11. *Computation of VaR using SPInE and spreadsheets.*

8.7 Future work and conclusions

Modeling and solving optimum decision problems under uncertainty is a challenging task. We have highlighted the need for an integrated environment for modeling and solving SP recourse problems. In this chapter, we introduced SPInE. This system provides practitioners with a powerful modeling system, based on two language extensions (SAMPL and SMPL) specifically designed for the definition of SP models. We have illustrated an asset and liability management model, and we have used SAMPL and SMPL to formulate the corresponding multistage SP model. The interface designed for the connection to the

scenario generators allows us to bring together the models of randomness with algebraic optimization models. The variety of solution algorithms embedded in our SP solver engine and the ability to connect to databases for the analysis of the results make SPInE a complete and flexible tool for the implementation and investigation of SP problems. We identify, however, a number of research issues which need further investigation:

1. Connecting specific scenario generators to the SP models remains an open issue. Indeed, in the generated model we work with a snapshot of the dynamic model.
2. Although SPInE's stochastic solver is able to solve distribution-based recourse problems, the language extensions SMPL and SAMPL are not designed to deal with such models. This issue is being investigated and further extensions planned.
3. A framework for the validation of the first-stage decisions through simulation as well as back testing has been recently implemented and is undergoing extensive testing [24].
4. We aim to develop a solver which can process quadratic inequalities. Thus a range of chance-constrained models can be also processed by the system.
5. We have earlier experience of parallelizing particular instances of SP models [16]. We wish to extend this work to include parallel implementations of Benders' as well as a Lagrangian relaxation-based integer SP solver [19].

Acknowledgments

The authors acknowledge financial support provided by UNILEVER Research and by the EU research contracts ESPRIT-RTD-26267 (Schumann project) and IST-1999-56410 (OSP-CRAFT project). Dr. T. Kyriakis supplied us the scenario generator used in the example problem and E. F. D. Ellison worked closely with us in the performance evaluation and testing of the stochastic solver.

Bibliography

[1] K. A. Ariyawansa and A. J. Felt, *On a New Collection of Stochastic Linear Programming Test Problems*, Preprint, 2001; available online from http://www.optimization-online.org.

[2] K. A. Ariyawansa and D. D. Hudson, *Performance of a benchmark parallel implementation of the Van Slyke and Wets algorithm for two-stage stochastic programs on the Sequent/Balance*, Concurrency Practice Experience, 3 (1991), pp. 109–128.

[3] J. R. Birge, M. A. H. Dempster, H. I. Gassmann, A. J. King, and S. W. Wallace, *A standard input format for stochastic linear programs*, COAL Newsletter, 17 (1987), pp. 1–20.

[4] J. R. Birge and D. Holmes, *A POrtable Stochastic Programming Test Set POSTS*, 2001; available online from http://users.iems.nwu.edu/~jrbirge//html/dholmes/post.html.

[5] J. Czyzyk, M. P. Mesnier, and J. J. Moré, *The NEOS server*, IEEE J. Comput. Science Engrg., 5 (1998), pp. 68–75.

[6] M. A. H. Dempster and R. T. Thompson, *EVPI-based importance sampling solution procedures for multistage stochastic linear programmes on parallel MIMD architectures*, Ann. Oper. Res., 90 (1999), pp. 161–184.

[7] B. Dominguez-Ballesteros, G. Mitra, C. A. Lucas, and N.-S. Koutsoukis, *Modelling and solving environments for mathematical programming: A comparative review and new directions*, J. Oper. Res. Soc., 53 (2002), pp. 1072–1092.

[8] E. F. D. Ellison, M. Hajian, R. Levkovitz, I. Maros, G. Mitra, and D. Sayers, *FortMP Manual Version* 3.02, OptiRisk Systems and Brunel University, West London, 2000.

[9] R. Fourer, D. M. Gay, and B. W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, Second edition, Duxbury Press/Brooks/Cole, Pacific Grove, CA, 2002.

[10] H. I. Gassmann and A. M. Ireland, *Scenario formulation in an algebraic modelling language*, Ann. Oper. Res., 59 (1995), pp. 45–75.

[11] H. I. Gassmann and A. M. Ireland, *On the formulation of stochastic linear programs using algebraic modelling languages*, Ann. Oper. Res., 64 (1996), pp. 83–112.

[12] A. M. Geoffrion, *Indexing in modelling languages for mathematical programming*, Management Sci., 38 (1992), pp. 325–344.

[13] T. Kyriakis, *Asset and Liability Management with Uncertainty and Risk*, Ph.D. thesis, Brunel University, West London, UK, 2002.

[14] *MPL Modelling System, Release* 4.11, Maximal Software, Arlington, VA, 2000.

[15] E. Messina and G. Mitra, *Modelling and analysis of multistage stochastic programming problems: A software environment*, Eur. J. Oper. Res., 101 (1997), pp. 343–359.

[16] S. A. MirHassani, C. A. Lucas, G. Mitra, E. Messina, and C. A. Poojari, *Computational solution of capacity planning models under uncertainty*, Parallel Comput., 26 (2000), pp. 511–538.

[17] J. M. Mulvey and A. Ruszczynski, *A new scenario decomposition method for large scale stochastic optimisation*, Oper. Res., 43 (1995), pp. 477–490.

[18] C. A. Poojari, C. A. Lucas, and G. Mitra, *A Heuristic Technique for Solving Stochastic Integer Programming Models—a Supply Chain Application*, CARISMA Technical Report CTR/29/04, Department of Mathematical Sciences, Brunel University, West London, UK, 2004.

[19] C. A. Poojari and G. Mitra, *A Solution Technique for Two-Stage Stochastic Programs with First Stage Integer Variables*, presented at the Ninth International Conference on Stochastic Programming, Berlin, August, 2001.

[20] C. A. POOJARI AND G. MITRA, *A Multistage Stochastic Programming Solver: FortSP*, CARISMA Technical Report CTR/30/04, Department of Mathematical Sciences, Brunel University, West London, UK, 2004.

[21] R. T. ROCKAFELLAR AND S. URYASEV, *Optimization of conditional value at risk*, J. Risk, 2 (2000), pp. 21–41.

[22] P. VALENTE, *SPInE User Manual*, OptiRisk Systems and Maximal Software, London, 2002.

[23] P. VALENTE, R. FOURER, G. MITRA, AND M. SADKI, *Extending Algebraic Modelling Languages for Stochastic Programming*, CARISMA Technical Report CTR/09/03, Department of Mathematical Sciences, Brunel University, West London, UK, 2003.

[24] P. VALENTE, N. DI. DOMENICA, G. BIRBILIS, AND G. MITRA, *Stochastic Programming and Scenario Generation within a Simulation Framework: An Information System Perspective*, CARISMA Technical Report CTR/26/04, Department of Mathematical Sciences, Brunel University, West London, UK, 2004.

[25] S. J. WRIGHT, *Solving optimization problems on computational grids*, Optima Mathematical Programming Society Newsletter, 2001.

[26] S. A. ZENIOS AND Y. CENSOR, *Parallel Optimization: Theory, Algorithms, and Applications*, Oxford University Press, Oxford, UK, 1997.

Chapter 9

Stochastic Modeling and Optimization Using STOCHASTICS

M. A. H. Dempster, J. E. Scott,* and G. W. P. Thompson**

9.1 Introduction

Algebraic modeling languages have greatly simplified the formulation and management of deterministic mathematical programming problems, but as yet none of these languages provide any explicit support for the specification of dynamic stochastic programming problems (DSPs). Nevertheless, it is possible to describe a DSP with only the constructs available in deterministic languages using a so-called nodal formulation, and in this chapter we show how this can be done for a simple example problem. There are, however, several situations when the nodal formulation becomes a limitation. Realistic stochastic programming problems may have very large deterministic equivalents, and if we can avoid it we do not wish to instantiate these in full at any point during the modeling process. Also, DSPs tend to have a highly repetitive structure, and it is worth going to some effort to exploit this for efficient problem generation. Finally the "algorithm's form" of a stochastic programming problem is different from that of its deterministic equivalent, and the modeling system should take this into account. Addressing these points, we describe `stochgen`, the stochastic modeling component of the STOCHASTICS system, which works in conjunction with either AMPL or XPRESS-MP and allows the efficient generation of large-scale stochastic programming problems. We give some details of such problems and describe `solgen`, an implementation of nested Benders decomposition which works either independently of or in conjunction with `stochgen` and has been used to solve a variety of real-world problems. We then discuss how visualization tools can be used to aid the DSP modeling process, and we set out the progress we have made toward an integrated stochastic programming environment in the development of STOCHASTICS.

*Centre for Financial Research, Judge Institute of Management, University of Cambridge, Cambridge, UK, and Cambridge Systems Associates Limited (mahd2@cam.ac.uk, jes23@cam.ac.uk, gwpt1@cam.ac.uk).

9.2 Dynamic stochastic programming

The canonical multistage dynamic stochastic program with linear constraints can be expressed as

$$
\begin{aligned}
&\min_{x_1} c_1 x_1 + \mathbb{E}_{\omega^2}\left\{ \min_{x_2(\omega^2)} f_2(x_2(\omega^2), \omega^2) + \mathbb{E}_{\omega^3|\omega^2}\left(\min_{x_3(\omega^3)} f_3(x_3(\omega^3), \omega^3) + \cdots \right.\right. \\
&\qquad \left.\left. + \mathbb{E}_{\omega^T|\omega^{T-1}}\left[\min_{x_T(\omega^T)} f_T(x_T(\omega^T), \omega^T) \right] \cdots \right) \right\} \\
&\text{s.t.} \quad A_{11}x_1 = b_1, \\
&\quad A_{21}(\omega^2)x_1 + A_{22}(\omega^2)x_2(\omega^2) = b_2(\omega^2), \\
&\quad A_{31}(\omega^3)x_1 + A_{32}(\omega^3)x_2(\omega^2) + A_{33}(\omega^3)x_3(\omega^3) = b_3(\omega^3), \\
&\quad \vdots \qquad\qquad \vdots \qquad\qquad \ddots \qquad \vdots \\
&\quad A_{T1}(\omega^T)x_1 + A_{T2}(\omega^T)x_2(\omega^2) + \cdots + A_{TT}(\omega^T)x_T(\omega^T) = b_3(\omega^T), \\
&\qquad x_1 \geq 0, \\
&\qquad x_t(\omega^t) \geq 0, \qquad t = 2, \ldots, T,
\end{aligned}
\tag{9.1}
$$

where ω_t is a random variable such that we have a stochastic process $\omega := (\omega_1, \ldots, \omega_T)$ on an abstract probability space $(\Omega, \mathcal{F}, P)$, and $\omega^t := (\omega_1, \ldots, \omega_t)$. Then, $\mathbb{E}_{\omega^t|\omega^{t-1}}(\cdot)$ denotes conditional expectation with respect to the current state of the data process ω conditioned on its past history, and all constraints hold almost surely, i.e., with probability one. The multistage problem may be tackled directly by considering its deterministic equivalent, or we may use versions of either Benders' decomposition or a scenario decomposition method, either of which can be extended to efficiently handle the multistage case. The multistage Benders' decomposition algorithm is known as nested Benders' decomposition [2, 14].

The modeler defines at each node in the event tree the coefficient process $\xi_t(\omega^t) := (A_{t1}(\omega^t), \ldots, A_{tt}(\omega^t), f_t(\cdot, \omega^t), b_t(\omega^t))$ either in a representation suitable for solution of the deterministic equivalent problem by a conventional deterministic solver or in some representation suitable for solution by a DSP-specific solver (for example, the SMPS file format [3]). This is often referred to in the modeling language literature as the algorithm's form. As in the case of traditional deterministic mathematical programming, it is more convenient for the modeler to consider the problem in modeler's form, using set-theoretic and algebraic notation to represent the real-world situation to be modeled. In this paper, after introducing an example problem, we examine how AMPL, a deterministic algebraic modeling language, can be used to specify stochastic programming problems and produce either deterministic equivalent or DSP-specific algorithm forms directly. However, realistic stochastic programs are often very large, both in terms of the size of the event tree and the constraint dimensions m_t and n_t, and typically they have a highly repetitive structure. In this case it becomes necessary to augment deterministic modeling languages with tools that can handle these structures, and in sections 9.4–9.6 we describe the `stochgen` toolchain, which provides such facilities for the modeling languages AMPL and XPRESS-MP. In section 9.7 we describe the STOCHASTICS `solgen` nested Benders solver and in section 9.8 demonstrate its performance on a variety of real-world large-scale DSP problems. In the final section we describe our current work on STOCHASTICS, in which we are developing a stochastic programming specific modeling language, as well as support tools to enable the visualization of problem and solution data for very large event trees.

9.3 An example problem

To provide a concrete example, we consider the modeling of a hypothetical portfolio management problem where the objective is to maximize the expected terminal value of a portfolio consisting of a stock and a bond. At each time period $t = 1, \ldots, T$ the decision vector should tell us the optimal portfolio composition on each scenario. We have constraints to specifiy the initial portfolio value and to disallow short selling (i.e., holding negative quantities of an asset). The first modeling task is to specify stochastic processes for the two asset classes; a standard financial model is to assume that the (two-dimensional) price process s is a correlated geometric Brownian motion; i.e., components of s follow the stochastic difference equation

$$ds_i = \mu_i s_i dt + \sigma_i s_i d\mathbf{Z}_i, \ i \in I, \tag{9.2}$$

where $\mathbf{Z}_i$ is a correlated Brownian motion and $I := \{\text{"stock"}, \text{"bond"}\}$. The drifts ($\mu$) and volatilities ($\sigma$) can be estimated from historical data. Of course the DSP framework assumes discrete time, so we model the movements of s with the stochastic difference equation

$$\frac{s_{i(t+1)} - s_{it}}{s_{it}} = \mu_i \Delta t + \sigma_i \sqrt{\Delta t} \phi_i, \tag{9.3}$$

where ϕ is a correlated standard normal random vector, and for $i \in I$, $s_{i1} := s_{i1}$, a constant. Given a covariance matrix Σ, a standard trick to generate values for ϕ is to generate uncorrelated standard normal deviates ϵ and find a matrix M such that $MM' = \Sigma$. Then $\phi = M\epsilon$, as from the definition of covariance $\Sigma = \mathbb{E}(\phi\phi') = M\mathbb{E}(\epsilon\epsilon')M' = MM'$.

Given the bivariate price process s as data process, the stochastic program which we wish to solve is

$$\begin{aligned}
&\max_{\substack{x_{it}(s^t):t=1,\ldots,T,\\ i\in I}} \mathbb{E}_{s^T}\left\{\sum_{i\in I} s_{iT}x_{iT}(s^T)\right\} \\
\text{s.t.}\ &\sum_{i\in I} s_{it}x_{i(t-1)}(s^{t-1}) = \sum_{i\in I} s_{it}x_{it}(s^t) \quad \text{a.s.}, \quad t = 2,\ldots,T, \\
&\sum_{i\in I} s_{i1}x_{i1} \leq 100, \\
&x_{it}(s^t) \geq 0 \quad \text{a.s.}, \quad i \in I, \quad t = 1,\ldots,T,
\end{aligned} \tag{9.4}$$

where $x_{it}(s^t)$ is the net asset value (NAV) held of asset i at time t given data history s^t. The constraints ensure that the total wealth is preserved between stages, that the initial cash requirement is less than or equal to \$100, and that the position is never shorted.

This particular model is too simple to be realistic. No account is taken of the investor's attitude to risk—the portfolio which maximizes expected terminal wealth is also likely to have a high terminal variance. Also a realistic problem would have (as well as a much larger asset set) constraints on the change in portfolio composition between different periods and would take account of taxation and transaction costs. Without these considerations, it is possible to see that this problem is solvable analytically. Indeed, the myopic strategy that at each node of the event tree invests the entire wealth in the asset with the highest expected

return in the next time step, i.e., that sets

$$x_{it}(s^t) = \begin{cases} \sum_{j \in I} s_{jt} x_{j(t-1)}(s^{t-1}) & \text{if} \quad i = \arg\max_{j \in I} \mathbb{E}_{s^{t+1}|s^t} \left(r_{j(t+1)}(s^{t+1}) \right), \\ 0 & \text{otherwise,} \end{cases} \tag{9.5}$$

where $r_{j(t+1)}(s^{t+1}) := (\frac{s_{j(t+1)} - s_{jt}}{s_{jt}})$, is an optimal strategy. Also realistic models use more sophisticated price processes than geometric Brownian motion. Nevertheless the model is adequate for illustrative purposes and in the course of this paper we will say how each of the mentioned extensions for realism can be handled.

9.4 Representing event trees

For concreteness, we shall model the example problem with three stages using the scenario set $\Omega := \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5\}$ over which we have a partition

$$\begin{aligned} \mathcal{A}_1 &= \{\Omega\}, \\ \mathcal{A}_2 &= \{\{\omega_1, \omega_2\}, \{\omega_3\}, \{\omega_4, \omega_5\}\}, \\ \mathcal{A}_3 &= \{\{\omega_1\}, \{\omega_2\}, \{\omega_3\}, \{\omega_4\}, \{\omega_5\}\}, \end{aligned} \tag{9.6}$$

and probability measure $P(\omega_i) = 0.2$. This gives us the scenario tree schema shown in Figure 9.1. Note that in reality a much larger event tree would be required to adequately discretize a data process such as the one given above.

There are two convenient event tree representations. The first, most commonly used representation uses a *tree string*. This is a string of integers which specify for each stage the number of branches for each node in that stage. We normally write tree strings as a product of powers, so, for example, the tree string $4.3.2.1^3$ generates a seven-stage event tree which has four branches in the first stage, three in each subtree of the second, etc. Obviously this allows the specification of only balanced trees (in the sense that each subtree in the same period has the same number of branches), but in the absence of more detail about the correct information structure this is normally adequate. For the specification of arbitrary event trees, we may use a *nodal partition* (NP) *matrix* [5]. This is a matrix with a row for each scenario and a column for each stage. We assign each node of the event tree a unique number, and the NP matrix entries n_{kt} are node numbers, so that each row of the matrix shows which nodes in the event tree a scenario passes through. We say that an NP matrix is in standard form if $n_{ij} \leq n_{i'j}$ whenever $i \leq i'$ and $n_{ij} \leq n_{i'j'}$ whenever $j < j'$ (for any i'). Table 9.1 gives an appropriate NP matrix for our example problem.

Note that the NP matrix is a redundant way of storing the tree, and it would be more efficient (for example) just to store the predecessor node of each node in the event tree. The NP-matrix representation, however, has been maintained, primarily for historical reasons, and the overhead of storing it is not significant when compared to the storage requirement of the full stochastic program. An alternative, similar representation for event trees is possible by considering the *scenario* partition; this is often referred to as the Lane matrix (after [19]). An assumption in both of these representations is that only trees with a uniform depth are considered. While this does not imply a loss of generality (as we can appropriately constrain decisions to be zero), it would not be the most efficient representation for event trees where this is not the case.

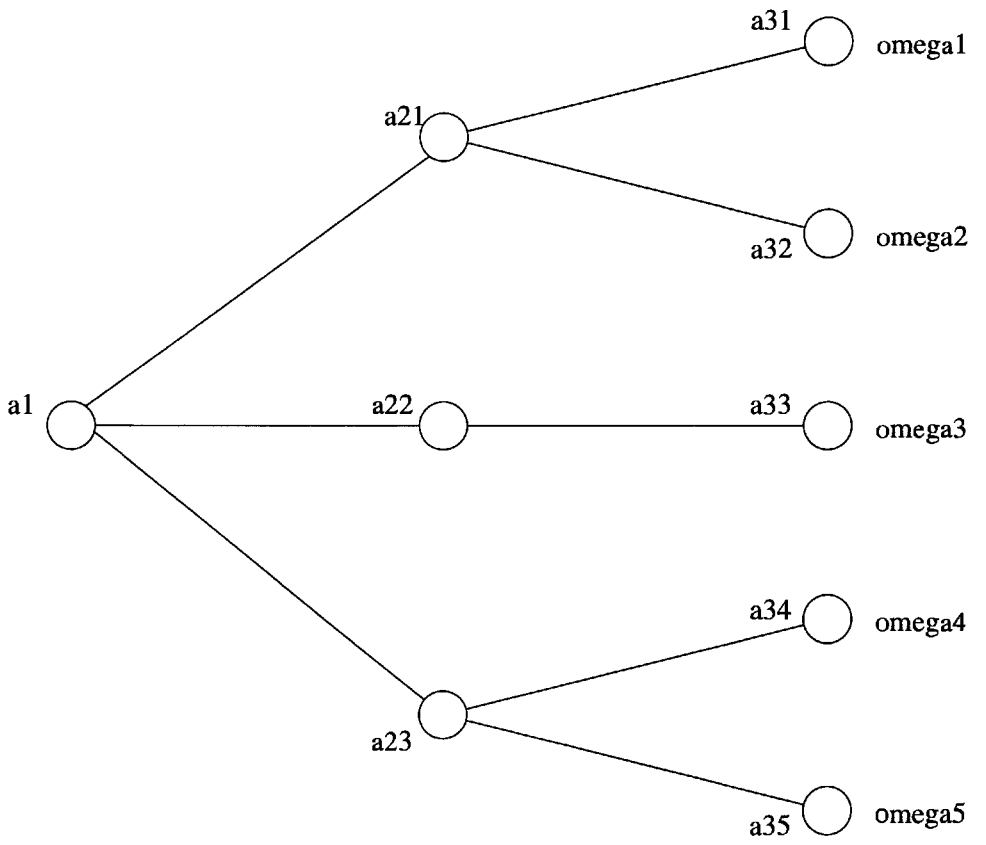


Figure 9.1. *Event tree and partitions.*

Table 9.1. *Nodal partition matrix representation of event tree.*

	$t=1$	$t=2$	$t=3$
$k=1$	1	2	5
$k=2$	1	2	6
$k=3$	1	3	7
$k=4$	1	4	8
$k=5$	1	4	9

9.5 The nodal formulation

Having defined the event tree, we are in a position to instantiate simulator data over it. For the example problem, the price process we wish to simulate is sufficiently simple to be

defined using the facilities available within AMPL. First we define the set of event tree nodes, a parameter to contain their probabilities, and a parameter to contain their predecessors:

```
set nodes ordered;
param pred{nodes};
param prob{nodes};
```

We define the set of nodes as an "ordered" set (partially ordered by time) so that we can easily refer to the root and leaf nodes. To define the event tree of the example problem, we instantiate the above set and parameters with the following data block:

```
data;
param:  nodes:      pred prob :=
        1             .   1.0
        2             1   0.4
        3             1   0.2
        4             1   0.4
        5             2   0.5
        6             2   0.5
        7             3   1.0
        8             4   0.5
        9             4   0.5;
```

The probabilities specified are the conditional probabilities of each node occurring, given that its predecessor has occurred. In the case of a realistically large event tree, these data would normally be generated automatically, either from a nodal partition matrix or a tree string, and read in from a file. The `stochgen` tool chain provides facilities for generating and manipulating tree structures and generating AMPL-compatible data files.

We also define the auxiliary objects `stage`, `stages`, `T`, and `uprob` as

```
param stage{n in nodes} := if n = 1 then 1 else stage[pred[n]] + 1;
param T := stage[last(nodes)];
set stages := 1 .. T;
param uprob{n in nodes} := if n = 1 then 1.0 else uprob[pred[n]]*prob[n];
```

The parameter `stage` maps nodes to time stages, `T` is defined as the last decision stage, and `stages` is the set of time stages. The parameter `uprob` is the unconditional probability of each node occurring.

To generate numeric data for asset prices, we define and assign data to the parameters μ, σ, and Δt from (9.3) and specify the correlation `c` between the two asset prices. We also require initial values for both assets. This is achieved by the following AMPL code:

```
set assets;

param mu    {assets};
param sigma{assets};
param c;
param dt;

param price1 {assets}; # price of assets at t=1
```

```
data;

param dt := 1.0;
param:  assets:     mu     sigma   price1 :=
        stock       0.15    0.2     50.0
        bond        0.10    0.1     50.0;

param c       := -0.3;

param e {n in nodes, i in assets} := Normal01();
```

As in the case of the event tree, if we had a large number of assets, we would store the data specification in a separate file. To generate correlated random deviates ϕ we first use the built-in AMPL function `Normal01` to assign standard (uncorrelated) normal random deviates to a parameter `e`, which is defined over each node in the event tree for each asset:

```
param e {n in nodes, i in assets} := Normal01();
```

This represents the statement

$$e_{ti} \sim N(0,1), \quad t = 1, \ldots, T, \quad i \in I. \tag{9.7}$$

Given the covariance matrix

$$\Sigma = \begin{pmatrix} 1 & c \\ c & 1 \end{pmatrix}, \tag{9.8}$$

it is easy to verify that

$$M = \begin{pmatrix} \sqrt{1-c^2} & c \\ 0 & 1 \end{pmatrix} \tag{9.9}$$

is a suitable factorization. The AMPL code to perform the matrix multiplication $\phi = M\epsilon$ is

```
param phi {n in nodes, i in assets} :=
        if i = "stock" then
          sqrt(1.0 - c*c)    * e[n, "stock"]
        + c                  * e[n, "bond"]
        else
          e[n, "bond"];
```

We have assigned correlated random deviates to each node of the event tree. To specify the price process all that remains is to implement the difference equation (9.3) over the event tree as follows:

```
param price {n in nodes, i in assets} :=
       if n = first(nodes) then
         price1[i]
       else
         price[pred[n], i]) * (1 + mu[i]*dt + sigma[i]*sqrt(dt)*phi[n,i]);
```

This definition is recursive (as are the definitions for `stage` and `uprob`); AMPL will automatically carry out the computation in such a way that `pred[n]` is always evaluated before `n`, provided we ensure that `pred[n]` $\neq$ `n`.

In principle it is possible to express arbitrarily complicated stochastic data processes using an algebraic modeling language such as AMPL in this way, but in practice it may not be all that convenient. For example, if we have a large number of assets, we can obtain a value of M by using Cholesky factorization of the covariance matrix Σ; however, AMPL does not provide a routine for this, and while it could be implemented using AMPL's imperative programming facilities, it would not be particularly readable or efficient. Furthermore, as we indicated earlier, difference equations for a realistic price process model (or, for that matter, any realistic stochastic process model) are normally substantially more complicated than (9.3) (see [25, 26] for an example) and in this case, obtaining coefficients from historical data involves sophisticated econometric modeling, for which dedicated software (such as RATS or S-PLUS) provides a more appropriate environment. In the worst case (which is not particularly unusual) the data process simulator may be available only as a "black box," and we have no possibility of integrating it into an AMPL model. To provide an interface between an arbitrary simulator and a modeling language, the `stochgen` tool chain provides a program called `procgen`. We assume that the data process simulator can be encapsulated by the function

$$(s_t, s_{t+1}, \ldots, s_T) = f(k, s_{t-1}, s_{t-2}, \ldots, s_{t-l}). \tag{9.10}$$

That is, future states of a data process s are a function of l previous states and a seed k for a random number generator. By allowing $l \neq 1$ we allow the possibility of non-Markovian data processes. It is possible to have $l > T$, but l is fixed for all scenarios (the path simulator may choose to ignore initial conditions in some scenarios if they are unnecessary). We have f produce $s_t, s_{t+1}, \ldots, s_T$ instead of just s_{t+1} to avoid incurring any setup costs on the simulator more often than is necessary. `procgen` takes as input the function f (it is either called as an external program or is dynamically linked) and an event tree, and runs it once for each path in the event tree, in such a way that no state $s_t(\omega^t)$ is requested more than once, and the seed k is updated in such a way that each generated path will be unique. The result is a set of nonredundant partial data paths in which the simulated data process realizations at nodes of the event tree with a common predecessor node have been generated conditional on the unique data path history to that node. Then numerical data for the entire event tree are output in a data format suitable for a variety of modeling languages or visualization tools. We must also allow for the possibility that multiple simulator time steps are taken per time period, and that different stages in the same problem may have different numbers of periods. It is quite typical, for example, that the simulator produces monthly data but decisions are required on a quarterly or annual basis and later stages contain several quarters or years.

Given this abstraction, the modeler need only define f in a suitable way; we have found that people generally find this much simpler and less error-prone than implementing a "tree-aware" simulator on a case-by-case basis. There is an assumption here that the simulator is such that we do not need to know s on another scenario, e.g., $s_t(\omega^{t'})$, to generate $s_t(\omega^t)$; one can imagine models where this is not the case (for example, in the generation of arbitrage free prices). In this situation, iteration over the tree is necessarily model-specific, but for the models we have encountered so far, the `procgen` abstraction has been sufficient.

We are now in a position to declare decision variables x over the set of event tree nodes and for all assets as

```
var x{nodes, assets} >= 0;
```

This incorporates the no-shorting constraint; it is common modeling-language practice to put simple bounds such as this in the variable definition. We can define the objective function over the leaf nodes of the event tree, using the unconditional probability parameter `uprob`,

```
maximize expected_terminalwealth:
        sum{i in assets, n in nodes : stage[n] = T} uprob[n]*price[n,i]*x[n,i];
```

and we define constraints similarly:

```
subject to self_financing{n in nodes : stage[n] <> 1}:
        sum{i in assets} price[n,i]*x[n,i]
                = sum{i in assets} price[n,i]*x[pred[n],i];

subject to budget:
        sum{i in assets} price[first(nodes),i]*x[first(nodes),i] <= 100;
```

This completes the nodal formulation of the example problem. In effect, we have used the modeling language to define the standard form deterministic equivalent linear programming problem corresponding to the stochastic program we wished to model. To solve this, AMPL will construct this problem in full and send it to an LP solver that the user has provided. This has negative implications for efficiency. For most DSPs of interest, the deterministic equivalent form contains a great deal of redundancy, since the number of coefficients that are stochastic is small compared to the total number of coefficients at each node, and data for each node in the event tree tend to have a structure similar to that for other nodes, especially other nodes in the same stage. By constructing the deterministic equivalent, we also disregard structure information that may be exploited by a DSP solution algorithm such as nested Benders decomposition or the primal-dual interior-point method.

9.6 `stochgen` formulation

To avoid creating the deterministic equivalent problem, and so that the modeler can avoid dealing with the recursive definitions necessary for a nodal formulation, `stochgen` requires that a problem be defined which is representative of *one scenario* of the dynamic stochastic program, that is, the problem that would be obtained if random variables were made deterministic. Instead of indexing variables and constraints over the set of event tree nodes as above, the modeler indexes over time stages. For the example problem, this leads to the following AMPL code:

```
var x{stages, assets} >= 0;

maximize expected_terminalwealth:
        sum{i in assets} price[T,i]*x[T,i];

subject to self_financing{t in 2 .. T}:
        sum{i in assets} price[t,i]*x[t,i]
                = sum{i in assets} price[t,i]*x[t-1,i];

subject to budget:
        sum{i in assets} price[1,i]*x[1,i] <= 100;
```

We refer to this problem as the core problem (it is the same as the core problem of the SMPS standard). The inputs to `stochgen` are the core problem, a description of the event tree, and a `procgen`-compatible data path simulator as described above. An advantage of this approach is that a user may start with an existing deterministic model and convert it to a stochastic program by simply adding information which describes the stochastic data process and model coefficients or functions.

Running AMPL on the above problem would produce a single scenario. We use the imperative programming features of AMPL to repeat this process for each scenario in the event tree, taking care that the data process parameters (in the example problem just the `price` parameter) point to the appropriate nodes in the event tree. How precisely this is done depends on the format of the stochastic data; `procgen`-generated data can be processed with standard headers which we supply to define objects `process` and `path`, so that the definition of the price parameter becomes

```
param price{t in stages, i in assets} := process[ord(i,assets),path[t]];
```

Normally AMPL sends its output directly to a solver, but in this case, no solution occurs until the last scenario has been processed. Instead, `stochgen` takes the output of AMPL and generates an SMPS representation of the DSP. To do this the model must be annotated with the dynamic structure of the problem. For this, we use the `suffix` notation of AMPL to assign stage information to each variable and constraint in the model. For the example problem, appropriate code would be

```
let {t in stages, i in assets} x[t,i].order := t;
let {t in 2..T} self_financing[t].order := t;
let budget.order := 1;
```

In terms of the linear program corresponding to the core problem, the effect of this is to impose a block lower-triangular form on the constraint matrix by permuting rows and columns so that they are ordered by stage. Figure 9.2 illustrates the effect of this annotation on the core problem's output by AMPL. The top picture shows the constraint matrix for a four-stage instance of the example problem with an arbitrary ordering of rows and columns (as is generated if the suffix information is not supplied). The bottom picture shows the problem with stage ordering imposed, so that it is in a block lower-triangular form. This specification of the problem's dynamic structure is essential in any DSP formulation technique, not only for generating an SMPS representation but also so that we can apply scenario decomposition or nested Benders solution methods.

If at this stage we wish to generate the deterministic equivalent form, the `stochgen` toolchain provides `detgen`, which generates the deterministic equivalent in MPS form from the SMPS representation.

As well as simplifying the AMPL representation of DSP models, the `stochgen` formulation allows for more efficient generation when the event tree becomes very large, because the deterministic equivalent form is never stored in memory in its entirety, and `stochgen` is aware of the redundancies present in the DSP formulation. In section 9.8 we will give details of problems which would have been impossible to generate using an AMPL nodal formulation in any reasonable amount of memory. Also, for example, in the context of importance sampling [8, 11, 16, 15] when it is necessary to regenerate the

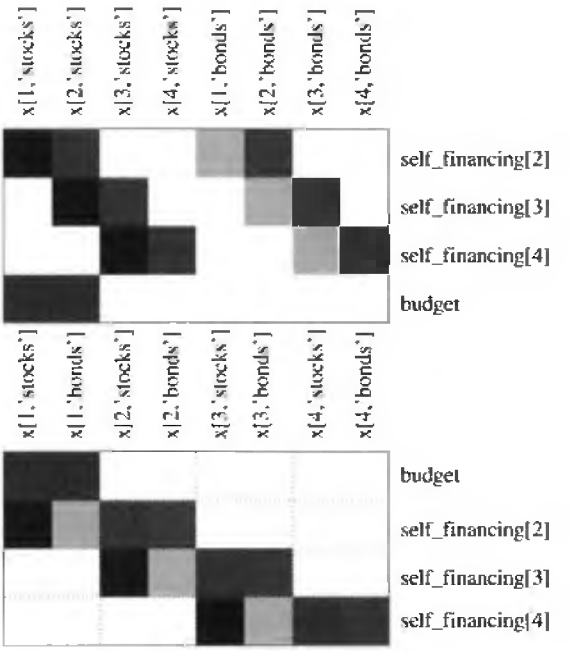


Figure 9.2. *Constraint matrices of unordered and ordered core problems.*

problem corresponding to only some part of the event tree, the `stochgen` framework has been used to do this efficiently, whereas by using a nodal formulation we would have been forced to regenerate the entire deterministic equivalent formulation. As far as we are aware, the `stochgen` component of STOCHASTICS is the only stochastic programming modeling system which has been used in this situation; other authors on the subject have had to hard code their models.

9.7 Nested Benders' decomposition using `solgen`

The superiority of nested Benders' decomposition over deterministic equivalent solution methods has been demonstrated repeatedly for a wide variety of problems. So far there has been only one generally available implementation, called MSLiP, originally described in [14], which was extended in [10] to integrate it with commercial LP solvers and provide a parallel capability. Other two-stage Benders' decomposition solvers are available (such as DECIS [17] and OSL [18]) which can be adapted to solve multistage problems by using aggregation, but in our experience problems with a large number of stages generally benefit from a multistage implementation [9]. Another possibility is to implement decomposition directly using the modeling language; the chapter by Gassmann and Gay in this volume (Chapter 10) gives details on how this might be achieved.

Part of the STOCHASTICS system is `solgen`, a new implementation of nested Benders' decomposition which has been designed to be tightly integrated with modern modeling

languages and LP solvers. The current version (1.30) of `solgen` uses CPLEX 7.1 to solve subproblems and can read in problems in MPS or SMPS format, or those generated using AMPL, XPRESS-MP, or the `stochgen` tools. Our emphasis has been to develop a solver which is both faster than currently available alternative methods and robust enough to be used in a general-purpose setting. In addition, the following features have been developed as part of our ongoing research:

- *Aggregation.* As was shown in [10], stage aggregation of the scenario tree can lead to accelerated solution times when using Benders decomposition. Our further research has shown that, if the correct aggregation strategy is known, it is possible to get orders of magnitude speed-up over the solution of the unaggregated problem or over the deterministic equivalent problem. If the modeler is frequently solving similar problems, it is worth the effort to find these aggregations. An area of our current research is to find a heuristic model for choosing a good aggregation strategy ab initio and then use feedback from the solver as it runs to tune the strategy.

- *Regularization.* It has been known for some time that decomposition methods (both Benders decomposition and Dantzig–Wolfe decomposition) can behave poorly on some problems—the sequence of proposals generated by the standard algorithm can lead to a piecewise linearization of the recourse function which is as hard to handle as the recourse function itself. To avoid this situation we have investigated the use of regularizing terms in the objective function (following the work in [22, 23, 24]). The idea is to maintain an incumbent solution, choose proposals which are near it, and change the incumbent only when there is a demonstrable improvement in the linearization. We have used this method to successfully solve several problems which were previously either only amenable to deterministic equivalent methods or insoluble and to accelerate the solution of other problems. Currently the regularized method is applicable only to two-stage problems, but a multistage implementation is in progress.

- *General concave objectives.* In financial applications it is important to be able to model general concave utility functions in order to handle investors' attitudes to risk, and `solgen` takes two approaches to tackling such problems. The first method is to perform a further decomposition of the problem so that the concave objective is contained in an artificial "final" stage which can then easily be solved (because it is an unconstrained maximization problem and therefore easily approximated linearly). The second method is to use concave subproblem solvers to handle the concave objective directly. Currently the latter method uses CPLEX barrier as a subproblem solver; we hope to use a pivoting QP method soon and then employ SQP techniques to handle the general case.

- *Non-Markovian nested decomposition.* When the matrices $A_{ts}(\omega^t)$, $s < t-1$, in (9.1) are nonzero, the problem is said to have a non-Markovian constraint structure. Such structures arise in multistage scheduling problems and in financial problems which have complicated taxation, liability, or liquidity structures. A naïve approach is to introduce splitting variables to induce a Markovian structure; however, this can result in a quadratic increase in the size of the problem. Instead, we have extended the nested

Benders algorithm to handle non-Markovian problems directly and have modified `solgen` appropriately. A number of problems have been solved using this new technique.

We have also been working on adaptations of the algorithm to solve certain nonconvex problems which arise in portfolio management. These include the fixed-mix portfolio problem (which has a bilinear constraint on asset holdings) and problems which have guaranteed return or value-at-risk requirements (both of which can be modeled as probabilistic constraints).

9.8 A problem test set and computation times

In this section, we look at five problems which are drawn both from our own work and from the stochastic programming literature.

- STORM. A two-stage stochastic freight scheduling problem (described in [21]) which is part of the POST standard problem set (http://users.iems.nwu.edu/~jrbirge/html/dholmes/post.html). Several other authors supply computation times for this problem, so we do so here for comparison.

- WATSON. An asset-liability management problem formulated by Dempster et al. for Watson Wyatt Worldwide, based on the CALM model [7, 4] which is now made publically available (http://www-cfr.jims.cam.ac.uk/research/stprog.html).

- CORO. A two-stage formulation of the HChLOUSO hydrocarbon logistics planning problem [12]. The problem here is from the Case 3 data set which involves planning the movement and spot market transactions of seven oil products between 41 ports, sales locations, and storage depots under uncertain local demands and prices.

- DROP. An alternative (more tightly constrained) formulation of the HChLOUSO problem [9], also involving refining and detailed in this volume.

- PFEX. Here we have taken the example problem described above for two assets and extended it so that it models nine asset classes (stocks and bonds) in three currency regions. Interest rates for the bond processes use a mean-reverting model. To make the problem realistic, a piecewise linear objective is used to give an attitude toward risk, and a liquidity constraint of the form

$$|s_{it}(\omega^t)x_{it}(\omega^t) - s_{it}(\omega^t)x_{i(t-1)}(\omega^{t-1})| \leq \nu \sum_{j \in I} s_{jt}(\omega^t)x_{jt}(\omega^t), \quad i \in I, \quad t = 2, \ldots, T,$$

is imposed, which stops changes in position between stages from being more than a fraction ν of total wealth.

All problems (apart from STORM) were formulated using the `stochgen` modeling system in conjunction with either AMPL or XPRESS-MP. Table 9.2 gives the numbers of scenarios, stages, and (standard form) deterministic equivalent problem dimensions for each problem.

Table 9.2. *Test set problem dimensions.*

Problem	Stages	Scenarios	Rows	Columns	Nonzeros	Objective
STORMG2.8	2	8	4394	10193	27424	15535235.73
STORMG2.27	2	27	14388	34114	90903	15508982.32
STORMG2.125	2	125	65936	157496	418321	15512091.18
STORMG2.1024	2	1024	526186	1259121	3341696	15802590.35
WATSON.10.256.C	10	256	43518	82177	218888	1849.40
WATSON.10.512.C	10	512	67070	128001	350728	1797.30
WATSON.10.1024.C	10	1024	134128	255987	701428	1798.42
WATSON.10.1920.C	10	1920	251442	479905	1315028	1778.36
WATSON.10.2688.C	10	2688	352014	671861	1841028	1687.72
CORO.2.10	2	10	155246	545633	1456120	24455.41
CORO.2.50	2	50	770446	2688633	7245640	24437.31
DROP.2.10	2	10	155271	500820	2182070	1179057.97
DROP.2.50	2	50	766471	2464820	10769670	1179083.72
PFEX.6.3840	6	3840	207043	109290	763379	14905.98
PFEX.6.7680	6	7680	412483	217770	1520819	7582.11
PFEX.6.30720	6	30720	1126723	609450	4147379	3019.06

We give these statistics with the usual proviso that a problem's size gives only a very rough indication of how difficult it is to solve. In particular, the frequent emphasis on producing problems with a very large number of scenarios is not always justified. We have observed (for example, for portfolio management problems) that the solution stabilizes with quite modest numbers of scenarios, providing the constraint structure acts to prevent myopic strategies such as the one discussed in section 9.3 from being optimal. Nevertheless these problems can be harder to solve than much larger problems which enjoy more separability. Although this point is obvious, there has been a tendency in the computational literature on stochastic programming to solve problems with huge numbers of scenarios with little evidence that the formulation is not unrealistically underconstrained.

Table 9.3 gives solution statistics for the test set solved using `solgen` and (where available) for the same problems solved using the fastest of the CPLEX primal and dual simplex and barrier methods (indicated by P, D, or B, respectively, in the final column). All experiments were run on an AMD Athlon 650 MHz with 512 MB RAM. Note that the solution times for all but the two smallest STORM problems are shorter for `solgen` than for any of the CPLEX algorithms, sometimes by an order of magnitude. Also, because `solgen` is aware of the redundancy in the DSP formulation, memory requirements are reduced, and with the same machine much larger problems can be solved. For the DROP problems it was necessary to use a regularized master objective; currently this requires the master problems to be solved using an interior-point method. Because interior-point methods cannot easily be hot-started, solution times are very long, but we are working on simplex-based methods which should be much faster.

Table 9.3. *Test set computational results—*`solgen` *and CPLEX.*

	`solgen`		CPLEX		
Problem	time (s)	memory (MB)	time (s)	memory (MB)	method
STORMG2.8	2.41	39	1.30	6	D
STORMG2.27	8.15	46	7.04	16	D
STORMG2.125	45.04	63	89.25	70	D
STORMG2.1024	350.19	238	1363.58	688	B
WATSON.10.256.C	11.16	16	25.01	35	B
WATSON.10.512.C	12.92	21	46.60	55	B
WATSON.10.1024.C	32.80	39	116.61	110	B
WATSON.10.1920.C	55.46	78	207.95	205	B
WATSON.10.2688.C	92.93	100	323.74	287	B
CORO.2.10	269.66	19	1981.35	160	P
CORO.2.50	1003.11	38	22584.52	801	P
DROP.2.10	1688.49	36	2231.00	175	P
DROP.2.50	5623.66	60	52250.13	861	P
PFEX.6.3840	201.47	72	924.25	142	B
PFEX.6.7680	461.92	146	-	-	-
PFEX.6.30720	2228.40	435	-	-	-

9.9 Future developments

`stochgen` 3

In this concluding section we document current and planned development of the STOCHASTICS system. The version of `stochgen` currently under development (shown in Figure 9.3) aims to provide the modeling, solution, and visualization features required by industrial DSP applications with a set of components and stand-alone tools controlled through a separate graphical user interface, as shown in Figure 9.3. Here we are using Excel for this, allowing simulator, model, and solver parameters to be managed easily. The model (shown being edited on the left) is an AMPL nodal formulation of a portfolio management problem (a version of the CALM model described in [4, 5]), while the description of the event tree and simulator parameters are maintained separately in an Excel spreadsheet. Visualization of scenario data and the solution are provided by separate Java components. Controlling the whole process from Excel allows automatic solve-resolve scripts (such as are needed to generate backtest results from historical data) to be written in VBA. All the `stochgen` 3 components can be accessed from other languages (Java, C, C++), if this is preferred to VBA, or even can be used as stand-alone tools.

Visualization

Once a stochastic program has been solved, there remains the problem of viewing and analyzing the solution. This forms a critical part of the model-solve-analyze cycle, which

Figure 9.3. *Using* `stochgen` 3 *to solve a portfolio management problem.*

must be repeated several times before a satisfactory solution to a DSP problem is found. Fast and flexible visualization tools are thus essential if DSP is to be widely applied commercially.

DSP solution data consist of the value of the decision variables at nodes on a event tree, together with constraint dual information. While the variables along each scenario in the tree can be extracted and viewed using a variety of visualization tools (Excel, MATLAB, etc., or the component shown in Figure 9.3), these "scenario viewers" have the restriction that the user can easily see data only along a few scenarios at once. Although this is often acceptable for visualizing paths of stochastic processes (see Figure 9.4, for example) the approach is less well suited to examining solutions of stochastic programs where one may have hundreds of thousands of scenarios which are subtly related to each other through the branching of the event tree.

For example, given a collection of different scenarios with a similar sequence of decisions, we cannot say if the decisions are the same because the scenarios have not reached the point in the event tree where they branch from one another, or whether the solution happens to be rather nonstochastic in this region of the tree. With a scenario-based view, it becomes impossible to view data for all scenarios at once without seeing a dense "cloud" of scenarios (as exemplified by Figure 9.4) in which this branching is obscured. The example in Figure 9.4 has only 500 scenarios, but it is already hard to distinguish one scenario from another. With 100,000 scenarios this problem would be insurmountable.

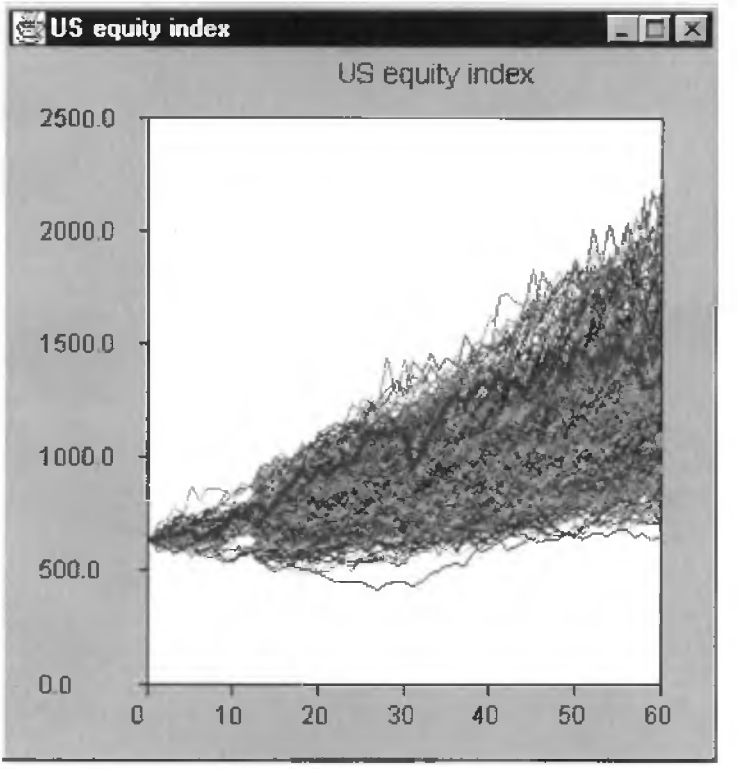


Figure 9.4. *Scenario-based visualization.*

As part of `stochgen` 3, a "tree viewer" component has been written which can be used to visualize and explore data defined on trees. This allocates a single rectangle for each node in the tree (Figure 9.5 shows the layout of rectangles for a tree with 2-2-2-2-2 branching) by placing nodes in the same time period in a vertical column, with each node to the right of its predecessor node. The vertical ordering of nodes can also be chosen based on some function of the nodal problem or solution data.

The different rectangles can then be used to display solution information. The benefit of the tree-based view is that it allows us to see properties such as stability of solution over the tree or to find regions of the tree in which the solution behaves unexpectedly. In Figure 9.6 we show the solution to the portfolio management problem of Figure 9.3 by dividing each node's rectangle into vertical bars with width proportional to the proportion of the portfolio invested in each asset. Thin black bands represent the divisions between time stages. Note that nodes in the final stage do not have decision variables in this model and hence the last column is blank. Figure 9.7 shows a zoom-in of part of the solution.

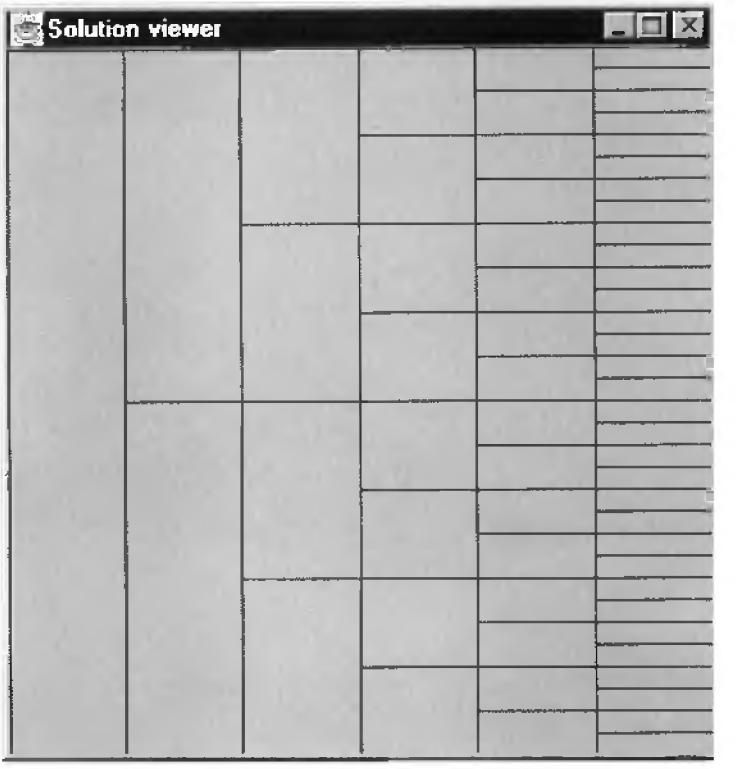


Figure 9.5. *Tree visualization, allocating a rectangle for each node, with nodes in the same time period stacked vertically.*

Problem generation

The main stumbling block to the solution of very large DSPs is to effect the generation of individual submatrices invoking the stochastic simulator only when required by the solver. Such an approach is also essential for resampling techniques [8, 11], where we may deliberately remove part of the event tree if the resampling algorithm decides that its effect on the solution is minimal and instead generate a more "bushy" tree in regions where the solution is sensitive to the discretization.

Stochastic programming modeling languages

The Stochplam [1] and AMPL extensions [13] to existing deterministic modeling languages have been proposed which implement different ways to express the extra information necessary to define a stochastic programming problem. Their emphasis has been on using a fixed event tree, for which the only extra information required by the problem generator is the knowledge of at what future time stochastic parameters values become known, variables

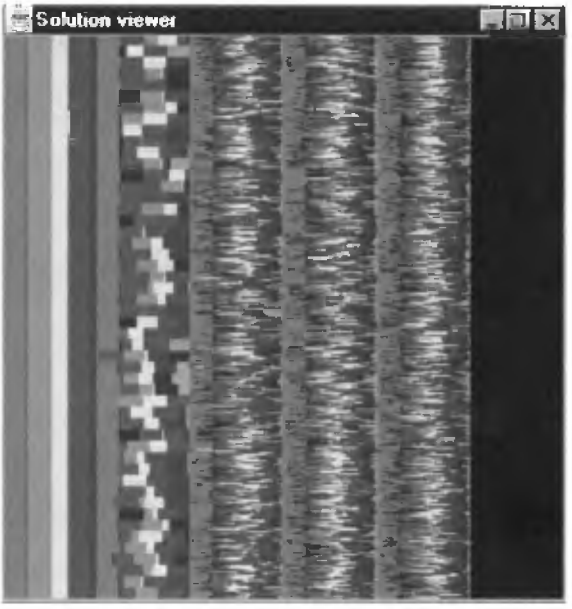


Figure 9.6. *Tree-based visualization the solution to a portfolio management problem. Here the branching in the tree is* 50-10-1-1-1.

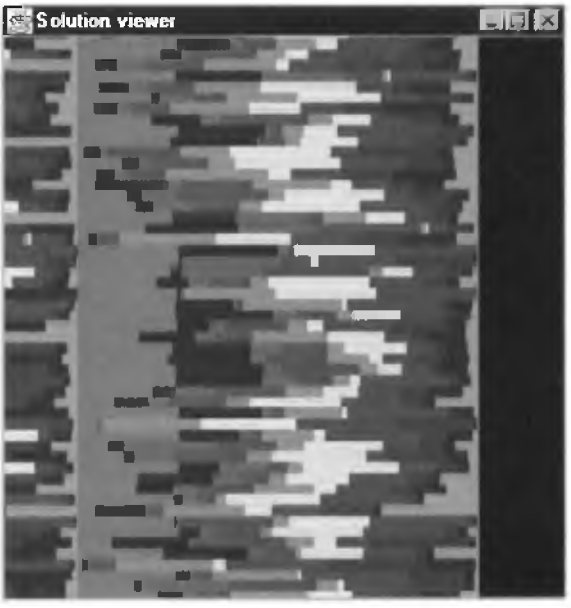


Figure 9.7. *Zoom-in on Figure* 9.6.

are chosen, and constraints are imposed. In AMPL this can be done either by using the suffixes facility to attach extra information to model entities (as described in section 9.6) or by imposing additional structure on the syntax of the model. We are examining both of these approaches to investigate whether these attempts to extend existing deterministic modeling languages are adequate or whether a purpose-built stochastic programming modeling language is needed as part of the STOCHASTICS system under development.

Note added in proof. This has now been effected in Release 4 of the STOCHASTICS system for which there are patents pending (http://www-cfr.jims.cam.ac.uk/stochastics).

Bibliography

[1] F. ALTENSTEDT, *Stochplam*, 2001; available online from http://www.cs.chalmers.se/~alten/stoplam/stochplam.html.

[2] J. R. BIRGE, *Decomposition and partitioning methods for multi-stage stochastic linear programs*, Oper. Res.. 33 (1985), pp. 989–1007.

[3] J. R. BIRGE, M. A. H. DEMPSTER, H. I. GASSMANN, E. A. GUNN, A. J. KING, AND S. WALLACE, *A standard input format for multiperiod stochastic linear programs*, Math. Program. Soc. Algorithms Newsletter, 17 (1987), pp. 1–20.

[4] G. CONSIGLI AND M. A. H. DEMPSTER, *The CALM stochastic programming model for dynamic asset-liability management*, in World Wide Asset and Liability Modelling, W. T. Ziemba and J. M. Mulvey, eds., Cambridge University Press, Cambridge, UK, 1998, pp. 464–500.

[5] G. CONSIGLI AND M. A. H. DEMPSTER, *Dynamic stochastic programming for asset-liability management*, Ann. Oper. Res., 81 (1998), pp. 131–162.

[6] M. A. H. DEMPSTER, ED., *Stochastic Programming*, Academic Press, London, 1980.

[7] M. A. H. DEMPSTER, *CALM: A Stochastic MIP Model*, Technical Report, Department of Mathematics, University of Essex, Colchester, UK, 1993.

[8] M. A. H. DEMPSTER, *Sequential Importance Sampling Algorithms for Dynamic Stochastic Programming*, Technical Report WP 32/98, Judge Institute of Management, University of Cambridge, Cambridge, UK, 1998.

[9] M. A. H. DEMPSTER, N. HICKS-PEDRÓN, E. A. MEDOVA, J. E. SCOTT, AND A. SEMBOS, *Planning logistics operations in the oil industry*, J. Oper. Res. Soc., 51 (2000), pp. 1271–1288.

[10] M. A. H. DEMPSTER AND R. T. THOMPSON, *Parallelization and aggregation of nested Benders decomposition*, Ann. Oper. Res., 81 (1998), pp. 163–187.

[11] M. A. H. DEMPSTER AND R. T. THOMPSON, *EVPI-based importance sampling solution procedures for multistage stochastic linear programmes on parallel MIMD architectures*, Ann. Oper. Res., 90 (1999), pp. 161–184.

[12] L. F. Escudero, F. J. Quintana, and J. Salmerón, *CORO, a modelling and algorithmic framework for oil supply, transformation and distribution optimization under uncertainty*, Eur. J. Oper. Res., 114 (1999), pp. 638–656.

[13] R. Fourer, *Proposed New AMPL Features: Stochastic Programming Extensions*, 1996; available online from http://www.ampl.com/cm/cs/what/ampl/NEW/FUTURE/stoch.html#scens.

[14] H. I. Gassmann, *MSLiP: A computer code for the multistage stochastic linear programming problem*, Math. Program., 47 (1990), pp. 407–423.

[15] J. L. Higle and S. Sen, *Stochastic Decomposition*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[16] G. Infanger, *Planning Under Uncertainty (Solving Large-Scale Stochastic Linear Programs)*, The Scientific Press Series, Boyd and Fraser, Boston, 1994.

[17] G. Infanger, *DECIS User's Guide*, 1997; contact the author in Belmont, CA.

[18] A. J. King, *SP/OSL Version* 1.0, *Stochastic Programming Interface User's Guide*, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1994.

[19] M. Lane and P. Hutchinson, *A model for managing a certificate of deposit portfolio under uncertainty*, in Stochastic Programming, M. A. H. Dempster, ed., Academic Press, London, 1980, pp. 473–493.

[20] K. Marti and P. Kall, eds., *Stochastic Programming: Numerical Techniques and Engineering Applications*, Lecture Notes in Control and Inform. Sci., Springer-Verlag, Berlin, 1995.

[21] J. Mulvey and A. Ruszczyński, *A new scenario decomposition method for large-scale stochastic optimization*, Oper. Res., 43 (1995), pp. 477–490.

[22] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Oper. Res., 1 (1976), pp. 97–116.

[23] A. Ruszczyński, *A regularized decomposition method for minimizing a sum of polyhedral functions*, Math. Program., 35 (1986), pp. 309–333.

[24] A. Ruszczyński and A. Świętanowski, *On the regularized decomposition method for stochastic programming problems*, in Stochastic Programming: Numerical Techniques and Engineering Applications, K. Marti and P. Kall, eds., Lecture Notes in Control and Inform. Sci., Springer-Verlag, Berlin, 1995, pp. 93–108.

[25] A. D. Wilkie, *Stochastic investment models—theory and applications*, Insurance Math. Econ., 6 (1987), pp. 65–83.

[26] A. D. Wilkie, *More on a stochastic asset model for actuarial use*, British Actuarial J., 1 (1995), pp. 777–964.

This page intentionally left blank

Chapter 10

An Integrated Modeling Environment for Stochastic Programming

Horand I. Gassmann and David M. Gay†*

10.1 Introduction

Over the past few years, stochastic programming has matured as a discipline. As this volume attests, there are numerous applications in finance, engineering, energy planning, and many other areas. General acceptance of the kind enjoyed by linear programming has been elusive, however. Stochastic programs tend to be large and require considerable user sophistication. Data requirements also tend to be high.

Due to the size and complexity of stochastic programs, teams of users with diverse skills must be assembled to develop, manage, and apply stochastic models. Such teams would typically include numerical analysts, algorithm developers, modelers, and end users. To make the system useful for everybody, seamless integration of all functions and presentation of the output must be achieved in a format familiar to each user.

For the end user this means report generators and interfaces to familiar database and spreadsheet programs, such as MS Access and MS Excel. Modelers may interact with the system through algebraic modeling languages such as AMPL [7], GAMS [4], MPL [13], or AIMMS [3]. Developers and numerical analysts are familiar with solvers such as CPLEX [12] or OSL [11] and the MPS [10] and SMPS data formats (see Chapter 2 in this volume).

This chapter describes an attempt at integrating the various functions into a comprehensive system. It is based largely on the algebraic modeling language AMPL and makes extensive use of AMPL's advanced features, such as ODBC database connectivity, the AMPL control language, and language constructs for stochastic programs developed in [8]. The financial model described here is not used as a production system. Rather, it is to

*School of Business Administration, Dalhousie University, Halifax, NS, B3H 3J5, Canada (horand.gassmann @dal.ca). This author was supported in part by a grant from the National Sciences and Engineering Research Council of Canada (NSERC).

†AMPL Optimization, LLC, New Providence, NJ 07974 (dmg@acm.org).

be thought of as an illustration of what can be done today. It is an exercise to test the limits of the AMPL control language and may be used as both a teaching tool and a work bench for the development of algorithms.

The plan of this chapter is as follows. The underlying financial model is briefly developed in section 10.2. Section 10.3 reviews basic principles of stochastic programming, section 10.4 introduces algebraic modeling languages and AMPL in particular, and section 10.5 describes the overall system in greater detail.

10.2 The problem

The problem underlying the system is a debt management problem similar to those facing large public utilities. A set of deterministic (or stochastic) cash flow targets must be met at a minimum cost for each of T periods into the future by issuing, servicing, and repaying suitable debt instruments in various markets, subject to operational and other institutional constraints. Since the utility may borrow in different markets, there are two sources of risk: interest rate risk and exchange rate risk.

A full description of this problem appears in [8], and an AMPL model file is contained in the appendix. We give an abbreviated mathematical formulation below.

Notation:

T is the length of planning period or *horizon*.

$s, t = 0, \ldots, T$ denote time periods.

$k = 1, \ldots, K$ denotes an available debt type.

$e_j := (e_{j1}, e_{j2}, \ldots, e_{jT})$, $j = 1, \ldots, J$, denotes a sequence of (rate) events or *scenario*.

The subscript (j, t) on a variable or parameter signifies dependence on the *evolution* of event sequence e_j up to (and including) period t, in other words, on $(e_{j1}, e_{j2}, \ldots, e_{jt})$.

E_t denotes the set of all distinct subscripts (j, t) that can occur in period t.
($E_T = \{(1, T), \ldots, (J, T)\}$ and E_1 is the singleton set $\{(1,1)\}$.)

Decision Variables (all nonnegative):

$B^k_{(j,t)}$ dollar amount at par of debt type k borrowed in period t.

$O^{k,s}_{(j,t)}$ dollar amount at par of debt type k borrowed in period s and outstanding at the end of period t.

$R^{k,s}_{(j,t)}$ dollar amount at par of debt type k borrowed in period s and retired in period t.

$S_{(j,t)}$ dollar value of surplus cash balance at the end of period t.

Parameters:

$r^{k,s}_{(j,t)}$ service cost in period t per dollar outstanding at the end of period $t-1$ of debt type k issued in period s.

$g^{k,s}_{(j,t)}$ cash outflows per dollar for debt type k issued in period s, if retired during period t.

$v^{k,s}_{(j,T)}$ market value (in base currency) per dollar of debt of type k borrowed in period s and outstanding at the end of period T.

$\rho^{k}_{(j,t)}$ exchange rate of foreign currency per unit of base currency appropriate to debt type k in period t.

$i_{(j,t)}$ interest paid in period t per dollar of surplus cash balance at the end of period $t-1$.

p_j probability of event sequence e_j, $j = 1, \ldots, J$. ($\sum_{j=1}^{J} p_j = 1.$)

f^{k}_{t} issue costs (excluding premium or discount) per dollar borrowed of debt type k issued in period t.

C_t cash requirement for period t. If negative, C_t indicates an operating surplus.

M_t maximum allowable cash outflows for debt service in period t.

N_t maximum total borrowing over all debt types in period t.

q^{k}_{t} minimum borrowing of debt type k in period t.

Q^{k}_{t} maximum borrowing of debt type k in period t.

L_t minimum dollar amount of debt (at par) retired in period t.

U_t maximum dollar amount of debt (at par) retired in period t.

O^{k}_{0} initial amount of debt type k outstanding (borrowed before the start of the planning period).

S_0 initial cash surplus.

Objective:

$$\min \sum_{j=1}^{J} p_j \left\{ \sum_{k=1}^{K} \sum_{t=0}^{T} v^{k,t}_{(j,T)} O^{k,t}_{(j,T)} - S_{(j,T)} \right\}$$

(expected cost of retiring outstanding debt at end of period T).

Constraints:

$$C_t = \sum_{k=1}^{K} \rho^{k}_{(j,t)} \left\{ (1 - f^{k}_{t}) B^{k}_{(j,t)} - \sum_{s=0}^{t-1} \left[r^{k,s}_{(j,t)} O^{k,s}_{(j,t-1)} + g^{k,s}_{(j,t)} R^{k,s}_{(j,t)} \right] \right\}$$
$$+ S_{(j,t-1)} + i_{(j,t)} S_{(j,t-1)} - S_{(j,t)} \qquad \text{for } (j,t) \in E_t \text{ and } t = 1, \ldots, T.$$

(cash requirements)

$$O^{k,s}_{(j,t)} - O^{k,s}_{(j,t-1)} + R^{k,s}_{(j,t)} = 0 \text{ for all } k, (j,t), \text{ and } s < t,$$
$$O^{k,t}_{(j,t)} - B^{k}_{(j,t)} = 0 \text{ for all } k \text{ and } (j,t). \qquad \text{(debt inventory by type)}$$

$$\sum_{k=1}^{K} \sum_{s=0}^{t-1} r^{k,s}_{(j,t)} O^{k,s}_{(j,t-1)} - i_{(j,t)} S_{(j,t-1)} \le M_t \text{ for all } (j,t),$$

(maximum cash outflows for debt service)

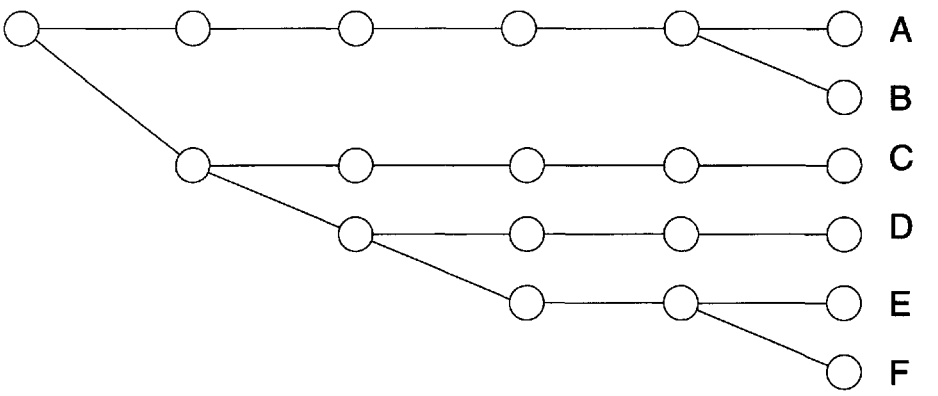


Figure 10.1. *A six-stage scenario tree.*

$$\sum_{k=1}^{K} \rho^k_{(j,t)} B^k_{(j,t)} \leq N_t \text{ for all } (j,t). \qquad \text{(maximum total borrowing)}$$

$$\text{Either } B^k_{(j,t)} = 0 \text{ or } q^k_t \leq B^k_{(j,t)} \leq Q^k_t \text{ for all } k, (j,t). \qquad \text{(maximum and minimum debt issue size)}$$

$$L_t \leq \sum_{k=1}^{K} \sum_{s=0}^{t-1} R^{k,s}_{(j,t)} \leq U_t \text{ for all (j,t).} \qquad \text{(maturity smoothing)}$$

This problem is an instance of the multistage stochastic recourse problem.

10.3 Scenarios and nonanticipativity

We will assume that a finite set of scenarios has been specified and that the problem will be formulated using only these scenarios. One key requirement is that the decision vector must be nonanticipative, that is, it may not anticipate specific realizations of future events. Hence scenarios that share a common history up to a point must have identical solutions until the information paths diverge. (The information flow is best represented by an event tree, as in Figure 10.1.) In this formulation the nonanticipativity is handled implicitly: Where only one set of data is available (and shared by multiple scenarios), only one set of decision variables is defined, so the nonanticipativity constraints are satisfied automatically.

For the purpose of this approach it is most useful to think of a scenario as being identified by a leaf node in the event tree. Each scenario has a starting point where it branches from a parent scenario. The starting point is the first stage in which a scenario has data that differ from those of the parent. One scenario is distinguished by the fact that it has no parent. This scenario is the *root scenario* and starts in period 1. Table 10.1 shows the starting period and parent for each scenario in the event tree of Figure 10.1. (The selection of the root scenario is arbitrary up to a point; we use the convention that the tree is to be

Table 10.1. *Starting period and parent scenario for the event tree of Figure* 10.1.

Scenario	Starting period	Parent
A	1	(root)
B	6	A
C	2	A
D	3	C
E	4	D
F	6	E

Table 10.2. *Label matrix for the event tree of Figure* 10.1.

	1	2	3	4	5	6
A	A	A	A	A	A	A
B	A	A	A	A	A	B
C	A	C	C	C	C	C
D	A	C	D	D	D	D
E	A	C	D	E	E	E
F	A	C	D	E	E	F

read from top to bottom and the topmost scenario in each bundle acts as a parent for all the others.)

Nonanticipativity can also be modeled explicitly, by setting up additional decisions, which are then made to coincide by enforcing the appropriate nonanticipativity constraints. Of course the resulting mathematical program is larger, but some algorithms may take advantage of the added structure (see, e.g., [15]).

The main advantage of the implicit formulation is that it reduces redundancies in the data handling, which makes for smaller data tables and reduced risk of misspecification when the data are modified. So for data representation it is desirable to use the implicit formulation. On the other hand, when one wants to report the solution, particularly historical information along a scenario path, then the explicit formulation may be more convenient.

It is possible to go from the implicit to the explicit representation using database queries. One such approach was described by Lane and Hutchinson [14]. Their label matrix shows for each scenario and each time stage the scenario in which the corresponding data are to be found. For instance, the label matrix for the tree of Figure 10.1 is given in Table 10.2.

10.4 Algebraic modeling languages

Algebraic modeling languages have seen extensive use in building (deterministic) linear and nonlinear programs. The aim is to use a problem description that is as close to the mathematical formulation as possible, making allowance for the limited character set avail-

able on a computer keyboard. Examples of algebraic modeling languages are GAMS [4], AMPL [7], MPL [13], and AIMMS [3]. Some algebraic modeling languages, AMPL among them, encourage the separation of model and data. This is a powerful concept, allowing the development and debugging of a model to occur on a small instance (small data set) while permitting easy substitution of a production-sized data set later on. Only the data tables need to be exchanged in going from one instance to another. Algebraic modeling languages further provide for consistency checks and data verification.

Some algebraic modeling languages also permit data to be drawn from relational databases. We will work extensively with AMPL, which is very convenient for its expressivity and scope of language features. AMPL allows database access through ODBC connectivity and has a control language in which algorithm development may take place.

To illustrate one of the many features of AMPL, we give the following elegant recursion that allows the automatic generation of the label matrix:

```
param label {s in SCENARIOS, t in PERIODS} symbolic in SCEN :=
    (if t >= starttime[s]
     then s
     else label[parent[s],t]);
```

This code specifies that for every scenario s and every time period t the information is to be found in the scenario itself if the start time of scenario s occurs no later than in period t and is to be inherited from the parent scenario otherwise.

10.4.1 AMPL control language

An early add-on to the AMPL system, the AMPL control language provides looping and control structures that can be used for the development of algorithms (see the AMPL website [6] for a description and examples). The model and data provided in conventional AMPL files can be configured into multiple problems, and data items can be dynamically updated. This permits a fairly sophisticated implementation of nested Benders decomposition (see [1]).

Another major feature is the linking to external databases for input as well as output. Using AMPL's built-in string-handling facility, this includes the possibility of dynamic SQL queries generated at run time.

10.5 System description

The system was designed as a visualization aid for stochastic programming and runs under Windows on an ordinary desktop PC. It uses an MS Access database and MS Excel spreadsheets to hold the data tables as well as the decision variables generated by the solver. This makes available to the end user the full power of Excel's graphics capabilities for report generation. Further system requirements can be handled through add-ins or external pieces of software (such as a scenario generator).

The model is written in AMPL, including an implementation of nested Benders decomposition. The resulting subproblems can be solved by any of the linear programming solvers connected to AMPL, such as CPLEX 7.1. The appendix shows the AMPL code

necessary to set up the model, retrieve the data from the database, implement the nested Benders decomposition algorithm, and write the generated solution back to Excel.

10.6 Future work

During the preparation of this project, several shortcomings were identified. First, AMPL's facility for treating generic entity names is incomplete. As explained in a little more detail in the appendix, specific problem knowledge (i.e., particular names of variable and constraint classes) are needed when creating the cuts for the decomposition algorithm. This has the disadvantage that the decomposition loop must be implemented afresh for every new model class. On the other hand, it turned out that special tuning of the cut-creation routines could identify sparsity patterns in the cuts, leading to storage efficiencies that would otherwise not have been realized. Hence we propose an extension of AMPL's generic name handling that would enable development and distribution of completely general algorithms.

Other algorithms that would be interesting to try are the multicut variant of [2] and the stochastic decomposition algorithm of [9]. Finally, since decomposition algorithms exhibit a very natural application of parallel computing, it would be interesting to apply the AMPL shell command to permit the solution of different subproblems on distributed computers, perhaps using ideas similar to the CONDOR net described by Ferris and Munson [5].

Appendix

The appendix contains the AMPL code, in four parts. The financial model looks as follows:

```
###### MIDAS model file

### SETS AND PARAMETERS FOR DYNAMIC STRUCTURE ###
param T > 0;                              # number of time periods
set   PERIODS := 0..T ordered;

### SETS AND PARAMETERS FOR SCENARIO STRUCTURE ###
set   SCEN ordered;
param root symbolic in SCEN := first(SCEN);   # root scenario
param prob {SCEN} >=0, <= 1;          # path probability
   check: 0.99999 < sum {s in SCEN} prob[s] < 1.00001;

param parent {SCEN} symbolic in SCEN;     # parent scenario
param starttime {SCEN} > 0 integer;
#   first period in which the scenario has data that differ from those
#   of the parent scenario. There should be exactly one scenario that
#   starts in the first period: the root scenario
    check: card {s in SCEN : starttime[s] = 1} = 1;

### Derived sets and parameters ###

# First we set up the label matrix of Lane and Hutchinson [14]
param label {w in SCEN, t in 1..T} symbolic in SCEN :=
      (if t >= starttime[w]
       then w
       else label[parent[w],t]);
```

```
set TREE := { w in SCEN, t in 1..T: t >= starttime[w]} ;
set SUCC { (w,t) in TREE} :=
       { (w1,t1) in TREE  : t1 = t + 1 and label[w1,t] = w};

set HIST := { w in SCEN, s in 0..T, t in 1..T:
                                        s <= t and t >= starttime[w]};
param previous { (w,t) in TREE: t > 1}  :=
       (if t = starttime[w] then parent[w] else w);

param pathprob { (w,t) in TREE } :=
   (if t = T
       then prob[w]
       else sum {(w2,t2) in SUCC[w,t]} pathprob[w2,t2]);
param condprob { (w,t) in TREE } :=
   ( if t = 1
        then pathprob[w,t]
        else (if t = starttime[w]
                 then pathprob[w,t]/pathprob[parent[w],t-1]
                 else pathprob[w,t]/pathprob[w,t-1]        ));

### SETS AND PARAMETERS FOR FINANCIAL MODEL ###
set DEBT;
param svc_cost { DEBT, HIST}  >= 0;     # debt service cost per dollar of debt
                                        # includes interest per period and
                                        # sinking fund contributions
param ret_cost { DEBT, HIST};           # retirement discount/premium
param end_val  { DEBT, SCEN, PERIODS}; # market value (in base currency)
                                        # per dollar of debt
param exch_rate{ DEBT, TREE};           # exchange rate of foreign currency
                                        # per unit of base currency
param cash_int { TREE}  >=0;            # interest on surplus cash
param issue_cost { DEBT,1..T} >=0;
param cash_req   {      1..T}    ;      # cash requirements
param max_cost   {      1..T} >=0;      # maximum debt cost per period
param max_Tbor   {      1..T} >=0;      # maximum total borrowing
param min_bor    { DEBT,1..T} >=0;      # minimum borrowing of a debt type
param max_bor    { DEBT,1..T} >=0;      # maximum borrowing of a debt type
param min_ret    {      1..T} >=0;      # minimum amount retired
param max_ret    {      1..T} >=0;      # maximum amount retired
param init_debt  { DEBT     } >=0;      # initial outstanding debt
param init_cash               >=0;      # initial surplus cash

###  VARIABLES ###
var Cash   { TREE}  >= 0;               # Amount of cash on hand
var Borrow { DEBT, TREE} >= 0;          # New borrowing
var Outst  { DEBT, HIST} >= 0;          # Outstanding debt by type
var Retire { DEBT, HIST} >= 0;          # Amount of debt retired
var Delta  { DEBT, TREE} >= 0, <= 1;
    # The deltas are used to code minimal and maximal borrowing.
    # They should be defined as binary, but in the decomposition
    # we solve the LP relaxation instead.

###  OBJECTIVE ###
minimize endvalue:
    sum { w in SCEN}
        prob[w] * ( sum { d in DEBT, s in 0..T}
                            end_val[d,w,s] * Outst [d,w,s,T]
```

```
                    - Cash [w,T]);

### CONSTRAINTS ###
subject to balance { (w,t) in TREE} :
    cash_req[t] = sum { d in DEBT}
        exch_rate[d,w,t] * ( (1-issue_cost[d,t])*Borrow[d,w,t]
            - sum { s in 0..t-1}
                  (svc_cost[d,w,s,t] *
                       ( if t = 1
                         then init_debt[d]
                         else Outst[d,previous[w,t],s,t-1])
                    + ret_cost[d,w,s,t]*Retire[d,w,s,t]      ) )
            + ( if t = 1
                then init_cash
                else (1 + cash_int[w,t]) * Cash[previous[w,t],t-1])
            - Cash[w,t];

subject to inventory { d in DEBT, (w,s,t) in HIST} :
    Outst[d,w,s,t] =
        (if s = t
         then Borrow[d,w,s]
         else (if t = 1
               then init_debt[d]                  - Retire[d,w,s,t]
               else Outst[d,previous[w,t],s,t-1] - Retire[d,w,s,t]));

subject to debt_cost { (w,t) in TREE} :
    sum { d in DEBT, s in 0..t-1}
        svc_cost[d,w,s,t] * ( if t=1
                              then init_debt[d]
                              else Outst[d,previous[w,t],s,t-1] )
          - cash_int[w,t] * ( if t=1
                              then init_cash
                              else Cash[previous[w,t],t-1] )
      <= max_cost[t];

subject to market_max { (w,t) in TREE} :
    sum { d in DEBT}  (exch_rate[d,w,t]*Borrow[d,w,t]) <= max_Tbor[t];

subject to max_issue { d in DEBT, (w,t) in TREE} :
    Borrow[d,w,t] - Delta[d,w,t] * max_bor[d,t] <=0;

subject to min_issue { d in DEBT, (w,t) in TREE} :
    Borrow[d,w,t] - Delta[d,w,t] * min_bor[d,t] >=0;

subject to mat_smooth { (w,t) in TREE} :
    min_ret[t] <= sum { d in DEBT, s in 0..t-1}  Retire[d,w,s,t]
                                                  <= max_ret[t];
```

The data for a particular instance of this problem are held in an Access database. The following code defines the table links and reads the data into AMPL:

```
# ----------------------------------------------------------

option MDBFILE "c:\datafi~1\research\slp_proj\midas_2k.mdb";

# ----------------------------------------------------------
# Define the tables:
```

```
table ScalarParams IN "ODBC" ($MDBFILE): [], T, init_cash ~ "Init_cash";

table SortedScenarioData IN "ODBC" ($MDBFILE)
    "SQL=SELECT scenarios, prob, parent, starttime \
         FROM ScenarioData ORDER BY scenarios;" :
    SCEN <- [scenarios], prob, parent, starttime;

table InitialDebt  IN "ODBC" ($MDBFILE):
    DEBT <- [debt_types], init_debt;

table CashInterest IN "ODBC" ($MDBFILE):
   [scenarios, current_period], cash_int;

table ServicingAndRetirement IN "ODBC" ($MDBFILE):
   [debt_types, scenarios, issue_period, current_period],
    ret_cost, svc_cost;

table EndValue IN "ODBC" ($MDBFILE):
   [debt_types, scenarios,   issue_period], end_val;

table ExchangeRates IN "ODBC" ($MDBFILE):
   [debt_types, scenarios, current_period], exch_rate;

table CashReqsAndSmoothing   IN "ODBC" ($MDBFILE):
   [current_period],
    cash_req, max_cost, min_ret, max_ret, max_Tbor;

table IssuingData IN "ODBC" ($MDBFILE):
   [debt_types, current_period],
    min_bor, max_bor, issue_cost;

# ----------------------------------------
# Read the input data:
 read table ScalarParams;
 read table SortedScenarioData;
 read table InitialDebt;
 read table CashInterest;
 read table ServicingAndRetirement;
 read table EndValue;
 read table ExchangeRates;
 read table CashReqsAndSmoothing;
 read table IssuingData;
```

The AMPL control code for the nested Benders decomposition is given below:

```
# ---------------------------------------

option solver cplexamp;

### Start by defining additional sets, parameters and relationships
### needed for the decomposition
set NODES_in_t {t in 1..T} := { (c,s) in TREE : s = t};

set DESCEND { (c,s) in TREE } := { (w,t) in TREE :
    t > s and t = s+1 and label[w,s] = c };

set NONTERM := { (c,s) in TREE : s < T}; # only non-terminal nodes have cuts
```

```
param n_cuts {NONTERM} >= 0 integer, default 0;
# tracks the number of cuts placed in each subproblem

var Exp_Future_Cost {NONTERM} default -Infinity;   # theta columns

set CUT_NODES := { (c,s) in TREE: s > 1};
# cut-generating nodes are those in periods other than the first

### The following parameters control the flow of the algorithm
### ===========================================================
param inhibit_opt_cut {TREE} binary, default 0;
# do not make optimality cuts if there are infeasible descendants

param new_data {TREE} binary, default 1;
# new primal information available; re-solve this problem on forward pass

param new_dual {TREE} binary, default 0;
# new dual information available; re-solve this problem on backward pass

param status {TREE} integer, default 0;
# this is trivariate
# = 1 - look at this node
# = 0 - don't look at this node
# =-1 - node is currently infeasible (so generate a feasibility cut)

param theta_enabled {NONTERM} binary, default 0;
# theta column is ignored until at least one optimality cut has been placed

param theta_weight {(c,t) in NONTERM, 1..n_cuts[c,t]} binary;
# distinguishes between feasibility and optimality cuts

param new_opt_cuts binary;
# indicates whether new cuts were generated during the current iteration

### Sets and Parameters to hold the cut coefficients (including RHS)
### ================================================================
set   ALLCUTS := { (c,t) in NONTERM, k in 1..n_cuts[c,t] };

param cut_coef_outst { (c,t,k) in ALLCUTS, DEBT, s in 0..t};
param cut_coef_cash  { ALLCUTS };
param cut_dual { ALLCUTS };
param cut_rhs  { ALLCUTS };

### These parameters are used to temporarily hold the dual
### information until the next cut has been placed
param balance_price {CUT_NODES};
param inventory_price {DEBT, (w,s,t) in HIST: (w,t) in CUT_NODES};
param debt_price {CUT_NODES};
param market_max_price {CUT_NODES};
param smooth_price {CUT_NODES};
param delta_price {DEBT, CUT_NODES};

### Here we define the cuts
### ======================
subj to
    Cut_Defn { (w,t,k) in ALLCUTS }:
        sum {d in DEBT, s in 0..t}
```

```
            (cut_coef_outst[w,t,k,d,s] * Outst[d,w,s,t])
        +    cut_coef_cash [w,t,k    ] * Cash [  w,  t]
        +  theta_weight[w,t,k]*Exp_Future_Cost[w,t] <= cut_rhs[w,t,k];

### Now set up the objective for each node
minimize Current_Cost { (w,t) in TREE} :
        ( if t = T
          then pathprob[w,t] * ( sum { d in DEBT, s in 0..T}
                                      end_val[d,w,s] * Outst [d,w,s,T]
                               - Cash [w,T])
          else  - theta_enabled[w,t]*Exp_Future_Cost[w,t]);

### Define the subproblems
### ======================
problem Sub {(w,t) in TREE}:
                Cash[w,t],
    { d in DEBT} Borrow[d,w,t],
    { d in DEBT} Delta [d,w,t],
    { d in DEBT, v in 0..t} Outst [d,w,v,t],
    { d in DEBT, v in 0..t} Retire[d,w,v,t],

    { d in DEBT, v in 0..t} inventory[d,w,v,t],
                balance[w,t],
                debt_cost [w,t],
                market_max[w,t],
    { d in DEBT} max_issue[d,w,t],
    { d in DEBT} min_issue[d,w,t],
                mat_smooth[w,t],

    {k in 1.. if t < T then n_cuts[w,t] else 0} Cut_Defn[w,t,k],
    {     1.. if t < T then      1      else 0} Exp_Future_Cost[w,t],
    Current_Cost[w,t];

### The actual Benders decomposition starts here
### ============================================

### Initialization
### ==============
set FORWARD  ordered    within   PERIODS := 1..T-1;
set BACKWARD ordered by reversed PERIODS := 1..T-1;
let status[1,root] := 1;

# Outer loop using fast-forward-fast-back
# ---------------------------------------
for outer_loop {itcount in 1..50} { printf "\nITERATION %d\n\n", itcount;

# Forward pass
# ------------
   for {t_curr in FORWARD} {
      for {(u,s) in NODES_in_t[t_curr]} {
         let inhibit_opt_cut[u,s] := 0;
         if status[u,s] = 1 then {

            problem Sub[u,s]; # Set up the problem associated with node [u,s]
            solve Sub[u,s];   # Solve the problem for node [u,s]

            if solve_result = 'solved'
```

```
            then {let {(u1,s1) in DESCEND[u,s]} status[u1,s1] := 1;
                  let status[u,s] := 0;
            }
            else {let {(u1,s1) in DESCEND[u,s]} status[u1,s1] := 0;
                  let status[u,s] := -1;
            };
         }          # end if status
      }        # end for (u,s)
   }       # end for t_curr

# Backward pass
# -------------
   for {t_curr in BACKWARD} {
      let new_opt_cuts := 0;

      for {(u1,s1) in NODES_in_t[t_curr]} {
         let inhibit_opt_cut[u1,s1] :=
             max {(u,s) in DESCEND[u1,s1] } inhibit_opt_cut[u,s];
         for solver_loop {(u,s) in DESCEND[u1,s1]} {
            if status[u,s] = 1
            then {
               problem Sub[u,s];
               solve Sub[u,s];
               let status[u,s] := 0;

### Store the dual information
### --------------------------
### (This requires specific knowledge of the problem (constraint classes))

               let balance_price[u,s] := balance[u,s].dual;
               let debt_price[u,s] := debt_cost[u,s].dual;
               let market_max_price[u,s] := market_max[u,s].dual;
               let smooth_price[u,s] := mat_smooth[u,s].dual;
               let {d in DEBT} delta_price[d,u,s] := Delta[d,u,s].rc;
               let {d in DEBT, s2 in 0..s}
                   inventory_price[d,u,s2,s] := inventory.dual[d,u,s2,s];

               if (u,s) in NONTERM then {
                  let {k in 1..n_cuts[u,s]}
                     cut_dual[u,s,k] := Cut_Defn[u,s,k].dual;
                  let {(u2,s2) in DESCEND[u,s]} status[u2,s2] := 1;
               };

               if solve_result = 'infeasible'
               then {
                  let n_cuts[u1,s1] := n_cuts[u1,s1] + 1;
                  let inhibit_opt_cut[u1,s1] := 0;
                  let theta_weight[u1,s1,n_cuts[u1,s1]] := 0;

### RHS and coefficients for a feasibility cut
### ------------------------------------------
### Here we access the subdiagonal matrix in blocks corresponding to the
### different row and variable classes. This code is model specific.

                  let cut_coef_cash[u1,s1,n_cuts[u1,s1]] :=
                      (1 + cash_int[u,s]) * balance_price[u,s]
                         - cash_int[u,s]  * debt_price[u,s];
```

```
                let {d in DEBT, s2 in 0..s1 }
                    cut_coef_outst[u1,s1,n_cuts[u1,s1],d,s2] :=
                    svc_cost[d,u,s2,s]*(debt_price[u,s]-balance_price[u,s])
                        + inventory_price[d,u,s2,s];

                let cut_rhs[u1,s1,n_cuts[u1,s1]] :=
                   balance_price[u,s] * cash_req[s]
                       + debt_price[u,s] * max_cost[s]
                       + market_max_price[u,s] * max_Tbor[s]
                       - smooth_price[u,s] *
                           sum {d in DEBT, s2 in 1..s} Retire[d,u,s2,s]
                       - sum { d in DEBT } delta_price[d,u,s]
                       - if (u,s) in NONTERM
                         then
                            sum { k in 1..n_cuts[u,s] }
                               cut_rhs[u,s,k]*cut_dual[u,s,k];
### ------------------------------------------------------------

                let status[u1,s1] := 1;
                break solver_loop;

             }               # end if solve_result
          }             # end if status
       }           # end solver_loop

       if inhibit_opt_cut[u1,s1] = 0
       then {
          if not(theta_enabled[u1,s1]) or Exp_Future_Cost[u1,s1] +
             sum {(u2,s2) in DESCEND[u1,s1] }
                Current_Cost[u2,s2] >
                (if s1 = T-1
                 then 0.00000001
                 else 0.0001*abs(Exp_Future_Cost[u1,s1]))
          then
          {
             let n_cuts[u1,s1] := n_cuts[u1,s1] + 1;
             let theta_enabled[u1,s1] := 1;
             let theta_weight[u1,s1,n_cuts[u1,s1]] := 1;
             let new_opt_cuts :=1;

### RHS and coefficients for an optimality cut
### ------------------------------------------
### This code is model-specific, just like the feasibility cuts

             let cut_coef_cash[u1,s1,n_cuts[u1,s1]] :=
                sum {(u2,s2) in DESCEND[u1,s1] }
                ((1 + cash_int[u2,s2]) * balance_price[u2,s2]
                   - cash_int[u2,s2]  * debt_price[u2,s2]);
             let {d in DEBT, s3 in 0..s1}
                cut_coef_outst[u1,s1,n_cuts[u1,s1],d,s3] :=
                   sum {(u2,s2) in DESCEND[u1,s1] }
                      (svc_cost[d,u2,s1,s2] * (debt_price[u2,s2]
                                              - balance_price[u2,s2])
                       + inventory_price[d,u2,s3,s2]);
             let cut_rhs[u1,s1,n_cuts[u1,s1]] :=
                sum {(u2,s2) in DESCEND[u1,s1] }
                (balance_price[u2,s2] * cash_req[s2]
```

```
                        + debt_price[u2,s2] * max_cost[s2]
                        + market_max_price[u2,s2] * max_Tbor[s2]
                        - smooth_price[u2,s2] *
                            sum {d in DEBT, s3 in 1..s2} Retire[d,u2,s3,s2]
                        - sum { d in DEBT } delta_price[d,u2,s2]
                        - if (u2,s2) in NONTERM
                          then
                             sum { k in 1..n_cuts[u2,s2] }
                                cut_rhs[u2,s2,k]*cut_dual[u2,s2,k]);
### ------------------------------------------------------------

               let status[u1,s1] := 1;
            };    # end if not theta_enabled
         }    # end if inhibit_opt_cut
      }    # end solver loop
   }    # end for t_curr

# Convergence test
# ----------------

   if new_opt_cuts = 0 then break;

   printf "\nForward pass\n\n";

}  # end outer loop

printf "\nOPTIMAL SOLUTION FOUND\n";
```

Finally, the optimal solution found by CPLEX is written back to an Excel spreadsheet for further processing and report generation.

```
# ------------------------------------------------------

option OUTFILE "c:\datafi~1\research\slp_proj\midas_2k.xls";

# ------------------------------------------------------
# Define the tables:

table BorrowingDecisions OUT "ODBC" ($OUTFILE):
   {d in DEBT,     s in SCEN, t in 1..T : t >= starttime[s] }
   -> [debt_types, scenarios, current_period],
   Borrow[d,s,t] ~ "New_debt",
   Delta [d,s,t] ~ Delta;

table CashHoldings OUT "ODBC" ($OUTFILE):
   {s in SCEN,    t in 1..T : t >= starttime[s]}
   -> [scenarios, current_period], Cash[s,t] ~ "Cash_holdings";

table InventoryAndRetire OUT "ODBC" ($OUTFILE):
   { d in DEBT, w in SCEN, s in 0..T, t in 1..T :
       s <= t and t >= starttime[w]}
   -> [debt_types, scenarios, issue_period, current_period],
    Outst [d,w,s,t] ~ "Debt_outstanding",
    Retire[d,w,s,t] ~ "Debt_retired";

# ----------------------------------------
# Write the output data:
```

```
write table BorrowingDecisions;
write table CashHoldings;
write table InventoryAndRetire;
```

Acknowledgments

Much of the computation presented was performed while the first author enjoyed the hospitality of the Centre of Advanced Study at the Norwegian Academy of Science and Letters.

Bibliography

[1] J. R. Birge, *Decomposition and partitioning methods for multi-stage stochastic linear programs*, Oper. Res., 33 (1985), pp. 989–1007.

[2] J. R. Birge and F. V. Louveaux, *A multicut algorithm for two-stage stochastic linear programs*, Eur. J. Oper. Res., 34 (1988), pp. 384–392.

[3] J. Bisschop and R. Entriken, *AIMMS: The Modelling System*, Paragon Decision Technology, Haarlem, The Netherlands, 1993.

[4] A. Brooke, D. Kendrick, and A. Meeraus, *GAMS: A User's Guide*, 2nd ed., Scientific Press, South San Francisco, CA, 1992.

[5] M. C. Ferris and T. S. Munson, *Modelling languages and Condor: Metacomputing for optimization*, Math. Program., 88 (2000), pp. 487–505.

[6] R. Fourer, *AMPL Web Site*, 2001; available online from http://www.ampl.com.

[7] R. Fourer, D. M. Gay, and B. Kernighan, *AMPL: A Modeling Language for Mathematical Programming*, Scientific Press, South San Francisco, CA, 1993.

[8] H. I. Gassmann and A. M. Ireland, *Scenario formulation in an algebraic modelling language*, Ann. Oper. Res., 59 (1995), pp. 45–77.

[9] J. L. Higle and S. Sen, *Stochastic Decomposition*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[10] IBM, *Passing Your Model to OSL Using Mathematical Programming System (MPS) Format*; available online from http://www6.software.ibm.com/sos/features/featur11.htm.

[11] IBM, *Optimization Solutions and Library*; available online from http://www-3.ibm.com/software/data/bi/osl/index.html.

[12] *ILOG CPLEX* 7.0 *User's Manual*, ILOG, Inc., Mountain View, CA, 2000.

[13] B. Kristjansson, *MPL Modelling System User Manual* (*Version* 2.8), Maximal Software, Inc., Arlington, VA, 1993.

[14] M. LANE AND P. HUTCHINSON, *A model for managing a certicificate of deposit portfolio under uncertainty*, in Stochastic Programming, M. A. H. Dempster, ed., Academic Press, London, 1980, pp. 473–495.

[15] I. J. LUSTIG, J. M. MULVEY, AND T. J. CARPENTER, *Formulating two-stage stochastic programs for interior point methods*, Oper. Res., 39 (1991), pp. 757–770.

This page intentionally left blank

Part II

Stochastic Programming Applications

This page intentionally left blank

Chapter 11

Introduction to Stochastic Programming Applications

Horand I. Gassmann, Sandra L. Schwartz,† Stein W. Wallace,‡ and William T. Ziemba†*

This section of the book contains 21 stochastic programming applications presented in five sections: (1) production, supply chain, and scheduling; (2) gaming; (3) environment and pollution; (4) finance; and (5) telecom and electricity.

The section on production, supply chain, and scheduling presents five applications.

In Chapter 12, Powell and Topaloglu address the problem of managing fleets of trailers, containers, boxcars, taxicabs, locomotives, and business jets. In many applications, requests to be moved from one location to another arrive randomly over time. Since it may take days to move from one location to the next, it is often useful (even necessary) to reposition equipment before a customer demand is known. They use the techniques of stochastic optimization to solve two-stage and multistage fleet management problems, using the problem of rail car distribution as the motivating application. Rail car distribution is characterized by a high degree of randomness: in the orders being placed by the customers, in the empty cars becoming available, and in the transit times between locations. Car distribution problems also introduce other issues that are not typically addressed in the stochastic programming literature, such as lagged information processes, the need for discrete solutions, and the challenges posed when the relevant attributes of a car are allowed to grow (a common progression when solving real problems). This chapter is intended to introduce the reader to a real problem, providing a bridge between the classical literature of stochastic programming and the modeling and algorithmic challenges that would arise while solving

*School of Business Administration, Dalhousie University, Halifax, NS, B3H 3J5, Canada (horand.gassmann@dal.ca).

†Saunders School of Business, University of British Columbia, Vancouver, BC, V6T 1Z2, Canada (schwartz@interchange.ubc.ca, ziemba@interchange.ubc.ca).

‡Molde University College, N-6402 Molde, Norway (stein.wallace@himolde.no).

a real problem.

In "Modeling Production Planning and Scheduling under Uncertainty" (Chapter 13), Alonso-Ayuso, Escudero, and Ortuño present a set of production planning and scheduling application cases, where the main uncertain parameters are the product price and demand, raw material supply cost, production cost, operation execution duration, resource requirement by the operations execution, and resource availability over a time horizon. The models range from a continuous model to mixed and pure 0–1 models; some are two-stage models and others are multistage models. Compact and splitting variable mathematical representations are proposed for the models. They are very amenable to such algorithmic approaches as Benders decomposition, Lagrangian decomposition, and branch-and-fix coordination for mixed and pure 0–1 models.

In Chapter 14, Tomasgard and Høeg present a supply chain optimization model for the meat industry, developed for Norwegian Meat. The purpose of the model is to achieve better capacity and resource utilization in the supply chain and to reduce the lead times. The focus of the decision support model is to maximize expected profit subject to uncertain demand and shortfall costs. This is done by coordination of production and inventories in different regions and at different levels in the supply chain. The model describes decisions at the operational level. The authors also discuss links to tactical decisions. They describe the underlying problems of operating a supply chain in the meat industry. The mathematical model is based on stochastic linear programming. The authors also discuss how to find the necessary data describing the uncertain demand for the decision support model by linking forecasting and scenario generation using quantile regression.

In Chapter 15, Dupačová and Popela introduce melt control modeled using two-stage and multistage stochastic programming models. Sources of uncertainties are described, and several methods of input generation are presented. The implementation compares decisions and costs obtained by solving stochastic programs with different numbers of stages and a different structure of the scenario tree. The results favor the stochastic programming methodology.

In Chapter 16, Higle and Sen present a stochastic programming model for network resource utilization in the presence of multiclass demand uncertainty. There are numerous applications in which a network of resources serves multiple classes of demand, each unit of which requires the allocation of resources on a collection of links. Such demand is usually classified by the origin-destination pair and by the customer class. One of the main sources of uncertainty in such operations arises from the uncertainty in total demand. For such systems, the admission policy has a direct impact on the revenues generated by the operation. The authors present a stochastic programming model to develop a set of bid prices which can be used for admission control. Simulation experiments reveal that the stochastic programming model can provide substantial revenue benefits over linear programming approaches. They also demonstrate that a bid-price policy based on the stochastic programming model is less prone to large errors than those resulting from the linear programming model. This combined advantage makes stochastic programming a viable alternative to a linear programming approach.

The section on gaming presents two papers.

In Chapter 17, Philpott describes some applications of stochastic optimization to high-performance yacht racing. He describes the ingredients that make up yachting optimization models with a special emphasis on modeling uncertainty. He discusses race modeling

programs and route optimization in both short in-shore yacht races and offshore and ocean racing. His paper concludes with a discussion of the application of stochastic programming to design optimization.

In Chapter 18, Berglann and Flåm discuss stochastic approximation, momentum, and Nash play. Stochastic programming is used to study repeated play of normal-form, noncooperative games. The concerned parties entertain local visions, form linear approximations, and hesitate in making large or swift adjustments. For the purpose of reaching Nash equilibrium, or for learning such play, the authors advocate and illustrate a procedure that combines stochastic gradient projection with the heavy-ball method. What emerges is a coupled, constrained, second-order stochastic process. Some friction feeds into and stabilizes myopic approximations. Convergence to Nash play is obtained under seemingly weak and natural conditions, an important one being that accumulated marginal payoffs remains bounded above.

The section on the environment and pollution contains four papers.

In Chapter 19, King, Somlyódy, and Wets discuss how stochastic programming was applied to lake eutrophication management during the Lake Balaton study—a research program that took place at the International Institute for Applied Systems Analysis during the early 1980s. The lake eutrophication study combined descriptive, simulation, and management optimization models. The sources of stochastic variability are twofold: the incompleteness of the actual historical data series relating phosporus flows to eutrophication processes and the variable impact of weather in the eutrophication process model for the lake. Two optimization approaches to selecting management options are described and compared: a stochastic recourse model, in which the recourse expressed penalties for missing management goals, and a linear programming approach that captured stochastic features of the problem through a linearized expectation-variance model. Of the many interesting aspects of this study (which inspired much original research in algorithms and statistical asymptotics for stochastic programming), one is that certain key features of the solution—the large reed basins—appear only when variance is considered in the objective. This important feature of the solution was observed in the linearized expectation-variance model, on which the original recommendations to the Hungarian government were based, and were later confirmed by the more accurate stochastic recourse formulation.

In Chapter 20, Yohe discusses mitigating anthropogenic climate change. An aggregate economic model is integrated with reduced-form representations of the link between the emission of greenhouse gases and damage attributable to associated climate change and used to examine mitigation policies in a dynamic context. Solutions to deterministic optimization problems inform the design of hedging experiments in stochastic environments where growth trajectories, climate sensitivities, and damage estimates are profoundly uncertain. The same modeling and solution techniques are also employed to explore second-best optimization problems in which concentration targets and/or differential emissions allocations across nations are determined outside of the economic context.

In Chapter 21, Watkins, McKinney, and Morton develop a two-stage stochastic mixed-integer nonlinear program for controlling a groundwater contaminant plume. First-stage decision variables select well locations in the face of uncertainty concerning the underlying hydraulic conductivity field. Second-stage variables select pumping rates at the wells in an attempt to maintain hydraulic head gradients designed to contain the contaminant plume. The objective is to minimize the expected value of a weighted sum of installation and

pumping costs, costs exceeding a prescribed target, and unmet environmental head-gradient constraints.

In Chapter 22, Ermolieva and Ermoliev present catastrophic risk management via flood and seismic risks case studies. They discuss a catastrophic risk management model that takes into account the specifics of catastrophic risks: highly mutually dependent losses, the lack of sufficient information, the need for long-term perspectives and geographically explicit analyses, as well as the involvement of various agents such as individuals, governments, insurers, reinsurers, and investors. As a specific case they consider the seismic activity prone Tuscany region of Italy. Special attention is given to the evaluation of a public loss-spreading program involving partial compensation to victims by the central government and the spreading of risks through a pool of insurers on the basis of location-specific exposures. GIS-based catastrophe models and stochastic optimization methods are used to guide policy analysis with respect to location-specific risk exposures. They use economically sound risk indicators leading to convex stochastic optimization problems strongly connected with nonconvex insolvency constraint and conditional value at risk.

The section on finance contains seven applications.

In Chapter 23, Frauendorfer and Schürle present a multistage stochastic programming model for refinancing mortgages with noncontractual maturity under liquidity restrictions in the market. An extension to the management of other products such as savings accounts is straightforward. The evolution of interest rates is modeled by principal components for short-term and a two-factor mean reversion model with long rate and spread for long-term planning. Barycentric approximation provides tight lower and upper bounds for the original problem with relative discretization errors on the order of one percent.

In Chapter 24, Zenios discusses optimization models for structuring index funds. Market indices are comprehensive measures of market trends. Passive strategies that manage index funds to mimic market trends are prevalent among portfolio managers. He develops optimization models for structuring index funds starting with a review of the basics of market indices and the discussion of two broad model classes for structuring indexed portfolios. The use of multiperiod stochastic optimization models is also discussed for this broad problem class. Creating index funds for international and corporate bond markets helps to clarify the issues and refine the models. Empirical results are given from several applications in tracking an International Government Bond index, the Merrill Lynch Euro Dollar index of corporate bonds, the Salomon Brothers Mortgage index, and an index of callable corporate bonds. The results illustrate the uses of the models, establish their efficacy, and show that, overall, multiperiod stochastic programs add value to the management of indexed funds.

In Chapter 25, Mulvey and Erkan discuss "Decentralized Risk Management for Global P/C Insurance Companies." Recent U.S. legislation provides for the merger of disparate financial organizations: banks, insurance companies, and security firms. Optimizing the overall company performance requires a method for allocating capital and aligning the goals of the divisions with headquarters. Decentralized optimization is an ideal approach for solving the capital allocation problem. An example shows the benefits of optimizing the risk-adjusted profit of each division.

In Chapter 26, MacLean, Zhao, and Ziemba discuss wealth goals investing. Investing in assets in a volatile financial market can present significant downside risk to accumulated capital. That risk can result from the inherent variability of trading prices for assets and also from the errors in estimation of the rates of return. The problem of misdirected

investment strategies based on erroneous forecasts is the motivation for a process control approach to volatility and risk. Upper and lower limits on the capital accumulation process are used to determine whether the current investment strategy continues. If a limit is reached then rebalancing occurs, where returns are re-estimated, new limits are established, and a new strategy is determined. This variable planning horizon approach is compared to the standard value-at-risk methodology, where the time horizon is fixed. In an application to asset allocation involving stocks, bonds, and cash, it is shown that for any value-at-risk strategy there exist process control limits so that the corresponding process control strategy has greater expected return with equivalent downside risk. The advantage in the process control approach comes from intervening (rebalancing) when the wealth process deviates significantly from expectations.

In Chapter 27, Mausser and Rosen discuss scenario-based risk management tools. Risk management requires tools that identify a portfolio's most significant sources of risk and that indicate how potential trades can improve the trade-off between risk and reward. These tools include simple risk analytics that consider each position independently, as well as stochastic optimization models having a broader scope. If a portfolio's losses are assumed to be normally distributed, the relevant risk analytics have closed-form representations, and efficient portfolios are obtainable from the well-known mean-variance model. This assumption often fails to hold in practice, notably for credit risk, for portfolios that contain options, and when risk-factor distributions are fat-tailed. Scenario-based tools provide a practical, general alternative for managing risk. Simulating the portfolio over a set of scenarios yields an empirical loss distribution that can be used to quantify a portfolios risk, and the relevant risk analytics are computed from the simulated data. Their paper describes scenario-based risk management tools and illustrates their use in various market and credit risk problems.

In Chapter 28, Medova and Sembos discuss price protection strategies through the example of an oil company. Crude oil price volatility has a significant impact on the planning decisions and budgets of oil companies. Taking account of such major activities as supply, storage, transformation, and transportation together with trading on the commodity markets, they investigate the influence of random prices and demands. Such problems may be formulated as dynamic stochastic programs with robust first-stage solutions in the face of future price and demand uncertainties. They describe the trading environment and investigate financial hedging policies in coordination with the logistics planning problem.

In Chapter 29, Krokhmal, Uryasev, and Zrazhevsky apply formal risk management methodologies to the optimization of a portfolio of hedge funds. They compare recently developed risk management methodologies: conditional value-at-risk and conditional drawdown-at-risk with mean-absolute deviation, maximum loss, and market neutrality approaches. The common property of considered risk management techniques is that they admit the formulation of a portfolio optimization model as a linear programming problem. Linear programming formulations allow for implementing efficient and robust portfolio allocation algorithms, which can successfully handle optimization problems with thousands of instruments and scenarios. The performance of various risk constraints is investigated and discussed in detail for in-sample and out-of-sample testing of the algorithm. The numerical experiments show that imposing risk constraints may improve the real performance of a portfolio rebalancing strategy in out-of-sample runs. It is beneficial to combine several types of risk constraints that control different sources of risk.

The section on telecom and electricity contains three applications.

In Chapter 30, Gröwe-Kuska and Römisch present a mixed-integer multistage stochastic programming model for the short-term unit commitment of a hydrothermal power system under uncertainty to load, inflow to reservoirs, and prices for fuel and delivery contracts. The model is implemented for uncertain load and tested on realistic data from a German power utility. Load scenario trees are generated by a procedure consisting of two steps: (i) simulation of load scenarios using an explicit representation of the load distribution and (ii) construction of a tree out of these scenarios. The dimension of the corresponding mixed-integer programs ranges up to 200,000 binary and 350,000 continuous variables. The model is solved by a Lagrangian-based decomposition strategy exploiting the loose coupling structure. Solving the Lagrangian dual by a proximal bundle method leads to a successive decomposition into single-unit subproblems, which are solved by specific algorithms. Finally, Lagrangian heuristics are used to construct nearly optimal first-stage decisions.

In Chapter 31, Deng and Oren propose a trinomial lattice approach for "real options"-based valuation of electricity generation capacity incorporating operational constraints such as start-up cost, ramp-up cost, and operating level–dependent heat rate. Stochastic prices of electricity and fuel are represented by a recombining trinomial tree. Generators are modeled as a strip of cross-commodity call options with a delay and a cost imposed on each option exercise. The authors illustrate implications of operational characteristics on valuation of generation assets under different price models for the underlying energy commodities. They find that the impact of operational constraints on real asset valuation is dependent on both the model specification and the nature of operating characteristics.

In Chapter 32, Gaivoronski surveys different optimization problems under uncertainty which arise in telecommunications. Three levels of decisions are distinguished: design of structural elements of telecommunication networks, top level design of telecommunication networks, and design of optimal policies of a telecommunication enterprise. Examples of typical problems from each level show that the stochastic programming paradigm is a powerful approach for solving telecommunication design problems.

Chapter 12

Fleet Management

*Warren B. Powell** *and Huseyin Topaloglu*†

12.1 Introduction

The fleet management problem, in its simplest form, involves managing fleets of equipment to meet customer requests as they evolve over time. The equipment might be containers which hold freight (ocean containers, truck trailers, boxcars), equipment such as locomotives, truck tractors, taxicabs, or business jets (companies in the fractional jet ownership business may own up to 1,000 jets). The equipment has to serve customers (people or freight) who typically want to move from one location to the next. We make the assumption throughout that a piece of equipment can serve one request at a time.

One of the challenges of fleet management is that customer requests arrive randomly over time, often requiring service within a narrow interval. Since it can take from several days to more than a week to move transportation equipment over long distances, it is not possible to wait until a customer request is known before moving the equipment. As a result, it is necessary to move equipment to serve demands before they are known. In actual applications, there are other sources of randomness, such as transit times and equipment failures.

Fleet management problems in practice are quite rich, and it is helpful to focus on a particular application to provide a motivating context. In this chapter we use the classic problem of car distribution in railroads. The problem of optimizing the flows of empty freight cars for railroads is typically formulated as a textbook transportation problem. There are supplies of empty cars and customers placing orders for these cars. Standard practice is to model the cost of an assigment as the time required to get from the location of a particular

*Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544 (powell@princeton.edu).

†Department of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY 14853 (huseyin@orie.cornell.edu).

car supply to a customer demand. Once we have the supplies, demands, and costs, we just throw it into our favorite LP package or network solver and we are done!

This is fairly close to what is being done in engineering practice today. At least two major railroads in North America are solving their car distribution problem with this basic model. The advantage of the model is that it is easy to understand, the data requirements are straightforward, and it is easy to solve. The question is, does it work?

The answer is, sort of. Any model is an approximation, and the car distribution models used in practice today involve many approximations, but they can still add considerable value compared to manual fleet management. Of particular interest in this chapter, models today either ignore customer demands in the future or resort to deterministic forecasts. By contrast, interviews with planners reveal different strategies to handle the uncertainties of rail operations.

Actual rail networks are complex, noisy operations, and the management of cars has to be done in a way that recognizes all the sources of uncertainty. Surprisingly, the demands of customers for rail cars is not generally the most significant source of noise, primarily because they are in the habit of booking their orders well in advance. Furthermore, many of the largest customers are very predictable (although some are notoriously unpredictable). Perhaps the most annoying source of uncertainty for rail is the transit times, which might range between two and eight days between a pair of cities.

Another source of uncertainty is the supply of new empty cars. Cars may be sent "off line" to another railroad. The other railroad may then return the car empty but with virtually no advance notice. As a result, empty cars will arrive from another railroad in a highly unpredictable manner. There is also the problem of cars breaking down or being judged unacceptable for service by the customer (typically because they are not clean enough).

Car distribution managers learn to deal with all this uncertainty by maintaining extra inventories of cars, allowing them to quickly substitute alternative cars when events do not proceed as planned. But how many extra cars should be provided, and what type? And where should they be stored? And how do we decide when there are simply too many cars and there is an opportunity to reduce the fleet? Furthermore, there are some periods of the year when demands are higher than normal, which means that inventories will have to be tightened.

In addition to the challenge of planning inventories, deterministic models have an annoying property that can produce serious practical problems. When a car is emptied, it is preferred that it be immediately moved to the area where it will eventually be needed. If not, it has to be brought into a local yard, stored, and then finally moved in response to a customer demand (typically a fairly long distance). It is much better to move the car right away to the general area where it will be needed, and then move the car a much shorter distance when the customer demand actually arises. Deterministic models can use a forecast of future demands, but what happens when the total supply of cars is greater than the point forecast of future car orders? A deterministic model is not able to move the car to a location where it *might* be needed.

Current advances in empty car distribution focus on improving the model of the physical process: the movement of trains, train capacities, yards, travel times, and times through yards. But these models completely ignore the modeling of the evolution of information. Stochastic programming provides a framework for modeling the evolution of information much more accurately than is done with current technologies.

12.2 The car distribution problem

We begin with a general description of the car distribution problem. Section 12.2.1 describes the physical processes that govern car distribution, and section 12.2.2 describes the information processes.

12.2.1 The physical process

Rail operations can be hopelessly complex, but for the purpose of our model, we are going to represent them in a very simple way. We start when a car becomes empty and available at a shipper's location. The shipper notifies the railroad, which then schedules a local train to come out and pick up the available car and bring it back to a regional yard. It is easiest for the railroad if the car already has an assignment to a final destination, since this allows the railroad to put the car on the right tracks at the regional yard so that it can move on the correct outbound train.

Cars are moved in blocks of cars that share a common origin and destination. However, it is not always possible to build a block of cars that goes all the way to their final destination. Instead, it may be necessary to build blocks of cars to intermediate locations called *classification yards*. Here, blocks can be broken up and divided on multiple outbound tracks, each representing a new block of cars with a common destination. Needless to say, breaking up a block of cars at a classification yard takes time and adds a considerable amount of scheduling uncertainty. On the other hand, it offers an important opportunity: the ability to make a decision.

It can take a week or more to move a car from origin to destination. Needless to say, a lot of information is arriving during this time, and it may be the case that the railroad would like to make a new decision about the car. This can happen only at classification yards. After moving through one or more classification yards, the car finally arrives at a regional depot, from which the car may be delivered to a number of customers in the region.

The network is depicted in Figure 12.1. There are four types of decision points: the customer location when the car first becomes empty, the regional depot where the car is first placed when pulled from the customer, intermediate classification yards, and the regional depot at the destination. Once a car is placed at a destination regional depot, it should not be moved again to another regional depot. However, a car arriving at a classification yard can, in principle, be sent to any other classification yard or any regional depot.

One of the most powerful strategies used by railroads to handle uncertainty is substitution. Substitution occurs across three primary dimensions:

1. *Geographic substitution.* Cars in different locations may be used to satisfy a particular order. The ability to use choose among cars at different points in space is referred to as geographical substitution.

2. *Temporal substitution.* The railroad may provide a car that arrives on a different day.

3. *Car type substitution.* The railroad may try to satisfy the order using a slightly different car type.

Rail cars come in a variety of types. There are about 30 car groups, which include cars such as box cars, gondolas, coal cars, tanker cars, and so on. Within a group, there could

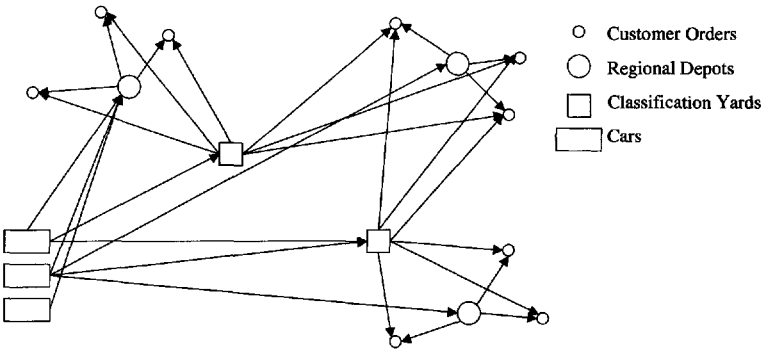


Figure 12.1. *Car distribution through classification yards.*

be anywhere from five or six to as many as two dozen specific types of cars. As a general rule, a customer will require a car from a particular group and may even require a car of a particular type within a group. Substitution across groups is typically not possible.

Temporal substitution, which also enables geographic substitution, is governed in large part by the characteristics of the customer order. Some car orders need to be satisfied on a particular day, while many can be satisfied any day within a week. Customers that offer within-week flexibility introduce other constraints. A customer may need 25 cars over the course of the week but is unable to take more than 10 cars on any given day and needs at least 2 or 3 cars each day to keep the warehouse busy.

12.2.2 The informational process

Our car distribution problem evolves as a result of flows of exogenous information processes and decisions (which might be thought of as endogenous information processes). There are five classes of exogenous information processes:

1. *Car orders.* Customers call in orders for cars, typically the week before they are needed. The car order does not include the destination of the order.

2. *Order destination.* The destination of the order is not revealed until after the car is loaded.

3. *Empty cars.* Empty cars become available from four potential sources: cars being emptied by shippers, empty cars coming on-line from other railroads, cars that have just been cleaned or repaired, and new cars that have just been purchased or leased.

4. *Transit times.* As a car progresses through the network, we learn the time required for specific steps (after they are completed).

5. *Updates to the status of a car.* Cars break down ("bad order" in the language of railroads) or are judged (typically by the customer) to be not clean enough.

The flows of cars are, of course, affected by the decisions that are made.

1. *Move car to a location.* An empty car may be moved to a regional depot or an intermediate classification yard.

2. *Assign to a customer order.* Here we make the explicit assignment of a car to a specific customer order.

3. *Clean or repair a car.* This produces a change in the status of the car.

4. *Switch pools.* Many cars belong to shipper pools which may be adjusted from time to time.

5. *Buy/sell/lease decisions.* These decisions affect the overall size of the fleet.

Our presentation will consider only the first two classes, although classes 3 and 4 represent trivial extensions.

It is useful to look at some actual data streams to gain an appreciation of the amount of demand variability that railroads have to deal with. Figure 12.2 shows an actual set of demands for a particular type of car at a single location. Also shown is a smoothed estimate of the demands (a point forecast) as well as 10th and 90th percentiles. This figure demonstrates the tremendous amount of noise that railroads have to accommodate. What is not shown is the timing of this information. Car demands are typically known the week before they have to be filled. By contrast, empty cars are known only as they become available. From this perspective, the noise in the empty car supply information is more difficult to deal with.

One of the more problematic issues for railroads (and their customers) is the uncertainty in the transit times. Data from a railroad demonstrated that a transit time estimated

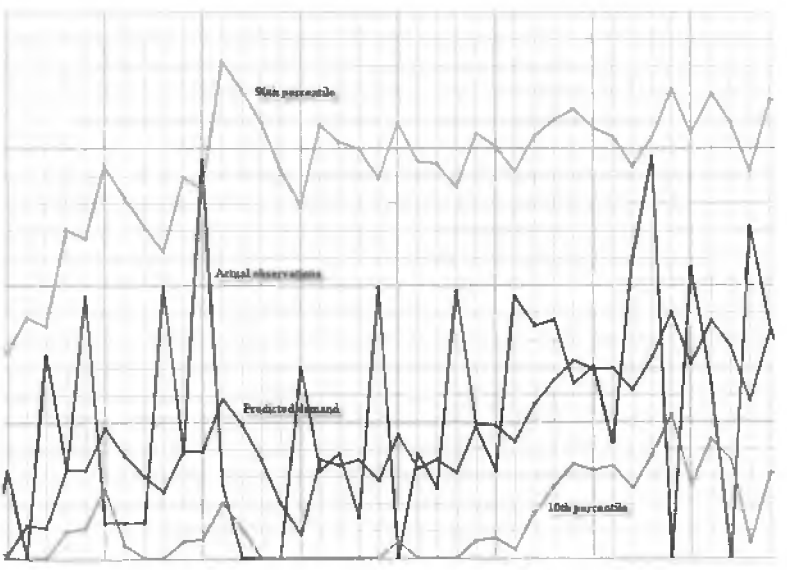


Figure 12.2. *Actual versus predicted forecasts of future demands, showing the 10th and 90th percentiles.*

at 6 days can easily range from 4 to 10 days. Shorter times can arise when additional trains are scheduled that provide faster than normal service. Longer times reflect delays in classification yards when cars are held for various reasons.

12.3 A car distribution model

We now translate our physical problem into a mathematical model. Section 12.3.1 provides our notation. Section 12.3.2 provides a simple model that is commonly used by railroads today.

12.3.1 Notation

Our notation is organized along the following elements: the network, the resources being managed, the decisions being made, exogenous information processes, system dynamics, and the objective function.

The network

We assume that cars move between three types of locations:

$\mathcal{I}^c$ = set of locations representing customers,

$\mathcal{I}^{rd}$ = set of locations representing regional depots,

$\mathcal{I}^{cl}$ = set of locations representing classification yards.

A car that has just become available at a customer can be assigned to any location, as can a car at a classification yard. Cars at regional depots fall in two categories: those that have been pulled in by a local train from a customer that just emptied the car, and those that have been moved empty to the depot with the idea of serving a customer in that region. Cars in the second group are supposed to be assigned only to customers in that region. As a result, the decision to move a car empty to a regional depot should be planned with the expectation that the car cannot later be moved to yet another regional depot (or to a customer not served by the first regional depot).

The resources classes

It is customary to model car distribution problems as multicommodity flow problems. This implies characterizing a car by a state (typically its location) and a class (the car type). Our work with real problems has shown that real cars, in practice, are characterized by a more complex set of attributes. For this reason, it is easier to characterize a car by a vector of attributes. Let

a = vector of attributes characterizing a car;

$\mathcal{A}$ = set of attributes of the cars;

$R^c_{t,at'}$ = number of cars with attribute a that we know about at time t that will be available at time t'; the attribute vector includes the location of the car (at time t') as well as its characteristics;

$R^c_{t,t'} = (R^c_{t,at'})_{a \in \mathcal{A}}$;

$R^c_t = (R^c_{t,t'})_{t' \in \mathcal{T}}$.

It is sometimes convenient to view the time t' at which a car will be available as simply another attribute. In this case, instead of writing $R^c_{t,at'}$ as the number of cars that we know about at time t that will be available at time t' with attribute a, we write R^c_{ta} as the number of cars that we know about at time t with attribute a, where one of the elements of a is the time at which the car will be available.

Although it is not conventional to think of a customer order as a "resource," mathematically, they are modeled the same way. We let

b = vector of attributes characterizing an order;

$\mathcal{B}$ = set of attributes of an order, including the number of days into the future on which the order should be served (in our vocabulary, its actionable time);

$R^o_{t,bt'}$ = vector of car orders with attribute $b \in \mathcal{B}$ that we know about at time t which are needed at time t'; $R^o_{0,bt'}$ is the set of orders that we know about now;

$R^o_{t,t'} = (R^o_{t,bt'})_{b \in \mathcal{B}}$;

$R^o_t = (R^o_{t,t'})_{t' \in \mathcal{T}}$.

As with cars, we may fold the time t' at which an order is actually available to be served into the attribute vector b.

The resource vector $R_t = (R^c_t, R^o_t)$ captures what is known at time t. The vector $R_{tt'} = (R^c_{tt'}, R^o_{tt'})$ captures what is known at time t and actionable at time t'. The difference between knowability and actionability is often overlooked in modeling. Deterministic models assume that everything is known now (typically $t = 0$). Stochastic models typically assume that resources are actionable as soon as they are knowable. For the car distribution problem, the difference between knowable and actionable times is a major modeling element which cannot be overlooked.

Decisions

The decision classes are as follows:

$\mathcal{D}^c$ = The decision class to send cars to specific customers, where $\mathcal{D}^c$ consists of the set of customers (each element of $\mathcal{D}^c$ corresponds to a location in $\mathcal{I}^c$).

$\mathcal{D}^o_t$ = The decision to assign a car to a specific order. In this case, $\mathcal{D}^o_t$ can be either the set of all orders available at time t (every element of $\mathcal{D}^o_t$ is a specific order) or the set of all order types (basically, $\mathcal{B}$). We use the latter interpretation, which also means that we can drop the index t (since the set of order types is static).

$\mathcal{D}^{rd}$ = The decision to send a car to a regional depot (the set $\mathcal{D}^{rd}$ is the set of regional depots—we think of an element of $\mathcal{I}^{rd}$ as a regional depot, while an element of $\mathcal{D}^{rd}$ as a decision to go to a regional depot).

$\mathcal{D}^{cl}$ = The decision to send a car to a classification yard (each element of $\mathcal{D}^{cl}$ is a classification yard).

d^{ϕ} = The decision to hold the car ("do nothing").

When we assign a car to an order, this means we are choosing a decision $d \in \mathcal{D}^o$ to assign a car to a particular type of order, which we designate using

$b_d \in \mathcal{B}$ = attributes of the car type associated with decision d.

Our complete set of decisions, then, is $\mathcal{D} = \mathcal{D}^c \cup \mathcal{D}^o \cup \mathcal{D}^{rd} \cup \mathcal{D}^{cl} \cup d^{\phi}$. We assume that we act only on cars (for example, in some industries it is possible to contract out a customer request, but we do not have this option). Of these, decisions in $\mathcal{D}^o$ are constrained by the number of orders that are actually available at time t (this constraint is captured in the resource vector R_t^o). Often, the set of decisions depends on the attributes of the resource being acted on. Let

$\mathcal{D}_a$ = set of decisions that can be used to act on a resource with attribute a.

Our decisions are represented using

x_{tad} = number of times that we act on a car with attribute a using decision d at time t,

$x_t = (x_{tad})_{a \in \mathcal{A}, d \in \mathcal{D}}$,

= vector of decisions at time t.

Our notation has been chosen to capture the informational dynamics of the problem. For example, standard notation in transportation is to let x_{ijt} be the flow from location i to location j departing at time t. The index j effectively presumes a deterministic outcome of the decision. (The notation $x_{ijt}(\omega)$ does not fix the problem; we would have to write $x_{i,j(\omega),t}$, which is quite ugly.) Our decision variable is indexed by what is known when a decision is made.

Our sequencing of the subscripts t, a, and d is also chosen for a reason. If x_{tad} is a particular decision, it is common to sum over all the decisions d that might act on a resource with attribute a. We then have to sum over all the resources with attribute a at time t. Finally, we might sum the activities over all time periods. Thus, there is a natural hierarchy among the indices t, a, and d which we reflect in how we have ordered them.

For the moment, we do not address the question of how a decision is made, but we represent our decision function using

$X^{\pi}(R_t)$ = decision function which returns a decision vector $x_t \in \mathcal{X}_t$, where $\mathcal{X}_t$ is our feasible region and R_t is our general resource vector at time t. We assume that there is a family of decision functions $(X^{\pi})_{\pi \in \Pi}$ from which we have to choose.

Exogenous information processes

Our system is driven by a series of exogenous information processes. These can be divided between two broad classes: updates to the "resources" being managed (where we include both cars and car orders) and updates to the parameters that drive our system (the modify function). We model updates to the resources using

$\hat{R}^o_{tt'}$ = vector of new car orders arriving in time period t that become actionable at time $t' \geq t$,

$\hat{R}^o_t = (\hat{R}^o_{tt'})_{t' \geq t}$.

We would define similar vectors $\hat{R}^c_{tt'}$ and $\hat{R}^c_t$ for new cars becoming empty.

The second important information process for our model is the transit times. The simplest model is to assume that transit times become known immediately after a decision is made. We might define, then,

$\hat{\tau}_{tad}$ = actual transit time resulting from a decision d acting on a car with attribute a at time t,

$\hat{\tau}_t = (\hat{\tau}_{tad})_{a \in \mathcal{A}, d \in \mathcal{D}}$.

A final source of exogenous information arises when the expected state of the car at the destination, which we have represented by the terminal attribute function $a^M(t, a, d)$, is different from what we expected. We can model this by letting $\hat{a}^M_{tad}$ be the random variable which determines the final attribute of a car after it is acted on. However, this degree of realism is beyond the scope of our presentation.

Let W_t be our general variable representing the information arriving at time t. Then

$$W_t = (\hat{R}^o_t, \hat{R}^c_t, \hat{\tau}_t)$$

is our exogenous information process. Let ω be a sample realization of $(W_t)_{t \in \mathcal{T}}$, and let Ω be the set of potential outcomes. Further, let $\mathcal{F}$ be the σ-algebra generated by $(W_t)_{t \in \mathcal{T}}$, and let $\mathcal{F}_t$ be the sub-σ-algebra generated by $(W_{t'})^t_{t=0}$, where $\mathcal{F}_t$ forms a filtration. If $\mathcal{P}$ is a probability measure on Ω, then we can let $(\Omega, \mathcal{F}, \mathcal{P})$ be our probability space.

Throughout this chapter, we use the index t to represent the information concept of a variable or function. Thus, a variable R_t or a function Q_t is assumed to be $\mathcal{F}_t$-measurable.

System dynamics

We assume that the effects of a decision are captured by the modify function, which can be represented using

$$M(t, a, d) \rightarrow (a', c, \tau), \tag{12.1}$$

where d is a decision acting on a car with attribute a at time t, producing a car with attribute a', generating a contribution c, and requiring time τ to complete the action. a', c, and τ are all functions, which we can represent using

$$(a^M(t, a, d), c^M(t, a, d), \tau^M(t, a, d)).$$

We call $a^M(t, a, d)$ the terminal attribute function. Normally, we represent the costs and times using the vectors $c_{tad} = c^M(t, a, d)$ and $\tau_{tad} = \tau^M(t, a, d)$. It is not generally the case that $(a^M(t, a, d), c^M(t, a, d), \tau^M(t, a, d))$ are deterministic functions of (t, a, d). This is particularly true of the transit time $\tau^M(t, a, d)$.

For algebraic purposes it is useful to define

$$\delta_{t',a'}(t, a, d) = \text{change in the system at time } t' \text{ given a decision executed at time } t$$
$$= \begin{cases} 1 & \text{if } M_t(t, a, d) = (a', \cdot, t' - t), \\ 0 & \text{otherwise.} \end{cases}$$

We now need to model the dynamics of our resource vector. At this point, it is useful to define two points at which we can measure our inventories of cars and orders: immediately before decisions are made and immediately after. We let R_t be the resource state vector immediately before decisions are made (the predecision state vector) but after new cars and orders have already been added in. We let R_t^x be the resource state vector immediately after decisions are made (this can also be thought of as the resources available at the beginning of time $t + 1$). The postdecision state vector is given by

$$R^x_{t,a't'}(\omega) = R_{t,a't'} + \sum_{d\in\mathcal{D}}\sum_{a\in\mathcal{A}} \delta_{t',a'}(t, a, d)x_{tad}, \quad a' \in \mathcal{A}, \quad t' > t. \tag{12.2}$$

The predecision state variable is given by

$$R_{t+1,a't'}(\omega) = R^x_{t,a't'}(\omega) + \hat{R}_{t+1,a't'}(\omega), \quad a' \in \mathcal{A}, \quad t' > t. \tag{12.3}$$

The postdecision state variable, which is implicit in many stochastic programming applications, is useful because it is a deterministic function of x_t, whereas the predecision state variable R_{t+1} is a stochastic function of x_t.

12.3.2 A simple car distribution model

We can quickly illustrate the basic deterministic car distribution models that are used in practice. A myopic model would be given by

$$\max_x \sum_{a\in\mathcal{A}}\sum_{d\in\mathcal{D}} c_{0ad}x_{0ad} \tag{12.4}$$

subject to

$$\sum_{d\in\mathcal{D}} x_{0ad} = R^c_{0a}, \quad a \in \mathcal{A}, \tag{12.5}$$

$$\sum_{a\in\mathcal{A}} x_{0ad} \leq R^o_{0b_d}, \quad d \in \mathcal{D}^o, \tag{12.6}$$

$$x_{0ad} \in Z_+. \tag{12.7}$$

Keep in mind that for every element d in the set of order decisions, $\mathcal{D}^o$ corresponds to an order type with attribute b_d.

A model that incorporates point forecasts of future car orders is a minor variation. Let $\bar{R}^o_{tb}$ be a point forecast of the car orders that are made (that is, become known) at time t of type b. A deterministic model that incorporates this forecast is given by

$$\max_x \sum_{a\in\mathcal{A}} \sum_{d\in\mathcal{D}} c_{0ad} x_{0ad} \tag{12.8}$$

subject to

$$\sum_{d\in\mathcal{D}} x_{0ad} = R^c_{0a}, \quad a \in \mathcal{A}, \tag{12.9}$$

$$\sum_{a\in\mathcal{A}} x_{0ad} \leq R^o_{ob_d} + \sum_{t\in\mathcal{T}^{ph}\setminus 0} \bar{R}^o_{tb_d}, \quad d \in \mathcal{D}^o, \tag{12.10}$$

$$x_{0ad} \in Z_+. \tag{12.11}$$

Equation (12.10) includes demands that are known now (R^o_{0b}) and all orders that are forecast to become known within the planning horizon. We have not included point forecasts of new empty cars becoming available. While this is easy to do, it is not common practice. As a result, this model has a bias wherein it includes future orders but not future cars.

These simple models are attractive in part because they are easy to understand, easy to formulate and solve, and appear to capture the way railroads actually make decisions. For example, it is common to assign cars available now to known (and sometimes even forecast) orders, ignoring cars that may become available in the future. Conversations with actual planners, however, reveal that they will look in the aggregate at total orders that are expected to arise, and total cars that are expected to become available, to determine whether they seem to be in a surplus or deficit situation. Often, a car may become available for which there is not a pending order. Rail operations work best if the car is assigned to a destination as soon as it becomes empty, forcing planners to move cars to locations where there appears to be a good likelihood of their being used.

The following sections introduce stochastic formulations of this model, beginning in section 12.4 with a two-stage stochastic program, followed by section 12.5, which reviews algorithms for two-stage problems. Section 12.6 presents a multistage stochastic programming formulation, which can be solved as sequences of two-stage problems.

12.4 A two-stage model

The management of rail cars over time is a multistage resource allocation problem. But it is not unreasonable to formulate it as a two-stage problem. The justification arises because of the nature of the car distribution problem. In any given week, a railroad has to decide how to move cars that week to meet demands that will arise in the next week. Transit times are quite large, and it will generally take several days to a week or more to move a car. For this reason, decisions to assign a car to an order within the same week are typically restricted to cars that are already at the regional depot and which now just have to be pulled by a local train from the regional depot to a customer within the region.

Reflecting these long transit times, it is common industry practice for customers to place an order in one week that needs to be met the following week. This means that by

Friday morning of a week, more than 90% of the orders for the next week are known. However, most car orders are made on Wednesday and Thursday. This means that decisions to move cars early in the week still have to be made without much of the information that will become available the following week. As a result, it is often necessary to make a decision to move a car on Monday or Tuesday, where the actual demand will become known only after the decision is made.

Once a car has been assigned to an order, the transit time typically takes the car past the end of a 2-week horizon. We are typically not concerned with where the car is ultimately going, since our problem right now is simply to cover the order. Thus, all the activity (including all the costs and revenues) that we are concerned with right now occurs within the next two weeks, during which time a car will, with rare exception, be able to cover at most one demand. For this reason, a two-stage formulation is a reasonable starting point (and helps to lay the foundation for a multistage model).

$$\max_{x_0} \left\{ c_0 x_0 + \bar{Q}(R_0^{c,x}) \right\} \tag{12.12}$$

subject to

$$\sum_{d \in \mathcal{D}} x_{0ad} = R_{0a}^{c}, \quad a \in \mathcal{A}, \tag{12.13}$$

$$\sum_{a \in \mathcal{A}} \sum_{d \in \mathcal{D}_a} x_{0ad} \delta_{1a'}(0, a, d) - R_{0a'}^{c,x} = 0, \quad a' \in \mathcal{A}, \tag{12.14}$$

$$R_{0a'}^{c,x} + \hat{R}_{1a'}^{c} - R_{1a'}^{c} = 0, \quad a' \in \mathcal{A}, \tag{12.15}$$

$$\sum_{a \in \mathcal{A}} x_{0ad} \leq R_{0b_d}^{o}, \quad d \in \mathcal{D}^o, \tag{12.16}$$

$$x_{0ad} \in Z_+. \tag{12.17}$$

$R_{0a'}^{c,x}$ represents the cars that will have attribute a' (for stage 1) as a result of decisions made in stage 0. The variable is indexed by time 0 because it is a function of the information available in time 0. This notation may seem a bit awkward in this setting, but it becomes more important in the context of multistage problems (and especially when we have to deal with multiperiod travel times).

The second stage problem consists of finding

$$\bar{Q}(R_0^{c,x}) = EQ(R_1^c, R_1^o). \tag{12.18}$$

The conditional second stage function is given by

$$Q(R_1^c(\omega), R_1^o(\omega)) = \max_{x_1(\omega)} \sum_{a \in \mathcal{A}} \sum_{d \in \mathcal{D}_a^o} c_{1ad} x_{1ad}(\omega) \tag{12.19}$$

subject to

$$\sum_{d \in \mathcal{D}_t^c} x_{1ad}(\omega) + x_{1d^{\phi}a}(\omega) = R_{1a}^{c}(\omega) \quad \forall a \in \mathcal{A}, \tag{12.20}$$

$$\sum_{a \in \mathcal{A}} x_{1ad}(\omega) \leq R_{1b_d}^{o}(\omega) \quad \forall d \in \mathcal{D}_a, \ a \in \mathcal{A}, \tag{12.21}$$

$$x_{1ad}(\omega) \geq 0 \quad \forall d \in \mathcal{D}_a, \ a \in \mathcal{A}. \tag{12.22}$$

This is a classical two-stage stochastic program with network recourse, with one twist. In practical problems, the space $\mathcal{A}$ may be too large to enumerate.

12.5 Solution algorithms for two-stage problems

There are a variety of algorithms for two-stage problems with network recourse. Our choice of algorithms has to be guided by several factors. First, we would like integer solutions. Second, the problem may be quite large, especially if the attribute vector a has enough dimensions that the space $\mathcal{A}$ is too large to enumerate. These characteristics effectively eliminate scenario methods (which produces problems that are too large) as well as stochastic linearization and Benders decomposition (which produce highly fractional solutions). What remains are methods that approximate the second-stage recourse function in a way that retains as much of the inherent network structure as possible.

We present two methods that have shown promise for this problem class. Both are techniques for deriving separable, nonlinear approximations of the value function. Since we are interested in integer solutions, we use piecewise linear approximations with breakpoints occurring only for integer values of supplies. Section 12.5.1 describes how to estimate these functions using auxiliary function methods. Section 12.5.2 describes a relatively new class of structured, adaptive function methods that we call SAFE algorithms.

Throughout this section, we assume that we are solving the problem iteratively, where n is the iteration counter. x^n and R^n, then, represent specific numerical values at iteration n, while x and R represent variables.

12.5.1 Auxiliary function methods

We are not able to estimate the function $\bar{Q}(R)$ in (12.18). However, we are able to solve (12.19)–(12.22) fairly easily for a single, sample realization. Let $\tilde{q}_1$ be the vector of dual variables for (12.20). We could use these duals, which are stochastic gradients of $\bar{Q}(R_0^{c,x})$, to create a linear approximation of the value function by first smoothing the gradients

$$\hat{q}^n = (1 - \alpha^n)\hat{q}^{n-1} + \alpha^n \tilde{q}^n.$$

We could then solve

$$x_0^n = \arg\max_{x_0} c_0 x_0 + \hat{q}^n R_0^{c,x}(x_0).$$

This sort of solution procedure, however, would be very unstable and would not converge. It is necessary to introduce the smoothing step of the form

$$\bar{x}^n = (1 - \alpha^n)\bar{x}^{n-1} + \alpha^n x_0^n.$$

Unfortunately, a smoothing step such as this would destroy the integrality of the solution. The better alternative is to use an auxiliary function. The SHAPE algorithm starts with an initial nonlinear approximation $\hat{Q}^0(R)$ which is then updated using the strategy

$$\hat{Q}^n(R) = \hat{Q}^{n-1}(R) + \alpha^n \left(\tilde{q}^n - \nabla \hat{Q}^{n-1}(R^n)\right) R. \tag{12.23}$$

We would then solve sequences of problems of the form

$$x_0^n = \arg\max_{x_0} c_0 x_0 + \hat{Q}^{n-1}(R_0^{c,x}(x_0)). \tag{12.24}$$

We then use $R^n = R_0^{c,x}(x_0)$ in (12.23). The approximation $\hat{Q}^0(R)$ should be concave (since the expected recourse function is also concave). It is also natural to choose separable approximations. Some choices are

$$\hat{Q}^0(R) = \rho_0 \left(1 - e^{-\rho_1 R}\right),$$
$$\hat{Q}^0(R) = \ln(R + 1),$$
$$\hat{Q}^0(R) = -\rho_0 (R - \rho_1)^2.$$

This algorithm is provably convergent if the expected recourse function $\bar{Q}(R_0^{c,x})$ (and the approximation $\hat{Q}^0(R)$) are continuously differentiable. However, it can be applied approximately to nondifferentiable functions. For these problems, we can choose $\hat{Q}^0(R)$ so that it is piecewise linear, concave, and separable. For example, we could use piecewise linear versions of (12.23). When this choice of function is made, (12.24) involves solving sequences of networks that are depicted in Figure 12.3. In this figure, the first set of arcs (moving left to right) represent assignments of cars to orders. The second set of parallel arcs represent the slopes of the recourse function at each node, where a node represents a geographical location. These are network subproblems and naturally produce integer solutions (assuming that the piecewise linear approximations have breaks at integer values of the supplies).

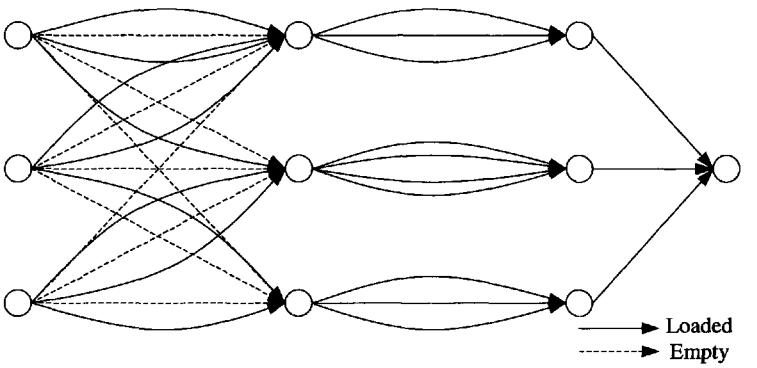


Figure 12.3. *Network structure of a one-period single commodity problem.*

The SHAPE algorithm yields sequences of computationally tractable approximations that are very easy to solve and that also naturally produce integer solutions. However, the shape of the recourse function is fixed by the choice of the initial approximation (such as (12.23)). The updating equation will tilt these functions but does not change their basic shape. This means that the quality of the solution produced by the algorithm may be fairly dependent on the choice of the parameters that characterize $\hat{Q}^0(R)$. The next section presents an algorithm that overcomes this limitation.

12.5.2 Structured, adaptive function estimators

Structured, adaptive function estimators, or "SAFE" algorithms, produce piecewise linear, concave, separable approximations of the recourse function by using sample gradients to update portions of the recourse function. No assumption is made about the shape of the recourse function, other than that it is piecewise linear and concave. Furthermore, it is necessary to maintain concavity at every iteration. SAFE algorithms still involve solving sequences of problems using the strategy given in (12.24) and produce problems with the same mathematical structure. The only difference is the calculation of the approximation $\hat{Q}^n(R)$.

SAFE algorithms use stochastic gradients to update estimates of a piecewise linear function while maintaining concavity after each update. There are several schemes for doing this, although here we describe a simple projection method that appears to work very well and also has proven convergence properties.

Let $Q(r), r \in \Re$, be a piecewise linear, concave function, where r is a scalar argument, and let

$$q_r = Q(r+1) - Q(r)$$

be the right derivative of $Q(R)$ at $R = r$. We can write $Q(r)$ in terms of its derivatives using

$$Q(R) = Q(0) + \sum_{r=0}^{R-1} q_r.$$

For our problems, we can always let $Q(0) = 0$. Let $\hat{q}_r^n$ be an estimate of q_r at iteration n, giving us the functional approximation

$$\hat{Q}^n(R) = \sum_{r=0}^{R-1} \hat{q}_r^n.$$

Let $\tilde{q}^n$ be a stochastic gradient of Q at iteration n, and assume the gradients have the property that $E[\tilde{q}_r^n] = q_r$. Assume that at iteration n we sample $r = R^n(\omega)$. If $\bar{q}_r^n$ is our estimate of the slope for $R = r$, then we can estimate the slopes using the simple updating equation:

$$\bar{q}_r^n = \begin{cases} (1-\alpha^n)\bar{q}^{n-1} + \alpha^n \tilde{q}^n & \text{if } r = R^n(\omega), \\ \bar{q}_r^{n-1} & \text{otherwise.} \end{cases} \tag{12.25}$$

If we assume that we are going to sample all the slopes infinitely often, then it is not hard to show that $\lim_{n\to\infty} \hat{q}_r^n = q_r$. But this updating scheme would not work in practice since it does not maintain the concavity of the function $\hat{Q}^n(R)$. We know from the concavity of $Q(R)$ that $q_0 \geq q_1 \geq \cdots \geq q_r$. Equation (12.25) would not maintain this relationship between the slopes. A loss of concavity at an intermediate iteration would make it very difficult to solve (12.24) (and obtain proper dual variables). Concavity is automatically maintained in (12.23) since we are updating a concave approximation with a linear updating term.

We can maintain concavity by using a simple projection algorithm,

$$\hat{q}^n = \Pi_{\mathcal{Q}}(\tilde{q}^n). \tag{12.26}$$

The projection $\Pi_{\mathcal{Q}}$ is the solution to the quadratic programming problem

$$\max_{\hat{q}^n} \|\hat{q}^n - \tilde{q}^n\|^2 \tag{12.27}$$

subject to

$$\hat{q}^n_{r+1} - \hat{q}^n_r \leq 0. \tag{12.28}$$

The projection is easily solved. Assume that after the update (12.25), we have an instance where $\tilde{q}^n_{r-1} < \tilde{q}^n_r$. Let $\bar{r} = \text{argmin}_{r'<r}\{\tilde{q}^n_{r'} < \tilde{q}^n_r\}$ be the smallest index such that $\tilde{q}^n_{\bar{r}} < \tilde{q}^n_r$. Our projection is found by finding the average over all these elements:

$$\bar{q}^n_{[\bar{r},r]} = \frac{1}{r - \bar{r} + 1} \sum_{r'=\bar{r}}^{r} \bar{q}^n_{r'}.$$

Finally, we let

$$\hat{q}^n_{r'} = \begin{cases} \bar{q}^n_{[\bar{r},r]} & \text{if } \bar{r} \geq r' \geq r, \\ \tilde{q}^n_r & \text{otherwise.} \end{cases}$$

12.5.3 Extension to large attribute spaces

A special challenge in resource allocation problems arises when the attribute vector a has multiple dimensions (even as few as five or six). In these cases, it may not be possible to enumerate the attribute space $\mathcal{A}$. This means that we may need an estimate of the value of a resource with attribute a' but we do not have an approximation $\hat{Q}_{a'}$. An effective strategy can be to produce approximations $\hat{Q}^{(n)}_{a^{(n)}}$, where $a^{(n)}$ is the nth aggregation of the attribute vector a. If $n = 0$ represents the most disaggregate level, then $\hat{Q}^{(n)}_{a^{(n)}}$ for increasing values of n will produce lower statistical error and higher structural error.

12.5.4 Experimental work for two-stage problems

Two-stage stochastic programs in general, and those with network recourse in particular, have been widely studied. Lacking special structure, the most widely cited approach for solving two-stage stochastic programs is based either on scenario methods (which are computationally intractable for our problems) or Benders decomposition. Benders decomposition comes in different flavors, including L-shaped decomposition, stochastic decomposition, and the recently proposed CUPPS algorithm. L-shaped decomposition assumes a finite, and relatively small, sample of scenarios which are used to generate cuts. Stochastic decomposition is an iterative algorithm that effectively assumes an infinite set of scenarios and provides almost sure convergence in the limit. CUPPS requires a finite set of scenarios (there is a particular operation that loops over all scenarios), but the operation is very simple

and the method can handle thousands of scenarios without difficulty. Although CUPPS requires somewhat more work per iteration, it has faster convergence, and hence we use it for our comparisons.

Figure 12.4 shows the relative performance of SAFE and CUPPS for different numbers of training iterations. This graph suggests generally better solution quality for SAFE over smaller number of iterations. The evaluation was based on a testing sample of 50. With 950 training iterations, the algorithms appear to be equivalent.

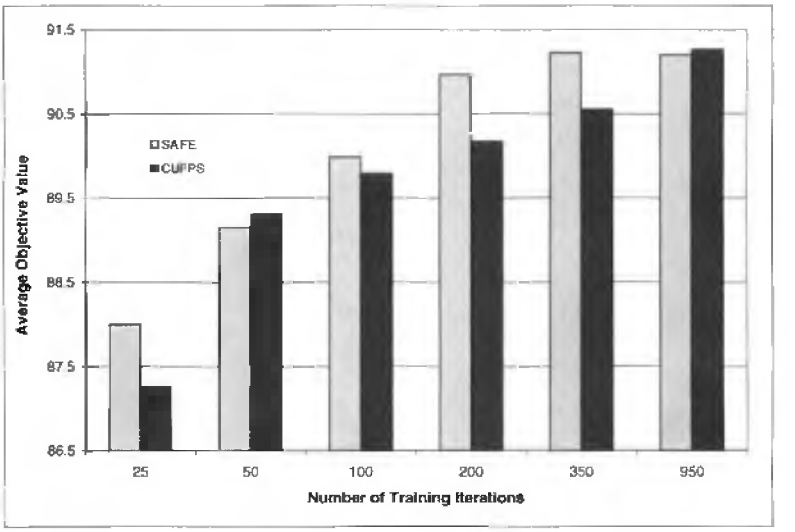


Figure 12.4. *The effect of the number of training iterations on SAFE and CUPPS, illustrating the faster convergence for SAFE.*

Figure 12.5 compares SAFE and CUPPS on problems with 20, 30, 40, and 90 locations, using 200 training iterations. For the smaller problems, the algorithms appear to be virtually the same, while for the 40- and 90-location problems, SAFE is providing better solutions. The chart shows the performance for each of the 50 testing samples. For the smaller data sets, the two algorithms appear to be providing the exact same solution for all 50 samples. For the 90-location data set, SAFE is providing a slightly better answer on all 50 samples. This behavior is somewhat surprising, in part because CUPPS is a provably convergent algorithm, and also because we would expect some noise around the behavior of each method.

12.6 Multistage resource allocation problems

Practical operational models for car distribution can be formulated very approximately as deterministic, single-stage models and quite credibly as two-stage stochastic programs. But there are other planning applications that require looking farther into the future, and for these we need a multistage model. Our approach is to solve the multistage problem as a sequence of two-stage stochastic programs using a dynamic programming framework.

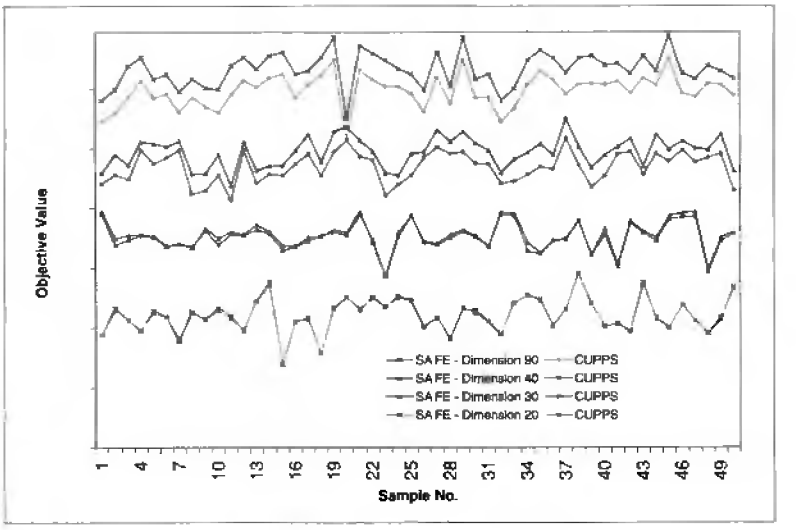


Figure 12.5. *SAFE outperforms CUPPS as the problem size grows for every outcome in the testing data set.*

12.6.1 Formulation

Let

$\hat{R}_t =$ vector of new arrivals of cars and orders in period t, where $\hat{R}_t = (\hat{R}_t^o, \hat{R}_t^c)$;

$W_t =$ complete vector of new information arriving in period t, including $\hat{R}_t$ as well as other information about system parameters (travel times, costs, and parameters governing the physics of the problem).

We model the state of our system using

$R_t =$ vector of resources available in period t after new arrivals are added in,

$K_t =$ what is known at time t after the new information W_t has been incorporated. K_t includes R_t plus what we know about parameters that govern the dynamics of the system.

Our process is controlled by the decisions we make:

$x_t =$ decisions which are made after new information in period t has become known,

$X_t^\pi(R_t) =$ decision function which returns a feasible decision vector x_t given the information available at time t (contained in K_t).

Our decisions are chosen to maximize the expected total contribution over a planning horizon. Our contribution function is expressed as

$c_t(x_t) =$ contribution generated in period t given decision x_t, and what is known, K_t.

The dynamics of the resource state vector for cars are given by (12.3). This equation can be represented more compactly in matrix form using

$$R_{t+1,t'} = R_{tt'} + A_{tt'}x_t + \hat{R}_{t+1,t'}.$$

The informational dynamics can be written generally as

$$K_t = U^K(K_{t-1}, W_t).$$

We can now state our optimization problem using

$$\max_{\pi \in \Pi} E\left\{\sum_{t \in \mathcal{T}} c_t(X_t^\pi(R_t))\right\}.$$

12.6.2 Our algorithmic strategy

Our solution strategy starts with the standard Bellman equation:

$$Q_t(R_t) = \max_{x_t \in \mathcal{X}_t} c_t x_t + E\left\{Q_{t+1}(R_{t+1}(x_t))|R_t\right\}. \tag{12.29}$$

We do not know the recourse function exactly, so we replace the expectation on the right-hand side with an approximation $\bar{Q}_t(R_{t+1})$. Assume that we choose a linear approximation:

$$\begin{aligned}\bar{Q}_t(R_{t+1}) &= \bar{q}_t \cdot R_{t+1} \\ &= \bar{q}_t \cdot (R_t + A_t x_t + \hat{R}_{t+1}).\end{aligned} \tag{12.30}$$

Replacing $E\{Q_{t+1}(R_{t+1})|R_t\}$ with $\bar{Q}_t(R_{t+1})$ gives

$$\begin{aligned}\bar{Q}_t(R_{t+1}) &= E\{\bar{q}_t \cdot (R_t + A_t x_t + \hat{R}_{t+1})|R_t\} \\ &= \bar{q}_t R_t + \bar{q}_t A_t x_t + E\{\bar{q}_t \hat{R}_{t+1}|R_t\}.\end{aligned} \tag{12.31}$$

The expectation on the right-hand side of (12.31) is not a function of x_t and can be dropped. Our decision function can now be written as

$$X_t^\pi(R_t) = \arg\max c_t x_t + \bar{q}_t A_t x_t.$$

Linear approximations are always the easiest to work with. At the same time, they tend to be unstable in practice and typically do not produce good solutions. Convergent algorithms require smoothing on the decision variables, but this will create fractional solutions.

A better approach is to use nonlinear approximations, but this prevents the convenient decomposition that we obtain in (12.31). For this reason, we have to start over using the postdecision state variable, which we originally introduced in (12.2). In matrix form, we write the postdecision state variable using

$R_{tt'}^x$ = resource state variable after the decision x_t has been implemented, but before the new arrivals from period $t+1$ have been added in

$= A_{tt'}x_t.$

We refer to $R_t^x = (R_{tt'})_{t' \geq t}$ as the postdecision state variable because it does not contain the new information (that would arrive during time period $t + 1$) that would be needed to make the decision x_{t+1} in the next time period. Our predecision state variable can now be written:

$$R_{t+1,t'} = R_{tt'}^x + \hat{R}_{t+1,t'}.$$

If we formulate our recursion using the postdecision state variable, we would obtain

$$Q_{t-1}^x(R_{t-1}^x) = E\left\{\max_{x_t \in \mathcal{X}_t} c_t x_t + Q_t^x(R_t^x(x_t)) | R_{t-1}^x\right\}. \tag{12.32}$$

The left-hand side of (12.32) is indexed by time period $t - 1$ because the right-hand side, after taking the expectation, is a function of the information up through time $t - 1$.

The expectation in (12.32) is computationally intractable, as it was in (12.29). However, we can take a sample realization and solve

$$\tilde{Q}_t^x(R_{t-1}^x(\omega)) = \max_{x_t \in \mathcal{X}_t(\omega)} c_t x_t + Q_t^x(R_t^x(x_t, \omega)). \tag{12.33}$$

We finally replace $Q_t^x(R_t^x(x_t))$ with an appropriate approximation $\hat{Q}_t(R_t^x)$ so that (12.33) is still computationally tractable:

$$X_t^{\pi,n}(R_t) = \arg\max_{x_t} c_t x_t + \hat{Q}_t^{x,n-1}(R_t^x(x_t)) \tag{12.34}$$

subject to

$$\sum_{d \in \mathcal{D}} x_{tad} = R_{t,at}^c, \quad a \in \mathcal{A}, \tag{12.35}$$

$$\sum_{a \in \mathcal{A}} x_{tad} \leq R_{tb_d}^o, \quad d \in \mathcal{D}^o, \tag{12.36}$$

$$x_{tad} \in Z_+. \tag{12.37}$$

Let $R_t^{x,n} = R_t^{x,n}(X_t^{\pi,n}(R_t))$ be the resource vector after we have made decision $X_t^{\pi,n}(R_t)$. Let $\tilde{q}_t^n$ be the dual variable (at iteration n) of (12.35). The process of updating the recourse function approximation, where we will use the same techniques that we did for the two-stage problem, can be represented simply using

$$\hat{Q}_{t-1}^{x,n} \leftarrow U^Q(\hat{Q}_{t-1}^{x,n-1}, \tilde{q}_t^n, R_t^{x,n}).$$

There are two ways to use this solution approach. The first is a one-pass procedure where the recourse function is updated as we step through time. The steps of this procedure are given in Figure 12.6. This procedure is the easiest to implement but can suffer from slow convergence since information is passed backward one time period at a time at each iteration.

More rapid communication is accomplished using a two-pass procedure, described in Figure 12.7. This procedure can be somewhat harder to implement, since it requires storing information at each time period for computing gradients that are not used until the end of

STEP 0: Initialization:

Initialize $\hat{Q}_t^0$, $t \in \mathcal{T}$.

Set $n = 0$.

Initialize R_0.

STEP 1: Do while $n \leq N$:

Choose $\omega^n \in \Omega$.

STEP 2: Do for $t = 0, 1, \ldots, T-1$:

STEP 2a: Solve (12.34) to obtain $x_t^n = X_t^\pi(R_t^n, \hat{Q}_{t+1}^{n-1})$ and the duals $\tilde{q}_t^n$ of the resource constraint (12.35).

STEP 2b: Update the resource state: $R_{t+1}^n = M_t(R_t^n, x_t^n, W_{t+1}(\omega^n))$.

STEP 2c: Update the value function approximations using $\hat{Q}_t^n \leftarrow U^Q(\hat{Q}_t^{n-1}, \tilde{q}_t^n, R_t^n)$.

STEP 3: Return the policy $X_t^\pi(R_t, \hat{Q}^N)$.

Figure 12.6. *Single-pass version of the adaptive dynamic programming algorithm.*

the forward pass. This is easiest to envision for the case of a single car type, since each subproblem is a pure network. In this case, we would first solve the problem for time period t. Then, we would loop over each location and add one additional car. The change in the flows forms a flow augmenting path from the location where the additional car was added to some destination node (from which recourse function arcs emanate). We do not have to store the flow augmenting path, but we have to store the total cost of the flow augmenting path (excluding the recourse function) and the location in the future that gains a car as a result of the perturbation. This calculation (which is quite fast) has to be done for each location in time period t. After this has been done for all time periods, we can step backward in time to compute the full gradient (which will not directly include any approximate recourse function values).

If we use a two-pass procedure, then our gradients have a nice property. Let

$$F_t^\pi(R_t, \omega^n) = \sum_{t'=t}^{T} c_t X_t^\pi(R_t(\omega)) \tag{12.38}$$

be the total contribution of following policy π from time t onward, given that we start at resource state R_t and follow the sample path ω. Then we have the following result.

Theorem 12.1. *Let $\tilde{q}_t^n = (\tilde{q}_{at}^n)_{a \in \mathcal{A}}$ be the vector of path costs from time t to the end of the horizon, given outcome ω^n and functional approximations $\{\hat{Q}_t^n\}_{t \in \mathcal{T}}$ computed from a*

STEP 0: Initialize $\hat{Q}_t^0$, $t \in \mathcal{T}$.

Set $n = 0$.

Initialize R_0.

STEP 1: Do while $n \leq N$:

Choose $\omega^n \in \Omega$.

STEP 2: Do for $t = 0, 1, \ldots, T-1$:

STEP 2a: Solve (12.34) to obtain $x_t^n = X_t^\pi(R_t^n, \hat{Q}_{t+1}^{n-1})$ and the duals $\tilde{q}_t^n$ of the resource constraint (12.35).

STEP 2b: Compute $R_{t+1}^n = M(R_t^n, x_t^n, W_{t+1}(\omega^n))$.

STEP 3: Do for $t = T-1, T-2, \ldots, 1, 0$:

STEP 3a: Compute marginal value of a resource, $\tilde{q}_t^n$, using $\hat{Q}_{t+1}^n$ and the optimal basis from the forward pass.

STEP 3b: Update the value function approximations, $\hat{Q}_t^n \leftarrow U^Q(\hat{Q}_t^{n-1}, \tilde{q}_t^n, R_t^n)$.

STEP4: Return policy $X_t^\pi(R_t, \hat{Q}^N)$.

Figure 12.7. *Two-pass version of the adaptive dynamic programming algorithm.*

backward pass. Then $\tilde{q}_t$ satisfies

$$F_t^\pi(R_t, \omega^n) - F_t^\pi(R_t', \omega^n) \geq \tilde{q}_t^n \cdot (R_t - R_t').$$

Furthermore, if the basis paths in each period t are flow augmenting paths into the supersink, then $\tilde{q}_t^n$ is a right gradient of $F_t^\pi(R_t, \omega^n)$.

12.6.3 Single-commodity problems

When we solve single-commodity problems in a multistage setting, we are simply solving sequences of two-stage problems (as was illustrated in Figure 12.3), so we can use all our tricks that we learned for the simple two-stage case. There are, of course, a few differences. First, it is best to obtain dual variables from the backward pass of a two-pass algorithm, as illustrated in Figure 12.7. For a two-stage problem, we obtain dual variables directly from the second stage. For multistage problems, we need to calculate these backward through time.

Second, multistage problems in transportation typically encounter the problem of multiperiod travel times. That is, it can take several time periods (and possibly many) to

get from one location to another. If these travel times are treated deterministically, then we can handle these through an augmented state variable such as

$R_{t,at'}$ = number of resources that, at time t, will have attribute a at time t'.

This is a family of resource variables over the range $t' \geq t$. The representation works as long as travel times are deterministic. For rail applications, this is actually a pretty poor approximation. The most general model retains the history of prior decisions:

$x^h_{t'ad,t}$ = number of resources which at time $t' \leq t$ had attribute a and were acted on by decision d and which by time t have not yet completed their trip.

We use the notation $x^h_{t't}$ rather than the resource variable $R_{tt'}$ since we are literally keeping a history of prior decisions. This representation is fairly complex but provides for any probability model for transit times.

12.6.4 Multicommodity problems

Multicommodity problems are common in rail applications because there are different rail cars that can be substituted. Rail cars are typically organized in groups, with different types of cars arising within a group. A customer may request a specific type of car within a group, while other customers can accept any car within a group. It is with the latter type of customer that substitution occurs.

If we use linear recourse function approximations, the single-period problems reduce easily to pure networks. When we use nonlinear approximations, the single-period problems produce sequences of (integer) multicommodity network flow problems, depicted in Figure 12.8. In sharp contrast with multiperiod, multicommodity flow problems, these single-period problems are actually relatively easy to solve and usually produce integer solutions.

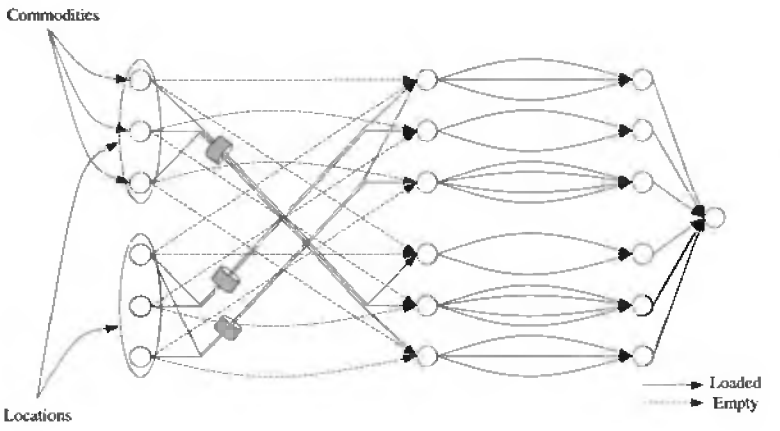


Figure 12.8. *Network structure of a one-period multicommodity problem.*

Table 12.1. *Percentage of integer optimal value obtained using CAVE for second set of deterministic experiments with single-period time windows (network problems).*

Locations	Planning Horizon		
	15	30	60
20	100.00%	100.00%	100.00%
40	100.00%	99.99%	100.00%
80	99.99%	100.00%	99.99%

12.6.5 Experimental results for an empty car distribution problem

Separable, piecewise linear value function approximations are especially easy to use for multistage stochastic, dynamic car distribution problems. They provide very-near-optimal solutions for two-stage problems, but this performance arises in part because the initial state is deterministic. As the value function approximations converge, the (incomplete) state variable for the second stage also converges.

This same property does not hold true with multistage problems. The vector of resources R_t for a general time $t > 0$ will be random, which means that the separable value functions for stage $t + 1$ must be "good" over a range of outcomes of R_t. It would be comparable to testing the logic for a two-stage problem where the starting state is random.

There are several ways to test the logic for a multistage problem. The first is to test it on deterministic instances. In practice, it is common for a company to use a snapshot of history (which is deterministic) to see how well the logic would have performed in the past. The research community might challenge this test, but it is a common one in industry. Besides, if the algorithm works well for stochastic problems, it should also work well for deterministic problems.

Table 12.1 gives a set of results reported in [13] where a variant of a SAFE algorithm (called the CAVE algorithm, given in [15]) is tested on a deterministic, dynamic network. The algorithm was tested on 20-, 40-, and 80-location problems, with 15-, 30-, and 60-period horizons. The results indicate near-optimal performance. The success for this problem class is closely related to the success for two-stage problems: the value functions steadily converge, meaning the state vector R_t converges to a point, allowing the separable approximations to be locally accurate.

A second set of experiments (from [37]) was conducted on multistage stochastic resource allocation problems. These experiments were run with multiple commodities and four different sets of substitution matrices. The substitution matrix gives the fraction of revenue (denoted γ_{kl}) if commodity type k is used to serve task type l (where $\gamma_{kk} = 1$). The approximate dynamic programming formulation was then compared to a rolling horizon procedure that made decisions in each time period by optimizing over the next 20 periods using a (rounded) point forecast of future events. This approach represents standard industry practice for these problems.

Figure 12.9 shows the comparison for four data sets of the approximate dynamic

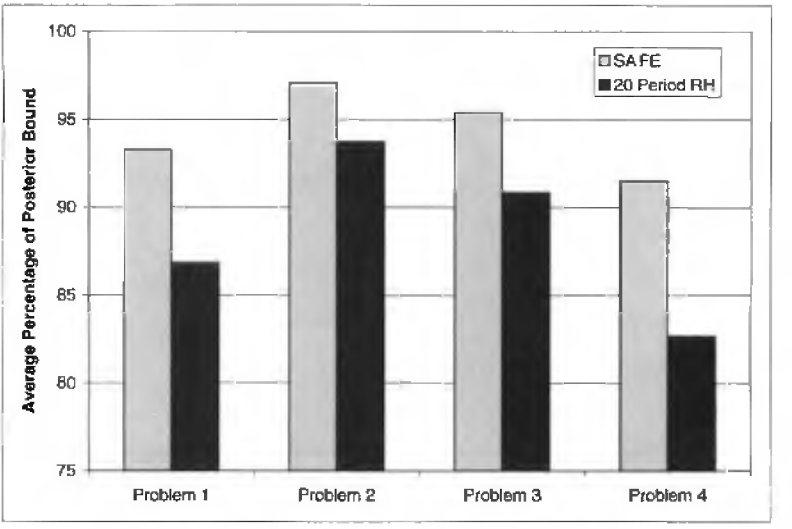


Figure 12.9. *Comparison of SAFE approximations to rolling horizon procedures for 20-location data sets and different substitution matrices.*

programming method (using a SAFE algorithm for performing functional approximations) against the rolling horizon procedure. The differences are substantial. These results do assume that if a demand is not served on time, then it is lost. Flexibility in the time at which a task can be served would probably reduce this gap.

An experiment was run using the car distribution data from a major railroad. This data set used actual demand forecasts (and the distributions derived from this forecasting process), an actual supply snapshot, and actual transit times. Orders due on one day of the week could be served earlier or later (with a penalty) but could not be carried over to the following week. These runs were done with a system being developed for production use and are thought to be quite realistic.

The figures that follow show several statistics as the algorithm progresses. The first iteration uses recourse functions equal to zero and therefore represents a classical myopic model of the sort widely used in industrial practice. Improvements past the first iteration represent a measure of how well our adaptive learning logic performs relative to the standard industrial model.

Figure 12.10 shows total revenue (the top line), total profits (second line), empty repositioning costs (the third line), service penalties (the fourth line for the first few iterations), and finally the costs of holding cars at certain locations (there are selected locations where there is no holding cost). We note that profits improve steadily over the first fifty iterations before roughly leveling out. Over this range, revenue actually decreases at first, but this is counterbalanced by reductions in empty cost and service penalties. We note that toward the end, the total revenue is slightly higher than it was for the first iteration, indicating that a simple myopic model will do an effective job of covering demand (but will just cover the demands as effectively as the adaptive learning algorithm).

Figure 12.11 shows empty miles as a percent of total miles (empty miles plus the

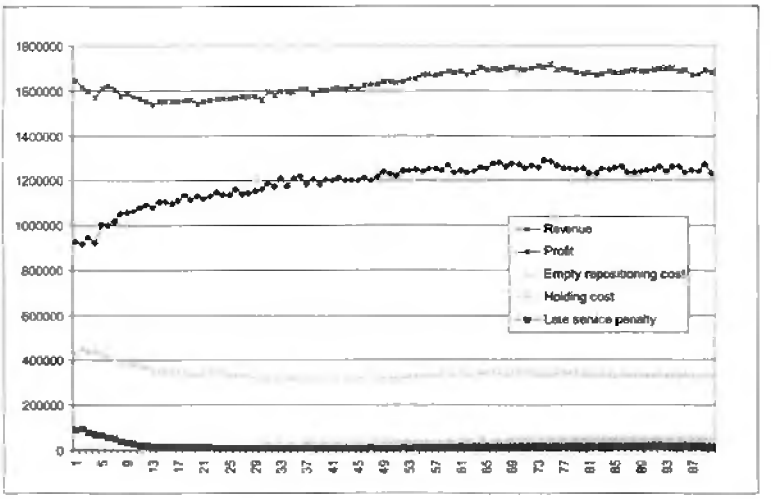


Figure 12.10. *Costs, revenues, and profits as the adaptive learning algorithm progresses.*

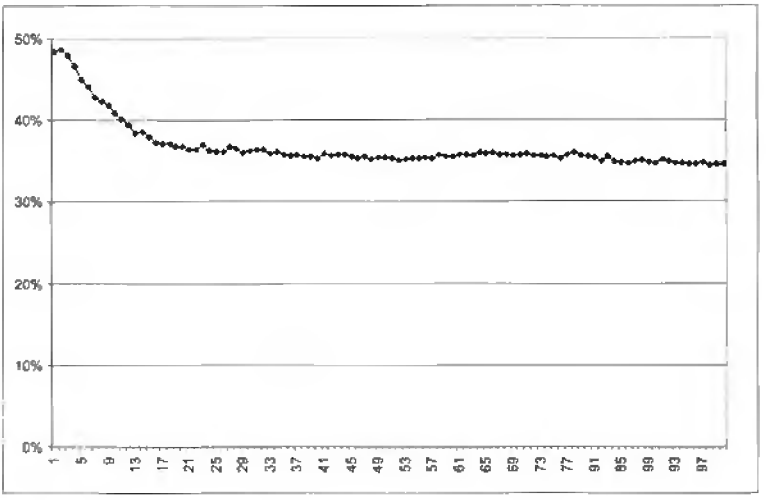


Figure 12.11. *Total empty miles traveled as the learning algorithm progresses.*

loaded miles for each car order). This graph indicates a fairly dramatic reduction in empty miles over the first 20 to 30 iterations. Although the total demand served is also dropping in the early iterations, we have expressed empty miles as a percent of the total. It is likely, however, that some of the orders that require longer empty movements (relative to the distance required for the order) are being turned down at first. Since total revenue is slightly higher than for the first iteration, it appears that the system "learns" how to serve these less profitable orders as the algorithm progresses.

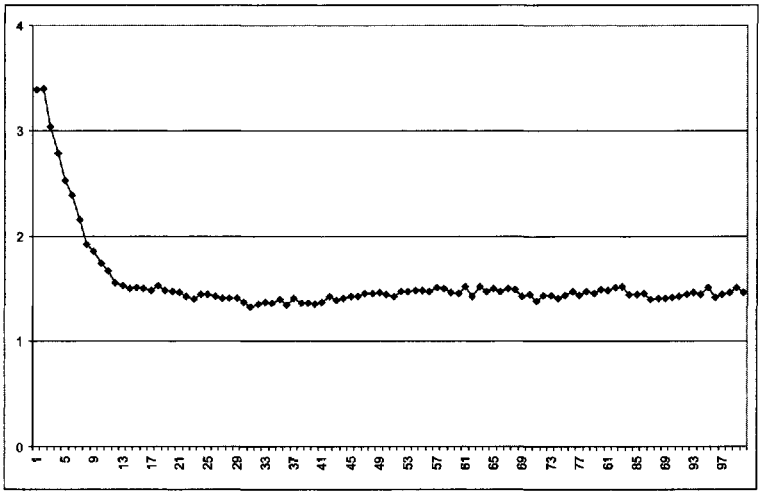


Figure 12.12. *Average days of lateness per order.*

Finally, Figure 12.12 shows the average days of lateness with which each order is being served. As the algorithm progresses, the lateness drops from an average of almost 3.5 days down to slightly more than 1 day.

12.7 Implementation issues

Real applications of these methods to actual problems in rail car distribution invariably introduce a set of issues that range from new areas of research to the usual collection of war stories. Some of these are as follows.

1. *Storing cars.* When there are excess cars, it may be necessary to store cars at predefined locations. One of the challenges of a stochastic model is to store the right number of cars, keeping in mind that we would like to keep enough cars on hand to handle unusual spikes in demand. This is a classic overage/underage problem that stochastic models should excel at. The problem is that the solution depends very much on the cost of overage (what is the cost of keeping idle cars on hand) and the cost of underage (what is the cost of unsatisfied demand). As with most of these problems, the cost of unsatisfied demands is difficult to estimate, and often we are choosing numbers to satisfied a target coverage level.

2. *Transit times.* Transit times are perhaps the Achilles heel of a car distribution system. Stochastic models can, in principle, handle stochastic transit times, but estimating these transit times can be quite a challenge. Standard procedure would be to simply use historical data, but these data can be notoriously unreliable. The solution is to combine engineered transit times (which would normally represent the fastest time a car can achieve between two points), but even this is not entirely reliable. Engineered

transit times assume that the car follows a schedule, while in practice some cars will move, even regularly, on special trains that do not appear in the schedule.

3. *Repositioning.* The heart of any stochastic model involves repositioning cars because of what might happen. Planners at railroads do this all the time, but trusting a model to do this is going to be a big step.

4. *Train schedules.* Our model does not reflect train capacities, which implies that we may try to move cars in a way that results in cars having to wait for available train capacity.

5. *Short cuts.* The data might say that it takes 6 days to move a car, while operations will say they can do it in 2 days by putting the car on a unit train that is not in the schedule. Although this is a clear data error, it represents one of a host of data issues that will arise in the course of the project.

12.8 Bibliographic notes

There is a long history of modeling car distribution problems (and related fleet management problems) as deterministic linear programs, initially as pure networks [11, 24, 16, 26, 41, 18, 19, 42, 25] followed by more general models [17]. Joborn [21] has developed and implemented a model for car distribution at the Swedish National Railway which considers trains' schedules and capacities, producing a relatively large integer multicommodity flow problem. Dejax and Crainic [7] provide a thorough review of the research in fleet management at the time, covering both rail and intermodal container applications.

Car distribution and similar problems in fleet management were some of the original motivating applications for stochastic programming [6, 10]. Jordan and Turnquist [22] and Powell [28] formulated stochastic fleet management problems as nonlinear programming problems using decision variables that sent a fraction of the supply of vehicles through a node to a particular destination. This modeling approach, however, was not able to provide an accurate model of the problem. Crainic, Gendreau, and Dejax [5] provide a general stochastic, dynamic model for container distribution but do not provide an algorithm.

Car distribution is a classic multistage stochastic programming problem. Most of the techniques we use are based on results from the theory of stochastic approximation procedures [35, 4, 8, 12], stochastic gradient methods [9], general stochastic linear programming [2, 20, 23], and dynamic programming (both classical methods, reviewed in [34], and approximate methods, such as those covered in [1, 36]). The car distribution problem represents a class of resource allocation problem that lends itself to a mixture of dynamic programming and stochastic programming formulations.

The detailed study of two-stage stochastic networks was initiated in [39, 40, 3] in the context of general two-stage stochastic programs and independently in [28, 29, 30] in the context of dynamic fleet management. The latter group of papers eventually led to a line of research [31, 33, 15, 13] which developed the idea of using Monte Carlo sampling of stochastic gradients to help build up separable, piecewise linear value function approximations. This particular class of functions is particularly convenient for problems requiring integer solutions. Recent research [27, 38, 32] has begun to formalize the theory

behind estimating separable concave approximations, with a growing body of experimental research supporting their effectiveness [13, 14, 37].

The work in this chapter is based in part on a project to develop and implement a production car distribution system for a major railroad. This project has provided us with a host of real issues involving the modeling of evolving information processes which stochastic programming (broadly defined) can handle. The problem is also attractive because the standard modeling and algorithmic techniques are all based on deterministic approximations.

Acknowledgment

This research was supported in part by grant AFOSR-F49620-93-1-0098 from the Air Force Office of Scientific Research.

Bibliography

[1] D. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA, 1996.

[2] J. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer-Verlag, New York, 1997.

[3] J. R. Birge and S. W. Wallace, *A separable piecewise linear upper bound for stochastic linear programs*, SIAM J. Control Optim., 26 (1988), pp. 725–739.

[4] J. Blum, *Multidimensional stochastic approximation methods*, Ann. Math. Statist., 25 (1954), pp. 737–744.

[5] T. Crainic, M. Gendreau, and P. Dejax, *Dynamic stochastic models for the allocation of empty containers*, Oper. Res., 41 (1993), pp. 102–126.

[6] G. Dantzig, *Linear programming under uncertainty*, Management Sci., 1 (1955), pp. 197–206.

[7] P. Dejax and T. Crainic, *A review of empty flows and fleet management models in freight transportation*, Transportation Sci., 21 (1987), pp. 227–247.

[8] A. Dvoretzky, *On stochastic approximation*, in Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, J. Neyman, ed., University of California Press, Berkeley, 1956, pp. 39–55.

[9] Y. Ermoliev, *Stochastic quasigradient methods*, in Numerical Techniques for Stochastic Optimization, Y. Ermoliev and R. Wets, eds., Springer-Verlag, Berlin, 1988, pp. 141–185.

[10] Y. Ermoliev, T. Krivets, and V. Petukhov, *Planning of shipping empty seaborne containers*, Cybernetics, 12 (1976), p. 664.

[11] G. Feeney, *Controlling the distribution of empty cars*, in Proceedings of the Tenth National Meeting, Operations Research Society of America, 1957.

[12] E. G. Gladyshev, *On stochastic approximation*, Theory Prob. Appl., 10 (1965), pp. 275–278.

[13] G. GODFREY AND W. B. POWELL, *An adaptive, dynamic programming algorithm for stochastic resource allocation problems* I: *Single period travel times*, Transportation Sci., 36 (2002), pp. 21–39.

[14] G. GODFREY AND W. B. POWELL, *An adaptive, dynamic programming algorithm for stochastic resource allocation problems* II: *Multi-period travel times*, Transportation Sci., 36 (2002), pp. 40–54.

[15] G. A. GODFREY AND W. B. POWELL, *An adaptive, distribution-free approximation for the newsvendor problem with censored demands, with applications to inventory and distribution problems*, Management Sci., 47 (2001), pp. 1101–1112.

[16] S. GORENSTEIN, S. POLEY, AND W. WHITE, *On the Scheduling of the Railroad Freight Operations*, Technical Report 320-2999, IBM Philadelphia Scientific Center, Philadelphia, 1971.

[17] A. HAGHANI, *Formulation and solution of a combined train routing and makeup, and empty car distribution model*, Transportation Res. B, 23 (1989), pp. 433–452.

[18] H. HERREN, *The distribution of empty wagons by means of computer: An analytical model for the Swiss Federal Railways (SSB)*, Rail Internat., 4 (1973), pp. 1005–1010.

[19] H. HERREN, *Computer controlled empty wagon distribution on the SSB*, Rail Internat., 8 (1977), pp. 25–32.

[20] G. INFANGER, *Planning under Uncertainty: Solving Large-Scale Stochastic Linear Programs*, The Scientific Press Series, Boyd and Fraser, New York, 1994.

[21] M. JOBORN, *Optimization of Empty Freight Car Distribution in Scheduled Railways*, Ph.D. thesis, Department of Mathematics, Linkoping University, Linkoping, Sweden, 2001.

[22] W. JORDAN AND M. TURNQUIST, *A stochastic dynamic network model for railroad car distribution*, Transportation Sci., 17 (1983), pp. 123–145.

[23] P. KALL AND S. WALLACE, *Stochastic Programming*, John Wiley, New York, 1994.

[24] C. LEDDON AND E. WRATHALL, *Scheduling empty freight car fleets on the Louisville and Nashville railroad*, in Second International Symposium on the Use of Cybernetics on the Railways, Montreal, Canada, 1967, pp. 1–6.

[25] V. MENDIRATTA AND M. TURNQUIST, *A model for the management of empty freight cars*, Trans. Res. Rec., 838 (1982), pp. 50–55.

[26] S. MISRA, *Linear programming of empty wagon disposition*, Rail Internat., 3 (1972), pp. 151–158.

[27] K. PAPADAKI AND W. B. POWELL, *A Discrete On-Line Monotone Estimation Algorithm*, Technical Report, Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ, 2002.

[28] W. B. Powell, *A stochastic model of the dynamic vehicle allocation problem*, Transportation Sci., 20 (1986), pp. 117–129.

[29] W. B. Powell, *An operational planning model for the dynamic vehicle allocation problem with uncertain demands*, Transportation Res. B, 21 (1987), pp. 217–232.

[30] W. B. Powell, *A comparative review of alternative algorithms for the dynamic vehicle allocation problem*, in B. Golden and A. Assad, eds., Vehicle Routing: Methods and Studies, North Holland, Amsterdam, 1988, pp. 249–292.

[31] W. B. Powell and T. A. Carvalho, *Dynamic control of logistics queueing network for large-scale fleet management*, Transportation Sci., 32 (1998), pp. 90–109.

[32] W. B. Powell, A. Ruszczynski, and H. Topaloglu, *Learning algorithms for separable approximations of stochastic optimization problems*, Math. Oper. Res., 29 (2004), pp. 814–836.

[33] W. B. Powell, J. A. Shapiro, and H. P. Simão, *An adaptive dynamic programming algorithm for the heterogeneous resource allocation problem*, Transportation Sci., 36 (2002), pp. 231–249.

[34] M. L. Puterman, *Markov Decision Processes*, John Wiley, New York, 1994.

[35] H. Robbins and S. Monro, *A stochastic approximation method*, Ann. Math. Statist., 22 (1951), pp. 400–407.

[36] R. Sutton and A. Barto, *Reinforcement Learning*, The MIT Press, Cambridge, MA, 1998.

[37] H. Topaloglu and W. B. Powell, *Dynamic programming approximations for stochastic, time-staged integer multicommodity flow problems*, Inform. J. Comput., to appear.

[38] H. Topaloglu and W. B. Powell, *An algorithm for approximating piecewise linear functions from sample gradients*, Oper. Res. Lett., 31 (2003), pp. 61–76.

[39] S. Wallace, *Solving stochastic programs with network recourse*, Networks, 16 (1986), pp. 295–317.

[40] S. Wallace, *A piecewise linear upper bound on the network recourse function*, Math. Program., 38 (1987), pp. 133–146.

[41] W. White, *Dynamic transshipment networks: An algorithm and its application to the distribution of empty containers*, Networks, 2 (1972), pp. 211–236.

[42] W. White and A. Bomberault, *A network algorithm for empty freight car allocation*, IBM Systems J., 8 (1969), pp. 147–171.

This page intentionally left blank

Chapter 13

Modeling Production Planning and Scheduling under Uncertainty

A. Alonso-Ayuso, L. F. Escudero,† and M. T. Ortuño‡*

13.1 Introduction

Production management is concerned with determining supply, production, and stock levels in raw materials, subassemblies at different levels of the given Bills of Material (BoM), end products and information exchange through (possibly) a set of factories, depots and dealer centers of a given production, and a service network to meet fluctuating demand requirements; see [32, 37, 49], among others. Four key aspects of the problem are identified, namely, supply chain topology, time, uncertainty, and cost. The uncertainty aspect of the problem is due to the stochasticity inherent to some parameters for dynamic (multiperiod) planning problems. The main uncertain parameters are product demand and price, raw material supply cost, production cost, operation execution duration, resource requirement by the operations execution, and resource availability over a time horizon.

In these circumstances, we follow the classical taxonomy of planning and scheduling problems in strategic, tactical, and operational problems proposed by Anthony [6]. The strategic production planning consists of deciding on the production topology, plant sizing, product selection, and product allocation among plants. The objective is the maximization (in constant terms) of the expected benefit given by the product net profit along the time horizon minus the investment depreciation and operation costs. The tactical production planning problem consists of deciding on the best utilization of the available resources including vendors, factories, depots, and dealer centers along the time horizon, such that

*Escuela de CC. Experimentales y Tecnología, Universidad Rey Juan Carlos, 28933 Móstoles, Madrid, Spain (a.alonso@escet.urjc.es).

†Centro de Investigación-Operativa, Universidad Miguel Hernández, 03202 Elche, Alicante, Spain (escudero @umh.es),

‡Departamento de Estadística e I. O., Universidad Complutense de Madrid, 28040 Madrid, Spain (mteresa@ mat.ucm.es).

given targets are met at a minimum cost. The tactical production planning assumes a given production topology. The operational problem consists of determining the operations assignment to machines as well as the sequencing and scheduling of jobs and operations along a time horizon, given a production topology as well as certain demand of jobs and operations to satisfy.

Given today's state-of-the-art optimization tools, deterministic strategic and tactical planning and operations sequencing and scheduling should not present major difficulties for problem solving for moderate-size instances, at least. However, it has long been recognized [10, 20] that traditional deterministic optimization is not suitable for capturing the truly dynamic behavior of most real-world applications, and, certainly, production planning and scheduling is one of them. The main reason is that such applications involve data uncertainties which arise because information that will be needed in subsequent decision stages is not available to the decision maker when the decision must be made. For good textbooks, see, e.g., [14, 39, 40, 59].

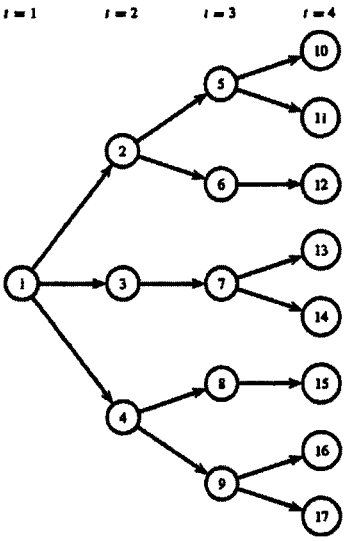
There is a vast literature on dynamic production planning and scheduling. See hierarchical approaches in [15]; single-level-based systems in [41]; multilevel-based systems in [65] and [29]; and systems with lot sizing, inventory holding, and setup considerations in [25, 48, 65, 71, 72, 73], among others.

However, most of the literature presents models and algorithmic schemes for deterministic environments. The uncertainty inherent in most of the important parameters is not dealt with. Moreover, the treatment of stochasticity is relatively recent in production planning. See [1, 9, 18, 26, 34, 49, 51, 68, 70], among others, for interesting approaches on production planning problem solving. Some of the above references are scenario-based approaches to deal with the uncertainty via the nonanticipativity principle; see [60].

Most of the stochastic approaches for production planning consider only tactical decisions (modeled by using continuous variables) usually related to supply, production, and market shipment of raw materials and products; see, e.g., [34]. Some schemes (see, e.g., [1, 2, 49]) address strategic production planning under uncertainty (modeled by using continuous and 0–1 variables). Very few schemes (e.g., see [2, 52]) deal with operational sequencing and scheduling problems (modeled by using 0–1 variables).

Stochastic 0–1 programming has a broad application field; see [3, 31, 32, 17, 44, 45, 47, 54, 55, 63, 64, 67, 74], among others. See also papers in this volume. These approaches use some sort of Benders [12] and Lagrangian decomposition schemes. See also [62].

We present a set of representative strategic, tactical, and operational production planning and scheduling application cases with uncertainty in their main parameters. The models that we consider aim to optimize the expected objective function value. However, there are some other approaches that additionally deal with the mean risk measures by considering semideviations (see [56]) and excess probabilities (see [22, 62, 64, 69], among others). In any case, the uncertainty is to be treated via a scenario analysis approach. To illustrate this concept, consider Figure 13.1: each node in the figure represents a point in time where a decision can be made. Once a decision is made, some contingencies can happen (e.g., in this example the number of contingencies is three for time period $t = 2$), and information related to these contingencies is available at the beginning of the next stage (here, time period). This information structure is visualized as a tree, where each root-to-leaf path represents one specific scenario and corresponds to one realization of the whole set of the uncertain parameters. Each node in the tree can be associated with a scenario group, such



$\Omega = \Omega_1 = \{10, 11, \ldots, 17\}; \Omega_2 = \{10, 11, 12\}$

$\mathcal{G}_2 = \{2, 3, 4\}; t(5) = 3$

$\mathcal{N}^9 = \{1, 4\}; \mathcal{N}_2 = \{2, 5, 6, 10, 11, 12\}$

Figure 13.1. *Scenario tree.*

that two scenarios belong to the same group in a given stage, provided that they have the same realizations of the uncertain parameters up to that stage. Accordingly with the nonanticipativity principle (see [60]), both scenarios should have the same value for the related variables with the time index up to the given stage.

Let the following notation related to the scenario tree be used in most of the models described in this paper:

$\mathcal{T}$ = set of time periods along the time horizon;

Ω = set of scenarios;

$\mathcal{G}$ = set of scenario groups;

$\mathcal{G}_t$ = set of scenario groups in time period t for $t \in \mathcal{T}$ ($\mathcal{G}_t \subseteq \mathcal{G}$);

Ω_g = set of scenarios in group g for $g \in \mathcal{G}$ ($\Omega_g \subseteq \Omega$);

$\mathcal{N}^g$ = set of scenario groups $\{k\}$ such that $\Omega_g \subseteq \Omega_k$ for $g \in \mathcal{G}$ ($\mathcal{N}^g \subset \mathcal{G}$), that is, set of scenario groups (one for each time period) whose subsets of scenarios include the set of scenarios in group g; see that the (unique) predecessor path from the associated node with scenario group g to the root node in the given scenario tree passes through all the associated nodes with the scenario groups in $\mathcal{N}^g$;

$\mathcal{N}_g$ = set of scenario groups $\{k\}$ such that $\Omega_k \subseteq \Omega_g$ for $g \in \mathcal{G}$ ($\mathcal{N}_g \subseteq \mathcal{G}$), that is, set of scenario groups whose subsets of scenarios are included in the set of scenarios in group g; there is a successor subtree in the scenario tree whose nodes are the nodes associated with the set $\mathcal{N}_g$ of scenario groups and the root node is associated with scenario group g for $g \in \mathcal{G}$; for technical reasons it is assumed that $g \in \mathcal{N}_g$;

$w^g =$ weight factor representing the likelihood that is associated with scenario group g for $g \in \mathcal{G}$; note that $w^g = \sum_{\omega \in \Omega_g} w_\omega$, where w_ω gives the likelihood that the modeler associates with scenario ω, for $\omega \in \Omega$, and $\sum_{\omega \in \Omega} w_\omega = 1$;

$t(g) =$ time period for scenario group g for $g \in \mathcal{G}$; note that $g \in \mathcal{G}_{t(g)}$.

We present different types of models depending on the type of recourse to consider, namely, simple, partial, and full recourse (which should be based on the type of problem to address). Given the potentially large number of scenarios that can be needed, decomposition schemes for problem solving are required. Two types of mathematical model representations are considered, namely, the compact representation and the splitting variable representation. The first is very amenable for Benders decomposition; the second is very amenable for such types of algorithmic schemes as augmented Lagrangian decomposition for models with only continuous variables and Lagrangian decomposition and branch-and-fix coordination for mixed and pure 0–1 models. Most of the problems to be addressed are two-stage and multistage problems. The decisions of interest in dynamic environments are the decisions to be made in the first stage, such that they are not subordinated to any scenario but all scenarios are considered via the conditional decisions to be made along the other stages.

Section 13.2 presents the mixed 0–1 deterministic equivalent model (DEM) for the two-stage strategic production planning problem to determine the production topology and product selection. Section 13.3 presents a mixed 0–1 modeling framework for determining the tactical production planning in a multistage production environment with logical constraints. Section 13.4 presents an LP model for the tactical supply chain planning in a multilevel, multistage environment for determining the production, stock, and supplying of the products and raw materials at the different levels of the product BoM. Of particular interest is the modelization of multiperiod linking variables. Section 13.5 presents a pure 0–1 model for the two-stage problem that consists of selecting a set of machines in the first stage such that a scenario-dependent set of jobs is assigned to the machines for processing (in the second stage) at an expected minimum cost. A second version of the problem is modeled by using a single recourse approach. This other problem consists of assigning a given set of jobs to the machines, where the uncertainty continues to rely on the processing time of the jobs in the (parallel, unrelated) machines. Section 13.6 presents a pure 0–1 model for production sequencing and scheduling in a multistage, multitask environment, where the uncertainty relies in the resource requirement by the operations to be performed, the resource availability over time, and the operation scheduling cost. The first source of uncertainty is known only during the operation assignment. However, the resource availability at each time period will be known at the beginning of the time period. Section 13.7 concludes.

13.2 Production topology and product selection with full recourse

13.2.1 Problem statement

A time horizon is a set of (consecutive and integer) time periods where the production planning is considered. A product is any item whose selection and production volume is

decided by management. The stock of a product is its available volume at the end of a given time period. Assume that the cycle time (i.e., lead time) of any unit product is smaller that the length of the given periods in the time horizon.

A plant is a capacitated location where the products are processed. The plants may have different production capacity levels. The term plant investment will be used for the amount of a given currency that is needed for expanding a plant from, say, level $k-1$ to level k. Observe that the expansion to level $k=1$ means that a plant will be open.

Some parameters are deterministic by nature, or the optimal solution may not be very sensitive to their variability. However, the product net profit and demand as well as the production cost are uncertain parameters, mainly, for long time horizons, as it is usually the case for strategic planning. The available information for the uncertain parameters can be structured in a set of scenarios.

The goal of the strategic production planning problem that is considered in this section consists of determining the production topology, i.e., plant sizing, product selection, and product allocation among plants. The objective is the maximization (in constant terms) of the expected benefit. It is given by the product net profit minus the operation costs and the plant investment depreciation cost over the scenarios along the time horizon.

Two stages are considered. The first stage is devoted to the strategic decisions about plant sizing and product allocation to plants. The second stage is devoted to the tactical decisions about the product volume to be processed and stored in the plants and the product volume to be shipped from the plants to the market sources at each time period along the time horizon, given the production topology decided at the first stage. The strategic decisions, besides satisfying their related first-stage constraints, will take into consideration the product expected net profit and operation expected cost related to the tactical environment besides the investment depreciation cost.

See in [5] a generalization of the model considered below to a multilevel production environment and vendor selection for raw material.

13.2.2 Mixed 0–1 DEM

The following are additional definitions and notation for the elements of the strategic production planning model.

Sets:

$\mathcal{I}$ = set of plants;

$\mathcal{J}$ = set of products;

$\mathcal{I}_j$ = set of plants that are available for processing product j for $j \in \mathcal{J}$ ($\mathcal{I}_j \subseteq \mathcal{I}$);

$\mathcal{T}_i$ = set of time periods where a capacity expansion for plant i is allowed, for $i \in \mathcal{I}$ ($\mathcal{T}_i \subseteq \mathcal{T}$), besides time period $t=0$ (i.e., first stage);

$\mathcal{K}_i$ = set of capacity expansion levels for plant i for $i \in \mathcal{I}$;

$\mathcal{M}_j$ = set of market sources for product j for $j \in \mathcal{J}$.

Technical and logistic parameters:

$\widetilde{N}$ = maximum number of plants to be open;

$\widehat{N}$ = maximum number of products to be processed;

$\underline{N}_j, \overline{N}_j$ = conditional minimum and maximum, respectively, number of plants, if any, where product j can be processed for $j \in \mathcal{J}$;

$\overline{N}^i$ = maximum number of products to be processed in plant i at any time period, if any, for $i \in \mathcal{I}$;

P_t = available budget for plant capacity building/expansion at time period t for $t \in \{0\} \cup \mathcal{T}$; note that by convention, plant building (i.e., capacity expansion level $k = 1$) can occur only at time period $t = 0$;

$\underline{X}^i_j, \overline{X}^i_j$ = conditional minimum and maximum, respectively, volume of product j, if any, that can be processed in plant i at any time period for $i \in \mathcal{I}_j$, $j \in \mathcal{J}$;

$\underline{S}^i_{jt}, \overline{S}^i_j$ = conditional minimum and maximum volume of product j, if any, that can be in stock in plant i at (the end of) time period t and at any time period, respectively, for $i \in \mathcal{I}_j$, $j \in \mathcal{J}, t \in \mathcal{T}$;

o^i_j = unit capacity consumption of plant i by product j for $i \in \mathcal{I}_j$, $j \in \mathcal{J}$;

$\underline{p}_i$ = minimum capacity usage of plant i, if any, at any time period for $i \in \mathcal{I}$;

p^k_i = production capacity increment from level $k-1$ to level k in plant i for $k \in \mathcal{K}_i$, $i \in \mathcal{I}$.

Deterministic cost coefficients:

a^k_{it} = budget required for the capacity expansion from level $k-1$ to level k in plant i at time period t for $k \in \mathcal{K}_i, t \in \{0\} \cup \mathcal{T}_i, i \in \mathcal{I}$;

q^k_{it} = depreciation cost (along the time horizon) of the investment a^k_{it} related to the kth capacity expansion level in plant i at time period t for $k \in \mathcal{K}_i, t \in \{0\} \cup \mathcal{T}_i$, $i \in \mathcal{I}$;

h^i_j = unit holding cost of product j in plant i at any time period for $i \in \mathcal{I}_j$, $j \in \mathcal{J}$.

Uncertain parameters and cost coefficients:

$D^{m\omega}_{jt}$ = demand of product j from market source m at time period t under scenario ω for $m \in \mathcal{M}_j$, $j \in \mathcal{J}, t \in \mathcal{T}, \omega \in \Omega$;

$p^{im\omega}_{jt}$ = unit net profit from selling product j from plant i to market source m at time period t under scenario ω, including local taxes, transport cost, and others for $i \in \mathcal{I}_j, m \in \mathcal{M}_j, j \in \mathcal{J}, t \in \mathcal{T}, \omega \in \Omega$;

$c_{jt}^{i\omega}$ = unit processing cost of product j in plant i at time period t under scenario ω for $i \in \mathcal{I}_j$, $j \in \mathcal{J}$, $t \in \mathcal{T}$, $\omega \in \Omega$.

Strategic variables

These are 0–1 variables, such that

$\alpha_j = 1$ if product j is selected for processing, and 0 otherwise, for $j \in \mathcal{J}$;

$\beta_j^i = 1$ if product j is processed in plant i, and 0 otherwise, for $i \in \mathcal{I}_j$, $j \in \mathcal{J}$;

$\gamma_{it}^k = 1$ if plant i has capacity level k *at least* at period t, and 0 otherwise, for $k \in \mathcal{K}_i$, $i \in \mathcal{I}$, $t \in \{0\} \cup \mathcal{T}$.

Operation variables

They are continuous variables for each product j, time period t, and scenario ω, for $j \in \mathcal{J}$, $t \in \mathcal{T}$, $\omega \in \Omega$.

$x_{jt}^{i\omega}$ = volume of product j to be processed in plant i at time period t under scenario ω for $i \in \mathcal{I}_j$;

$s_{jt}^{i\omega}$ = stock volume of product j in plant i at (the end of) time period t under scenario ω for $i \in \mathcal{I}_j$;

$y_{jt}^{im\omega}$ = volume of product j to be shipped from plant i to market source m at time period t under scenario ω for $i \in \mathcal{I}_j$, $m \in \mathcal{M}_j$.

Let the following be the compact representation of the DEM for the two-stage stochastic problem.

Objective

Determine the production topology and product selection and allocation to maximize the total expected benefit, subject to the constraints (13.2)–(13.23).

$$\max \sum_{\omega\in\Omega} w_\omega \left[\sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{I}_j} \sum_{m\in\mathcal{M}_j} p_{jt}^{im\omega} y_{jt}^{im\omega} - \sum_{t\in\mathcal{T}} \sum_{j\in\mathcal{J}} \sum_{i\in\mathcal{I}_j} (c_{jt}^{i\omega} x_{jt}^{i\omega} + h_j^i s_{jt}^{i\omega}) \right.$$
$$\left. - \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i\setminus\{1\}} \sum_{t\in\mathcal{T}_i} q_{it}^k (\gamma_{it}^{k\omega} - \gamma_{i,t-1}^{k\omega}) \right] - \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} q_{i0}^k \gamma_{i0}^k. \quad (13.1)$$

Stage 1 (strategic) constraints are

$$\sum_{i\in\mathcal{I}} \gamma_{i0}^1 \le \widetilde{N}, \quad (13.2)$$

$$\gamma_{i0}^{k-1} \ge \gamma_{i0}^k \quad \forall k \in \mathcal{K}_i \setminus \{1\},\ i \in \mathcal{I}, \quad (13.3)$$

$$\sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} a_{i0}^k \gamma_{i0}^k \le P_0, \quad (13.4)$$

$$\sum_{j\in\mathcal{J}} \alpha_j \le \widehat{N}, \tag{13.5}$$

$$\underline{N}_j \alpha_j \le \sum_{i\in\mathcal{I}_j} \beta_j^i \le \overline{N}_j \alpha_j \quad \forall j \in \mathcal{J}, \tag{13.6}$$

$$\sum_{j\in\mathcal{J}/i\in\mathcal{I}_j} \beta_j^i \le \overline{N}^i \gamma_{i0}^1 \quad \forall i \in \mathcal{I}, \tag{13.7}$$

$$\beta_j^i \le \alpha_j \quad \forall i \in \mathcal{I}_j,\ j \in \mathcal{J}, \tag{13.8}$$

$$\beta_j^i \le \gamma_{i0}^1 \quad \forall i \in \mathcal{I}_j,\ j \in \mathcal{J}, \tag{13.9}$$

$$\alpha_j \in \{0,1\} \quad \forall j \in \mathcal{J}, \tag{13.10}$$

$$\beta_j^i \in \{0,1\} \quad \forall i \in \mathcal{I}_j,\ j \in \mathcal{J}, \tag{13.11}$$

$$\gamma_{i0}^k \in \{0,1\} \quad \forall k \in \mathcal{K}_i,\ i \in \mathcal{I}. \tag{13.12}$$

The cover-induced constraint (13.2) ensures that the number of plants in the production system does not exceed the allowed maximum. The variable upper bounding (VUB) constraints (13.3) ensure that the γ-variables are well defined. The knapsack constraint (13.4) takes into account the investment budget. The cover-induced constraint (13.5) bounds the number of products to select. Constraints (13.6) conditionally lower and upper bound the number of plants for processing each product. Constraints (13.7) ensure that the number of products for processing in each plant will not exceed the allowed maximum. The VUB constraints (13.8) and (13.9) are 0–1 redundant inequalities, but their appending results in a tighter model.

Stage 2 (operation) constraints for each scenario are as follows. The time period indexed capacity expansion constraints for scenario $\omega \in \Omega$ are

$$\gamma_{i,t-1}^{1\omega} = \gamma_{it}^{1\omega} \quad \forall i \in \mathcal{I},\ t \in \mathcal{T}, \tag{13.13}$$

$$\gamma_{i,t-1}^{k\omega} = \gamma_{it}^{k\omega} \quad \forall k \in \mathcal{K}_i \setminus \{1\},\ t \in \mathcal{T} \setminus \mathcal{T}_i,\ i \in \mathcal{I}, \tag{13.14}$$

$$\gamma_{i,t-1}^{k\omega} \le \gamma_{it}^{k\omega} \quad \forall k \in \mathcal{K}_i \setminus \{1\},\ t \in \mathcal{T}_i,\ i \in \mathcal{I}, \tag{13.15}$$

$$\gamma_{it}^{k-1,\omega} \ge \gamma_{it}^{k\omega} \quad \forall k \in \mathcal{K}_i \setminus \{1\},\ i \in \mathcal{I},\ t \in \mathcal{T}, \tag{13.16}$$

$$\sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i\setminus\{1\}} a_{it}^k (\gamma_{it}^{k\omega} - \gamma_{i,t-1}^{k\omega}) \le P_t \quad \forall t \in \mathcal{T}, \tag{13.17}$$

$$\underline{p}_i \gamma_{i0}^1 \le \sum_{j\in\mathcal{J}:i\in\mathcal{I}_j} o_j^i x_{jt}^{i\omega} \le \sum_{k\in\mathcal{K}_i} p_i^k \gamma_{it}^{k\omega} \quad \forall i \in \mathcal{I},\ t \in \mathcal{T}, \tag{13.18}$$

where $\gamma_{it}^{1\omega} \equiv \gamma_{it}^1\ \forall t, i$ and $\gamma_{i0}^{k\omega} \equiv \gamma_{i0}^k\ \forall k, i, \omega$.

The time period indexed operation constraints for product $j \in \mathcal{J}$ and time period $t \in \mathcal{T}$ for scenario $\omega \in \Omega$ are

$$s_{j,t-1}^{i\omega} + x_{jt}^{i\omega} = \sum_{m\in\mathcal{M}_j} y_{jt}^{im\omega} + s_{jt}^{i\omega} \quad \forall i \in \mathcal{I}_j, \tag{13.19}$$

$$\underline{X}_j^i \beta_j^i \le x_{jt}^{i\omega} \le \overline{X}_j^i \beta_j^i \quad \forall i \in \mathcal{I}_j, \tag{13.20}$$

$$\underline{S}_{jt}^i \beta_j^i \le s_{jt}^{i\omega} \le \overline{S}_j^i \beta_j^i \quad \forall i \in \mathcal{I}_j, \tag{13.21}$$

$$\sum_{i \in \mathcal{I}_j} y_{jt}^{im\omega} \leq D_{jt}^{m\omega} \alpha_j \quad \forall m \in \mathcal{M}_j, \tag{13.22}$$

$$y_{jt}^{im\omega} \geq 0 \quad \forall i \in \mathcal{I}_j,\ m \in \mathcal{M}_j. \tag{13.23}$$

The stage-two-related constraints have been grouped in two blocks, namely, capacity expansion–related constraints (13.13)–(13.18) and operation-related constraints (13.19)–(13.23). Constraints (13.13) ensure that the plants are open only at time period $t = 0$. Constraints (13.14) ensure that the capacity expansion of the plants will occur only at allowed time periods. Constraints (13.15) and (13.16) assure that the γ-variables are well defined. Constraints (13.17) take into account the capacity expansion budget. The additive VUB (AVUB) constraints (13.18) limit the production from each plant to a conditional minimum, as well as to the maximum capacity given by the expansion plan. Atamtürk, Nemhauser, and Savelsbergh [7] derive several classes of valid inequalities for AVUB inequalities. Constraints (13.19) are the stock balance equations for products. Constraints (13.20) and (13.21) define the semicontinuous character of the production and stock variables, respectively. These constraints imply the nonnegativity of the variables $x_{jt}^{i\omega}$ and $s_{jt}^{i\omega}$. Constraints (13.22) ensure that the product shipment to the market sources does not exceed the related demand, if any.

The compact representation (13.2)–(13.23) can be transformed into a splitting variable representation such that the α-, β-, and γ-variables for time $t = 0$ (i.e., first stage) are replaced by their siblings, say the α^{ω}-, β^{ω}-, and γ^{ω}-variables for each scenario $\omega \in \Omega$. The nonanticipativity constraints (13.24)–(13.26) are appended to the model for $\omega, \omega' \in \Omega$: $\omega \neq \omega'$.

$$\alpha_j^{\omega} - \alpha_j^{\omega'} = 0 \quad \forall j \in \mathcal{J}, \tag{13.24}$$

$$\beta_j^{i\omega} - \beta_j^{i\omega'} = 0 \quad \forall i \in \mathcal{I}_j,\ j \in \mathcal{J}, \tag{13.25}$$

$$\gamma_{it}^{k\omega} - \gamma_{it}^{k\omega'} = 0 \quad \forall k \in \mathcal{K}_i,\ i \in \mathcal{I},\ t \in \mathcal{T}. \tag{13.26}$$

By relaxing (or, for the matter, dualizing) the constraints (13.24)–(13.26), we obtain $|\Omega|$ independent scenario related mixed 0–1 models. The splitting variable representation is very amenable to Lagrangian decomposition approaches; see, e.g., [42, 45, 47, 54, 61, 62, 64]. Additionally, it is very appropriate for branch-and-fix coordination (BFC); see below.

13.2.3 Families of twin nodes in the BFC scheme

Assume the following representation of the model (13.13)–(13.23),

$$\begin{aligned} Z_{IP} = \max\ & ax + \sum_{\omega \in \Omega} w_{\omega}(by^{\omega} + c^{\omega} z^{\omega}) \\ \text{s.t. } & A^1 x = q, \\ & A^2 x + By^{\omega} + Cz^{\omega} = p^{\omega} \quad \forall \omega \in \Omega, \\ & x \in \{0,1\},\ y^{\omega} \in \{0,1\},\ z^{\omega} \geq 0 \quad \forall \omega \in \Omega, \end{aligned} \tag{13.27}$$

where a, b, and c^{ω} are the vectors of the objective function coefficients for scenario $\omega \in \Omega$; x, y^{ω}, and z^{ω} are the vectors of the variables, such that x represents the 0–1 first-stage variables, the strategic α-, β-, and γ-variables. y^{ω} represents the strategic 0–1 second-stage

variables, y_t^ω, $t \in \mathcal{T}$, and z^ω represents the continuous second-stage variables for scenario $\omega \in \Omega$. The tactical variables, A^1 and A^2 are the first-stage and second-stage constraint matrices related to the x-variables, respectively; B and C are the second-stage constraint matrices related to the y^ω- and z^ω-variables, respectively. q and p^ω are the right-hand-side vectors for the first stage and second stage, respectively, for scenario $\omega \in \Omega$. All parameters have conformable dimensions.

The splitting variable representation via scenario of model (13.27) is

$$
\begin{aligned}
Z_{IP} = \max & \sum_{\omega \in \Omega} w_\omega (a x^\omega + b y^\omega + c^\omega z^\omega) \\
\text{s.t.} \quad & A^1 x^\omega = q \quad \forall \omega \in \Omega, \\
& A^2 x^\omega + B y^\omega + C z^\omega = p^\omega \quad \forall \omega \in \Omega, \\
& x^\omega - x^{\omega+1} = 0 \quad \forall \omega = 1, 2, \ldots, |\Omega| - 1, \\
& x^\omega, y^\omega \in \{0, 1\}, z^\omega \geq 0 \quad \forall \omega \in \Omega,
\end{aligned}
\tag{13.28}
$$

where the nonanticipativity constraints are

$$
x^\omega - x^{\omega+1} = 0 \quad \forall \omega = 1, 2, \ldots, |\Omega| - 1. \tag{13.29}
$$

The relaxation of the constraints (13.29) in model (13.28) results in a set of $|\Omega|$ independent models; we can execute, say, a branch-and-fix (BF) procedure for each scenario-related model to ensure the integrality condition. Instead of obtaining independently the optimal solution for each one, elsewhere (see [4, 5]) we propose the ad hoc approach BFC. It is specifically designed to coordinate the node and variable branching for each scenario-related BF tree, such that the relaxed constraints (13.29) are satisfied when fixing the appropriate variables to either one or zero. The proposed approach also coordinates and reinforces the scenario-related BF node pruning, the variable fixing, and the objective function value bounding of the subproblems attached to the nodes.

The main concept in the BFC approach is the families of twin nodes. Any two active nodes are said to be twin nodes if the path from the root node to each of them in their own BF trees has branched or fixed on the same values of the x-variables. A family of twin nodes is a set of nodes such that any node is a twin node to all the other nodes in the family. To satisfy the nonanticipativity constraints (13.29) the branching and fixing of the x-variables must be with the same 0–1 value for the twin nodes. Carøe and Schultz [16] use a similar decomposition approach. However, that approach focuses more on using Lagrangian relaxation to obtain good lower bounds and less on branching and variable fixing. In any case, Lagrangian relaxation schemes can be added on top.

13.2.4 Computational results

Alonso-Ayuso et al. [4, 5] report a very promising experience for a set of instances using the BFC approach. The optimal solution was obtained in 15 of 22 instances within the allowed 30-minute limit on a PC PIII 800 MHz, 128 Mb RAM. The instances have the following dimensions: $|\mathcal{I}| = 6$ plants/warehouses, $|\mathcal{K}_i| = 3$ capacity levels per plant, $|\mathcal{J}| = 12$ products, $|\mathcal{V}| = 24$ vendors (that have not been considered in the model above), $|\mathcal{M}_j| = 2$ markets per product, $|\mathcal{T}| = 10$ time periods, and $|\Omega| = 23$ scenarios. The decisions about

Table 13.1. *Test bed dimensions.*

Case	Single scenario			DEM		
	m	nc	$n01$	m	nc	$n01$
c1	3388	2937	107	76318	65989	899
c2	3458	3068	108	77928	68980	900
c3	3145	2663	103	70795	59775	895
c4	3405	3065	105	76775	68977	897
c5	3933	3654	114	88743	82326	906
c6	3145	2663	103	70795	59775	895
c7	3081	2543	103	69411	57015	895
c8	3894	3634	114	87824	81866	906
c9	3388	2937	107	76318	65989	899
c10	3101	2533	103	69871	56785	895

Table 13.2. *Stochastic solution.*

Case	Z_{LP}	$\underline{Z}_{IP}$	GAP	nn	T_{LP}	T_{IP}	T
c1	238471.13	178366.79	25.20	654	1213.53	1800.00	3013.53
c2	64128.62	0.00(*)	100.00	7	337.03	62.78	399.81
c3	286773.63	224564.20	21.69	2286	548.82	1800.00	2348.82
c4	255419.80	197487.36	22.68	2201	535.76	1800.00	2335.76
c5	53297.06	0.00(*)	100.00	17	825.53	443.85	1269.38
c6	285728.66	226578.02	20.70	2224	585.88	1800.00	2385.88
c7	180256.99	144181.28(*)	20.01	641	293.02	771.26	1064.28
c8	140115.70	89607.39	36.05	269	2104.70	1800.00	3904.70
c9	237866.97	174250.56	26.74	208	1286.03	1800.00	3086.03
c10	173404.62	139738.36(*)	19.41	877	274.15	1439.70	1713.85

(*) Optimality has been proved.

capacity expansion have been restricted to intermediate periods besides the decisions that can be made in the first stage.

To build the scenario tree, different levels of product demand and raw material supply cost have been combined. There are two schemes, one with five demand and five cost levels and one with nine demand and three cost levels. In both schemes the extreme cases have been eliminated.

Table 13.1 shows the dimensions of the scenario-related deterministic model as well as the dimensions of the DEM for a test bed of 10 cases. The headings are as follows: m, number of constraints; nc, number of continuous variables; and $n01$, number of 0–1 variables. Table 13.2 shows our computational experimentation. The headings are as follows: Z_{LP}, solution value of the LP relaxation; $\underline{Z}_{IP}$, value of the incumbent solution for the original problem; GAP, optimality gap (%) defined as $(Z_{LP} - \underline{Z}_{IP})/Z_{LP} \times 100$;

Table 13.3. *The value of the stochastic information.*

Case	EV	WS	$EVPI$	$\underline{Z}_{IP}$	EEV	VSS	SR
c1	208670.34	202582.05	24215.26	178366.79	175113.79	3253.00	93696.07
c2	13812.86	19447.83	19447.83	0.00	-12220.54	12220.54	0.00
c3	259863.19	251195.44	26631.24	224564.20	222660.23	1903.97	123173.46
c4	218827.75	217078.93	19591.57	197487.36	195699.61	1787.75	105932.22
c5	0.00	6525.56	6525.56	0.00	0.00	0.00	0.00
c6	271117.43	249449.34	22871.32	226578.02	217922.07	8655.95	90653.87
c7	169135.22	157942.74	13761.46	144181.28	137174.62	7006.66	0.00
c8	103102.67	107655.30	18047.91	89607.39	89607.39	0.00	30000.55
c9	219875.02	201494.28	27243.72	174250.56	168523.01	5727.55	13870.34
c10	157558.52	145404.96	5666.60	139738.36	139738.36	0.00	52960.84

nn, number of branching nodes for the whole set of $|\Omega| = 23$ BF trees; T_{LP} and T_{IP}, the elapsed time (seconds) to obtain the LP solution and the additional time to obtain the integer solution, respectively; and T, total time.

Given the relaxation of the nonanticipativity constraints (13.24)–(13.26), it is not a surprise that GAP is very big. This fact together with the extremely high dimensions of the problem makes it unrealistic to pretend to prove solution optimality. The 30-minute time limit for branching activity was reached in 6 of the 10 cases. Moreover, Table 13.3 shows some parameters for analyzing the goodness of the stochastic approach; see, e.g., [14] for more details.

The headings are as follows. EV is the solution value for the average scenario (i.e., the expected value approach). $WS = \sum_{\omega\in\Omega} w_\omega Z^\omega_{IP}$ is the wait-and-see solution, where Z^ω_{IP} is the solution value for scenario ω. $EVPI = WS - \underline{Z}_{IP}$ is the expected value of the perfect information for $\underline{Z}_{IP} = Z_{IP}$. $EEV = \sum_{\omega\in\Omega} w_\omega Z^\omega$ is the expected result of the expected value, such that Z^ω is the solution value for the scenario ω deterministic model where the solution for the first stage has been fixed to the optimal solution for the average scenario model. $VSS = \underline{Z}_{IP} - EEV$. SR is the value of the simple recourse approach.

We see that VSS is positive in 7 of the 10 cases; i.e., it is worth the effort to use the stochastic approach instead of obtaining the strategic decisions based on the average scenario parameters. For example, observe that the stochastic solution does not recommend starting business for case c2 (by analyzing the uncertainty of the parameters given by the set of scenarios), but the average scenario solution does. As a consequence there is an EEV expected loss derived from the (wrong) decision based on the average scenario. On the other hand, observe that WS is a stronger upper bound of the optimal full recourse solution than the LP bound.

Finally, Figures 13.2 and 13.3 give the solution value for the best and worst scenarios. The legend is as follows. SP1, RP1, and EV1 are optimal solution values for the scenario problem under study, where the first-stage variables are fixed to the optimal solution for the own scenario problem, the incumbent solution for the full recourse problem, and the optimal solution for the average scenario problem, respectively. A conclusion that can be drawn from the computational experimentation is that, although in most of the cases the expected values RP and EEV do not differ much, the stochastic solution is more robust.

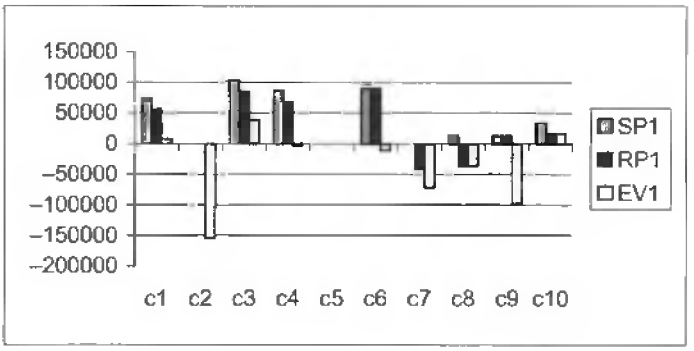


Figure 13.2. *Objective value solution of the worst scenario.*

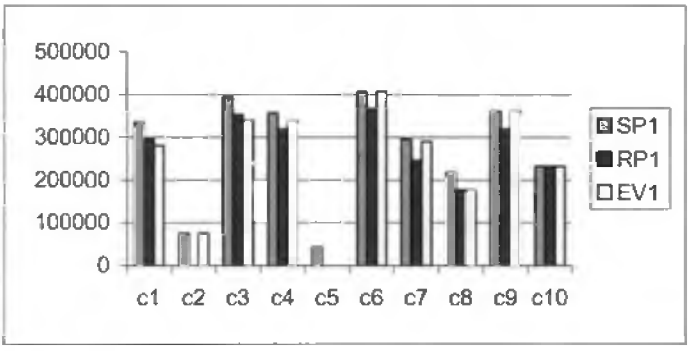


Figure 13.3. *Objective value solution of the best scenario.*

The stochastic solution value RP1 is slightly smaller than the average solution value EV1 for the best scenarios, but it is greater for the worst scenarios.

13.3 Recourse types in production planning under uncertainty

13.3.1 Problem statement

The planning of production capacity utilization is one of the most important managerial responsibilities in manufacturing. In particular, the problem consists of deciding how much production and how much product demand loss can be expected at each period along a time horizon. The production capacity constraints, the product stock limitations, some logistic constraints related to the production lot sizing, and the product demand requirements should be satisfied at a minimum cost. There is a vast literature on the deterministic version of the problem. See the seminal paper of [71] for considering only continuous variables. See [11, 25, 48, 58, 65, 66, 72], among others, for considering lot sizing limitations and other logical constraints (and, then, considering 0–1 variables).

However, very frequently production decisions must be made in the presence of uncertainty in several important parameters, such as product demand and resource availability along a multistage time horizon. It is well known (see [43]) that a tighter formulation can be obtained in the presence of linear constraints, by considering the product stock in an implicit way. In this case the production variable at a given time period is split into the production variables for satisfying the demand in each of its next periods along the time horizon.

We present below a tight modeling approach for production planning where the uncertainty is treated via scenario analysis, such that the occurrence of the events is represented by a multistage scenario tree; see Figure 13.1 and also [1]. Several alternative recourse types are considered, namely, full recourse, partial recourse for production, partial recourse for demand losing, and simple recourse.

13.3.2 Full recourse mixed 0–1 DEM

The following are additional notation for the sets and parameters used in the tactical production planning model.

Sets:

$\mathcal{J}$ = set of products;

$\mathcal{R}$ = set of resources.

Deterministic parameters:

$\widehat{N}$ = maximum number of products to be produced in a single time period;

$\underline{X}_{jt}, \overline{X}_j$ = conditional minimum and maximum volume of product j, if any, that can be produced at time period t and at any time period, respectively, for $j \in \mathcal{J}, t \in \mathcal{T}$;

$\overline{S}_j$ = maximum volume of product j that can be in stock at any time period for $j \in \mathcal{J}$;

o_{rj} = unit capacity consumption of resource r by product j for $r \in \mathcal{R}$, $j \in \mathcal{J}$;

h_j = unit holding cost of product j at any time period for $j \in \mathcal{J}$;

f_j = fixed cost to be incurred for producing product j at any time period for $j \in \mathcal{J}$.

Uncertain parameters and cost coefficients:

O_r^g = available capacity of resource r at time period $t(g)$ under scenario group g for $r \in \mathcal{R}, g \in \mathcal{G}$;

D_j^g = demand of product j under scenario group g for $j \in \mathcal{J}, g \in \mathcal{G}$;

c_j^g = unit processing cost of product j under scenario group g for $j \in \mathcal{J}, g \in \mathcal{G}$;

p_j^g = unit lost demand penalty for product j under scenario group g for $j \in \mathcal{J}, g \in \mathcal{G}$.

Variables:

$\delta_j^g =$ 0–1 variable such that its value is 1 if product j is produced under scenario group g, and 0 otherwise, for $j \in \mathcal{J}, g \in \mathcal{G}$;

$x_j^{g\tau} =$ production volume of product j at time period $t(g)$ under scenario group g to satisfy the demand from time period τ for $j \in \mathcal{J}$, $\tau \in \mathcal{T} : t(g) \leq \tau$, $g \in \mathcal{G}$; note that the production volume $x_j^{g\tau}$ will be in stock during the periods $t(g), t(g)+1, \ldots, \tau - 1$, for $t(g) < \tau$, independently of the scenario to occur;

$z_j^g =$ lost demand of product j from time period $t(g)$ under scenario group g for $j \in \mathcal{J}, g \in \mathcal{G}$.

The following is a compact representation of the DEM for the multistage stochastic problem with full recourse.

Objective

Determine the production and stock management policy to minimize the expected production and stock cost and the lost demand penalty plus the production fixed cost over the scenarios along the time horizon, subject to the constraints (13.31)–(13.38).

$$\min \sum_{g \in \mathcal{G}} w^g \sum_{j \in \mathcal{J}} \left[c_j^g \sum_{\tau \in \mathcal{T} : t(g) \leq \tau} x_j^{g\tau} + h_j \sum_{\ell \in \mathcal{N}^g \cup \{g\}} \sum_{\tau \in \mathcal{T} : t(g) < \tau} x_j^{\ell\tau} + p_j^g z_j^g + f_j \delta_j^g \right]. \quad (13.30)$$

The constraints are

$$\sum_{j \in \mathcal{J}} o_{rj} \sum_{\tau \in \mathcal{T} : t(g) \leq \tau} x_j^{g\tau} \leq O_r^g \quad \forall r \in \mathcal{R},\ g \in \mathcal{G}, \quad (13.31)$$

$$\underline{X}_{j,t(g)} \delta_j^g \leq \sum_{\tau \in \mathcal{T} : t(g) \leq \tau} x_j^{g\tau} \leq \overline{X}_j \delta_j^g \quad \forall j \in \mathcal{J},\ g \in \mathcal{G}, \quad (13.32)$$

$$\sum_{j \in \mathcal{J}} \delta_j^g \leq \widehat{N} \quad \forall g \in \mathcal{G}, \quad (13.33)$$

$$\sum_{\ell \in \mathcal{N}^g \cup \{g\}} \sum_{\tau \in \mathcal{T} : t(g) < \tau} x_j^{\ell\tau} \leq \overline{S}_j \quad \forall j \in \mathcal{J},\ g \in \mathcal{G}, \quad (13.34)$$

$$\sum_{\ell \in \mathcal{N}^g \cup \{g\}} x_j^{\ell, t(g)} + z_j^g = D_j^g \quad \forall j \in \mathcal{J},\ g \in \mathcal{G}, \quad (13.35)$$

$$z_j^g \geq 0 \quad \forall j \in \mathcal{J},\ g \in \mathcal{G}, \quad (13.36)$$

$$x_j^{g\tau} \geq 0 \quad \forall j \in \mathcal{J},\ \tau \in \mathcal{T} : t(g) \leq \tau,\ g \in \mathcal{G}, \quad (13.37)$$

$$\delta_j^g \in \{0, 1\} \quad \forall j \in \mathcal{J},\ g \in \mathcal{G}. \quad (13.38)$$

The knapsack constraints (13.31) ensure that the consumption of the resources does not exceed the availability. Constraints (13.32) define the semicontinuous character of the production volume. The cover-induced constraints (13.33) do not allow one to produce more products in a single time period than the set maximum. Constraints (13.34) force the upper bound on the product stock. Constraints (13.35) define the demand balance equations, such that the demand deficit is lost.

The instances of the mixed 0–1 DEM (13.30)–(13.38) can have such large dimensions that using state-of the-art optimization engines can make it unaffordable. Benders decomposition schemes can be used, although the instances' dimensions should be medium-size. See [14, 44] for the integer case, among others.

Alternatively, a splitting variable representation can be used by replacing the z-, x-, and δ-variables by their siblings such that z_j^g, $x_j^{g\tau}$, and δ_j^g are replaced by z_{jt}^{ω}, $x_{jt}^{\tau\omega}$, and δ_{jt}^{ω}, respectively, for $t = t(g)$, $\tau \in \mathcal{T} : t \leq \tau$, $\omega \in \Omega_g$, $g \in \mathcal{G}$. Additionally, the nonanticipativity constraints (13.39)–(13.41) are appended to the model for $\omega, \omega' \in \Omega_g$: $\omega \neq \omega'$, $j \in \mathcal{J}$, $g \in \mathcal{G}_t$, $t \in \mathcal{T}$.

$$z_{jt}^{\omega} - x_{jt}^{\omega'} = 0, \tag{13.39}$$

$$x_{jt}^{\tau\omega} - x_{jt}^{\tau\omega'} = 0 \quad \forall \tau \in \mathcal{T} : t \leq \tau, \tag{13.40}$$

$$\delta_{jt}^{\omega} - \delta_{jt}^{\omega'} = 0. \tag{13.41}$$

By dualizing the constraints (13.39)–(13.41), Lagrangian decomposition schemes can be used; see [42, 61, 62, 64], among others. However, some heuristics can be needed for obtaining feasible solutions for the independent-scenario-related mixed 0–1 models, such that the constraints (13.39)–(13.41) are also satisfied. With or without dualizing these constraints, a BFC scheme should be used in one way or another. Alonso-Ayuso, Escudero, and Ortuño [5] present a BFC scheme that handles two-stage mixed 0–1 DEM (where the first-stage constraints include only 0–1 variables) and multistage pure 0–1 DEM. A BFC scheme for multistage mixed 0–1 DEM is an open problem as far as we know.

13.3.3 Partial and simple recourse approaches

The model (13.30)–(13.38) obtains the production policy for the first stage by considering all scenarios but without being subordinated to any of them. However, in other environments it could be necessary to determine in advance the production policy for all or some other stages as well. In this case, only one production variable, say x_{jt}^{τ}, is considered for product j at time period t to satisfy the demand from period τ, $\tau \in \mathcal{T} : t \leq \tau$, and only one logistic variable, say δ_{jt}, is considered for each period. By replacing the related x- and δ-variables by the new variables in the model (13.30)–(13.38), we obtain a mixed 0–1 model with partial recourse. It strongly simplifies the model. However, if the anticipation of the production is not needed, this policy results in a stock increase under some scenarios to satisfy the related demand or, alternatively, it results in a lost demand increase.

On the other hand, by anticipating the z-variables, we may have recourse on the x- and δ-variables. In this case, the meaning of the z-variables could be more related to outsourcing production than demand loss.

Finally, we may have an anticipative policy for both types of production, namely, in-house and outsourcing. In this case we have a simple-recourse-based model, where there is no recourse on the x-, z-, and δ-variables. The only action left available to the decision maker is to build stock to hedge the uncertainty, and it could be expensive. See [33] for the linear case, where the stock is explicitly modeled, and then, the variable $x_j^{g\tau}$ is replaced by the variable x_{jt} for $t = t(g)$ (production volume of product j at time period t) and the stock variable s_j^g is added.

13.4 Tactical multilevel supply chain management

13.4.1 Problem statement

A global multinational player (e.g., in the computer and automation sectors) would ideally like to take business decisions which span sourcing, manufacturing, assembly, and distribution. Thus, a company with multiple suppliers at different production levels and multiple markets may seek to allocate demand quantities to different plants over a given time horizon. Its objective can be to determine the production, procurement, and stock policy that best utilize the available resources in the whole supply chain system.

The problem elements are as follows. An end product is the final output of the manufacturing network. A subassembly is any part number that is assembled by the manufacturing network and is used to assemble another part number. The term product refers to both end products and subassemblies. The term component describes any part number that is required for the production. The BoM of a product is the structuring of the set of components that are required for its manufacturing/assembly; see Figure 13.4. Let us name as the first tier component of a product any component that is directly required for the production of the product. As an example, the components 11, 12, and 14 are the first tier components of product 8 in Figure 13.4. We will name as raw material any component whose BoM is not a concern of the decision-making process. The cycle time of a product is the set of consecutive and integer time periods that are required for its completion from its release in the assembly line until its availability for use. A production period is a time period in the product's cycle time. Define the backlog of a product at the end of a time period as the (nonnegative) difference between its accumulated demand and shipment up to that period. Multiple market sources for end products are allowed. Escudero [30] presents an LP modeling approach for the deterministic case. See also [19, 65].

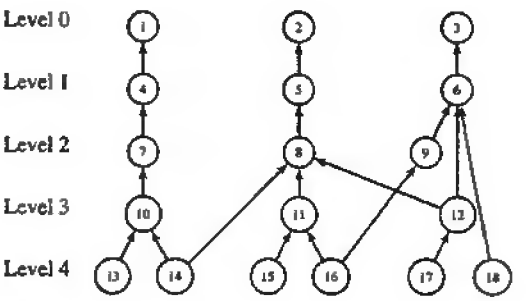


Figure 13.4. *Bill of material (BoM).*

However, very frequently important parameters are uncertain at the planning time period, although they are known at the occurrence of the scenarios. These parameters are the product demand and the lost demand fraction, the resource availability, the unit production cost, and the raw material supply cost. Escudero et al. [33] present a two-stage stochastic LP model that also includes some other features, such as different modes for component procurement, say, standard and expediting modes, effective period segments where the components are required in the BoM (a very interesting structure for modeling

engineering changes), alternate components of the so-called prime components, raw material and product groups, etc.

One of the important decisions to be made in supply chain planning consists of determining the ordering time period for raw material supplying and product manufacturing/assembling along the time horizon. There is usually a time interval between the ordering and the delivering of the components in the supply chain. In the case that the interval is not subject to specific constraints, a deterministic model can consider the ordering time to be the same as the delivering time. However, in the stochastic setting the related time interval is important, since the production and market environments can vary along the interval. In this section we present an LP DEM for the multistage stochastic problem with significant time interval between the component ordering and delivering times.

13.4.2 LP DEM

The following is additional notation for the sets and parameters used in the tactical supply chain model.

Sets:

$\mathcal{J}$ = set of products;

$\mathcal{JE}$ = set of end products ($\mathcal{JE} \subseteq \mathcal{J}$);

$\mathcal{JS}$ = set of subassemblies; $\mathcal{J} = \mathcal{JE} \bigcup \mathcal{JS}$;

$\mathcal{DS}_j$ = set of market sources for end product j for $j \in \mathcal{JE}$; it is assumed that no subassembly has external demand, but the assumption can be easily removed;

$\mathcal{I}$ = set of components;

$\mathcal{IR}$ = set of raw materials ($\mathcal{IR} \subseteq \mathcal{I}$); note that $\mathcal{I} = \mathcal{JS} \bigcup \mathcal{IR}$;

$\mathcal{I}_j$ = set of first tier components for product j for $j \in \mathcal{J}$;

$\mathcal{R}$ = set of resources;

$\mathcal{DS} = \bigcup_{j \in \mathcal{J}} \mathcal{DS}_j$.

Deterministic parameters:
Market

$\overline{B}_{dt}$ = maximum backlog from market source d that is allowed at time period t for $d \in \mathcal{DS}, t \in \mathcal{T}$;

τ_d = delivery lag time, i.e., number of time periods after its completion to deliver the related product to market source d, for $d \in \mathcal{DS}$;

ρ_{dt} = unit lost demand penalty for market source d at time period t for $d \in \mathcal{DS}, t \in \mathcal{T}$.

BoM

c_j = cycle time of product j for $j \in \mathcal{J}$;

p_{ij} = production time period in the cycle time of product j, where first tier component i is needed, for $i \in \mathcal{I}_j$, $j \in \mathcal{J}$;

a_{ij} = volume of first tier component i that is needed per unit of product j for $i \in \mathcal{I}_j$, $j \in \mathcal{J}$;

τ_{ij} = number of time periods required to deliver component i from its depot to the plant where product j is manufactured/assembled for $i \in \mathcal{I}_j$, $j \in \mathcal{J}$.

Component availability

$\overline{X}_{it}$ = maximum volume of raw material i that can be ordered at time period t for $i \in \mathcal{IR}, t \in \mathcal{T}$;

τ^i = number of time periods required to supply raw material i to its depot for $i \in \mathcal{IR}$.

Production and stock restrictions

$\overline{Z}_{jt}$ = maximum released volume that is allowed for product j at time period t for $j \in \mathcal{J}, t \in \mathcal{T}$;

$\underline{S}_{jt}, \overline{S}_j$ = minimum and maximum volume of product or raw material j that can be in stock at time period t and at any time period, respectively, for $j \in \mathcal{J} \bigcup \mathcal{IR}$, $t \in \mathcal{T}$;

o_{rj} = unit capacity consumption of resource r by product j for $r \in \mathcal{R}$, $j \in \mathcal{J}$. It is assumed that the resource is required at the release time of the product.

Cost coefficients

h_j = unit holding cost of product or raw material j at any time period for $j \in \mathcal{J} \bigcup \mathcal{IR}$.

Uncertain parameters and cost coefficients:

D_d^g = demand from market source d at time period $t(g)$ under scenario group g for $d \in \mathcal{DS}, g \in \mathcal{G}$;

f_d^g = lost fraction of nonserved accumulated demand from market source d under scenario group g for $d \in \mathcal{DS}, g \in \mathcal{G}$;

O_r^g = available capacity of resource r under scenario group g for $r \in \mathcal{R}$, $g \in \mathcal{G}$;

pc_j^g = unit production cost for product j under scenario group g for $j \in \mathcal{J}$, $g \in \mathcal{G}$;

sc_i^g = unit supply cost for raw material i under scenario group g for $i \in \mathcal{IR}$, $g \in \mathcal{G}$.

Variables:

z_j^g = volume of product j that is released in the production line at (the beginning of) time period $t(g)$ under scenario group g for $j \in \mathcal{J}$, $g \in \mathcal{G}$;

x_i^g = volume of raw material i that is *ordered* under scenario group g for $i \in \mathcal{IR}$, $g \in \mathcal{G}$;

y_d^g = volume of served demand from market source d that is being shipped under scenario group g for $d \in \mathcal{DS}$, $g \in \mathcal{G}$;

s_j^g = stock volume of product or raw material j under scenario group g for $j \in \mathcal{J} \bigcup \mathcal{IR}$, $g \in \mathcal{G}$;

b_d^g = backlog volume from market source d under scenario group g for $d \in \mathcal{DS}$, $g \in \mathcal{G}$.

The following is a compact representation of the DEM for the multistage stochastic problem.

Objective

Determine the master production planning to minimize the expected production, supply, and stock cost plus the expected penalties due to demand loss and backlogging over the scenarios along the time horizon, subject to the constraints (13.44)–(13.55).

$$\min \sum_{g \in \mathcal{G}} w^g \left[\sum_{j \in \mathcal{J}} pc_j^g z_j^g + \sum_{i \in \mathcal{IR}} sc_i^g x_i^g + \sum_{j \in \mathcal{J} \bigcup \mathcal{IR}} h_j s_j^g + \sum_{d \in \mathcal{DS}} (\rho_{d,t(g)} \ell_d^g + \sigma_{d,t(g)} b_d^g) \right], \tag{13.42}$$

where ℓ_d^g gives the lost demand from market source d at time period $t(g)$ under scenario g, is

$$\ell_d^g \equiv f_d^g (b_d^k + D_d^g - y_d^e) \geq 0, \tag{13.43}$$

where $k \in \mathcal{N}^g : t(k) = t(g) - 1$ and $e \in \mathcal{N}^g : t(e) = t(g) - \tau_d$. Note that k is the scenario group whose associated node in the scenario tree is the predecessor of the related node for scenario group g. So, b_d^k gives the backlog of the previous time period to $t(g)$ under scenario group k. Similarly, y_d^e gives the served demand that is shipped at time period $t(e)$ (under scenario group e) to satisfy the product demand from market source d at time period $t(g)$ (under any scenario in group $G_{t(g)}$). For the remainder of the section, the parameters k and e will have the meaning given here.

The constraints are

$$s_j^k + z_j^n = \sum_{d \in \mathcal{DS}_j} y_d^g + s_j^g \quad \forall j \in \mathcal{JE},\ g \in \mathcal{G},\ k \in \mathcal{N}^g, \tag{13.44}$$

$$\text{where } n \in \mathcal{N}^g : t(n) = t(g) - c_j + 1, \tag{13.45}$$

$$s_i^k + \gamma_i^n = \sum_{j \in \mathcal{J} : i \in \mathcal{I}_j} a_{ij} z_j^h + s_i^g \quad \forall i \in \mathcal{I},\ g \in \mathcal{G}, \tag{13.46}$$

$$\text{where } \gamma_i^n \equiv \begin{cases} z_i^n & \text{for } i \in \mathcal{JS}, \text{ where } n \text{ is as in (13.45)}, \\ x_i^n & \text{for } i \in \mathcal{IR}, \text{ where } n \in \mathcal{N}^g : t(n) = t(g) - \tau^i, \end{cases}$$

$$h \in \mathcal{N}^g \cup \{g\} : t(h) = t(g) + \tau_{ij} - p_{ij} + 2, \tag{13.47}$$

$$s_i^k \geq \sum_{j \in \mathcal{J} : i \in \mathcal{I}_j} a_{ij} z_j^h \quad \forall i \in \mathcal{I},\ g \in \mathcal{G}, \tag{13.48}$$

where h is as in (13.47),

$$b_d^g + (1 - f_d^g) y_d^e - (1 - f_d^g) b_d^k = (1 - f_d^g) D_d^g \quad \forall d \in \mathcal{DS},\ g \in \mathcal{G}, \tag{13.49}$$

$$y_d^e - b_d^k \leq D_d^g \quad \forall d \in \mathcal{DS},\ g \in \mathcal{G}, \tag{13.50}$$

$$\sum_{j \in \mathcal{J}} o_{rj} z_j^g \leq O_r^g \quad \forall r \in \mathcal{R},\ g \in \mathcal{G}, \tag{13.51}$$

$$0 \leq z_j^g \leq \overline{Z}_{j,t(g)} \quad \forall j \in \mathcal{J},\ g \in \mathcal{G}, \tag{13.52}$$

$$\underline{S}_{j,t(g)} \leq s_j^g \leq \overline{S}_j \quad \forall j \in \mathcal{J} \cup \mathcal{IR},\ g \in \mathcal{G}, \tag{13.53}$$

$$0 \leq x_i^g \leq \overline{X}_{i,t(g)} \quad \forall i \in \mathcal{IR},\ g \in \mathcal{G}, \tag{13.54}$$

$$0 \leq b_d^g \leq \overline{B}_d^g \quad \forall d \in \mathcal{DS},\ g \in \mathcal{G}. \tag{13.55}$$

Constraints (13.44) and (13.46) are the stock balance equations for the end products and the components, respectively. Constraints (13.48) allow the synchronization of the production release. The constraints (13.46) and (13.48) refer to the pairs given by the components and the scenarios. Note that $t(h)$ gives the period for the release of the product, say, j; see (13.47). The scenario group h either belongs to the predecessor path $\mathcal{N}^g$ or $h = g$.

The lost demand variable $\ell_d^g \geq 0$ of (13.43) is not explicitly included in the model. So, the constraint system given by the product demand balance equation

$$y_d^e + \ell_d^g + b_d^g = b_d^k + D_d^g \quad \forall d \in \mathcal{DS},\ g \in \mathcal{G}, \tag{13.56}$$

and the nonnegativity constraint of the variable has been replaced by the system (13.49)–(13.50).

Constraints (13.51) ensure that the consumption of the resources (to be used at the release time of the products) does not exceed their availability. Finally, the system (13.52)–(13.55) bounds the variables.

The instances of the compact representation (13.42)–(13.55) can have such big dimensions that decomposition approaches are needed. For illustrative purposes use the dimensions of a real instance from the automation sector: $|\mathcal{T}| = 13$ time periods, $|\mathcal{JE}| = 23$ end products, $|\mathcal{JS}| = 104$ subassemblies, $|\mathcal{IR}| = 5821$ raw components, and $|\mathcal{DS}| = 525$ market sources. The related dimensions of the compact DEM for the two-stage stochastic version with 2 periods in the first stage, 11 periods in the second stage, and $|\Omega| = 100$ scenarios are 2,893,683 constraints, 6,014,547 variables, and 83,304,251 nonzero constraint elements. Benders' decomposition schemes can be used; see [13, 21, 36], among others.

The compact representation (13.42)–(13.55) can be transformed into a splitting variable representation such that the variable, say x_i^g, is replaced by its sibling, say x_{it}^ω for $t = t(g)$, $\omega \in \Omega_g$, etc. Additionally, the nonanticipativity constraints (13.57)–(13.61) are

appended to the model for $\omega, \omega' \in \Omega_g : \omega \neq \omega', g \in \mathcal{G}_t, t \in \mathcal{T}$.

$$x_{it}^{\omega} - x_{it}^{\omega'} = 0 \qquad \forall i \in \mathcal{IR}, \tag{13.57}$$

$$z_{jt}^{\omega} - z_{jt}^{\omega'} = 0 \qquad \forall j \in \mathcal{J}, \tag{13.58}$$

$$s_{jt}^{\omega} - s_{jt}^{\omega'} = 0 \qquad \forall j \in \mathcal{J} \cup \mathcal{IR}, \tag{13.59}$$

$$y_{dt}^{\omega} - y_{dt}^{\omega'} = 0 \qquad \forall d \in \mathcal{DS}, \tag{13.60}$$

$$b_{dt}^{\omega} - b_{dt}^{\omega'} = 0 \qquad \forall d \in \mathcal{DS}. \tag{13.61}$$

The splitting variable representation is very amenable to using Benders decomposition as well as augmented Lagrangian decomposition; see [31, 53].

13.5 Parallel unrelated machines investment and assignment

13.5.1 Problem statement

Consider a type of combinatorial problem that arises in the production of printed circuits in the semiconductor industry, but it has enough generality to be found in other sectors as well. In any case, its structure can be embedded in more general settings. The problem consists of a single-phase manufacturing process with parallel unrelated machines with setups (PUMS). In such a system, different machines are used to perform a single operation on a number of different jobs. The processing time for a job depends on both the job and the machine. In addition, each machine requires significant setup time between processing different job types.

See in [23] a broader description of the deterministic version of the problem together with a heuristic algorithm for minimizing the makespan. Dietrich and Escudero [24] present two equivalent representations of a 0–1 model with a minmax function. Several types of cuts to tighten further the tighter model are also presented.

The stochastic version of the problem is as follows. Assume a set of machines for selection, whose availability is an uncertain parameter that is only known with certainty at the production time. Assume also that an uncertain set of jobs can be available for processing in the machines. The given set is not known with certainty at the machine selection time, but obviously it will be known at the production time. However, the job processing time is a stochastic parameter that will be known only when the actual work is in progress. Given the (known) setup time required by changing the job type to be processed in a machine, the problem consists of determining (1) the set of machines to select and (2) the set of jobs to be processed in each selected machine. Two applications cases are considered. The first is a machine selection problem and the second is a machine assignment problem. In both of them the following constraints should be satisfied:

1. A job must be processed by one and only one machine, i.e., job splitting is not allowed.
2. A machine can work on only one job at a time.
3. Job preemption is not allowed.

4. The total time a machine will be busy (setup time plus processing time) cannot exceed the available time.

It is assumed that jobs that belong to the same type are processed sequentially.

13.5.2 Machine selection. Pure 0–1 DEM

The first problem to be addressed consists of determining the set of machines to select for job processing such that the investment depreciation cost is minimized, the first-stage constraints (related to machine compatibility) are satisfied, and the second-stage constraints (related to job processing) are also satisfied over the scenarios.

The following is additional notation for the sets and parameters used in the machine selection model.

Sets:

$\mathcal{I}$ = set of candidate machines;

$\mathcal{J}^\omega$ = set of jobs to be processed under scenario ω for $\omega \in \Omega$; note that $\mathcal{J} = \bigcup_{\omega\in\Omega} \mathcal{J}^\omega$ is the set of potential jobs to be processed;

$\mathcal{I}_j$ = set of machines that can process job j for $j \in \mathcal{J}$;

$\mathcal{J}_i$ = set of jobs that can be processed by machine i for $i \in \mathcal{I}$; note that $j \in \mathcal{J}_i \Leftrightarrow i \in \mathcal{I}_j$;

$m(j)$ = type of job j for $j \in \mathcal{J}$;

$\mathcal{M}^\omega$ = set of job types to be processed under scenario ω for $\omega \in \Omega$; note that $m \in \mathcal{M}^\omega \Leftrightarrow \exists j \in \mathcal{J}^\omega : m(j) = m$;

$\mathcal{M} = \bigcup_{\omega\in\Omega} \mathcal{M}^\omega$ = set of potential job types to be processed;

$\mathcal{I}^m$ = set of machines where job type m can be processed for $m \in \mathcal{M}$; note that $i \in \mathcal{I}^m \Leftrightarrow \exists j \in \mathcal{J}_i : m(j) = m$;

$\mathcal{M}_i$ = set of job types that can be processed by machine i for $i \in \mathcal{I}$; note that $m \in \mathcal{M}_i \Leftrightarrow i \in \mathcal{I}^m$.

Technical and logistic parameters:

$\widetilde{N}$ = maximum number of machines that are allowed;

O_i = available time (i.e., capacity) of machine i for $i \in \mathcal{I}$;

P = available budget for machine investment;

d_{im} = setup (i.e., preparation time) required by machine i when processing a job of type m after processing a job of a different type;

Deterministic cost coefficients:

$a_i =$ budget required for selecting machine i for $i \in \mathcal{I}$;

$q_i =$ depreciation cost of the investment a_i in machine i for $i \in \mathcal{I}$;

$b_{im} =$ setup cost of machine i for processing job type m for $i \in \mathcal{I}^m$, $m \in \mathcal{M}$.

Uncertain parameters:

$\rho_i^\omega =$ 0–1 indicator about the availability of machine i under scenario ω, such that its value 1 means that the machine is available, and 0 otherwise, for $i \in \mathcal{I}$, $\omega \in \Omega$;

$p_{ij}^\omega =$ processing time of job j in machine i under scenario ω for $i \in \mathcal{I}_j$, $j \in \mathcal{J}^\omega$, $\omega \in \Omega$;

$c_{ij}^\omega =$ cost of processing job j in machine i under scenario ω for $i \in \mathcal{I}_j$, $j \in \mathcal{J}$, $\omega \in \Omega$. Usually, $c_{ij}^\omega = f(p_{ij}^\omega)$.

Strategic variables

These are 0–1 variables, such that

$\gamma_i = 1$ if machine i is selected, and 0 otherwise, for $i \in \mathcal{I}$.

Operation variables

These are 0–1 variables for each product j and scenario ω, for $j \in \mathcal{J}^\omega$, $\omega \in \Omega$ such that

$x_{ij}^\omega = 1$ if job j is processed in machine i under scenario ω, and 0 otherwise, for $i \in \mathcal{I}_j$;

$y_{im}^\omega = 1$ if job type m is processed in machine i under scenario ω, and 0 otherwise, for $i \in \mathcal{I}^m$, $m \in \mathcal{M}^\omega$.

The following is a compact representation of the DEM for the two-stage stochastic problem.

Objective

Select the set of machines to minimize its investment depreciation cost plus the expected setup and processing cost over the scenarios, subject to the constraints (13.63)–(13.69).

$$\min \sum_{i \in \mathcal{I}} q_i \gamma_i + \sum_{\omega \in \Omega} w_\omega \left[\sum_{j \in \mathcal{J}^\omega} \sum_{i \in \mathcal{I}_j} c_{ij}^\omega x_{ij}^\omega + \sum_{m \in \mathcal{M}^\omega} \sum_{i \in \mathcal{I}^m} b_{im} y_{im}^\omega \right]. \tag{13.62}$$

The Stage 1 (strategic) constraints are

$$\sum_{i \in \mathcal{I}} \gamma_i \leq \widetilde{N}, \tag{13.63}$$

$$\sum_{i \in \mathcal{I}} a_i \gamma_i \leq P, \tag{13.64}$$

$$\gamma_i \in \{0,1\} \quad \forall i \in \mathcal{I}. \tag{13.65}$$

The cover-induced constraint (13.63) ensures that the number of selected machines does not exceed the allowed maximum. The knapsack constraint (13.64) takes into account the investment budget.

The Stage 2 (assignment) constraints for each scenario $\omega \in \Omega$ are

$$\sum_{i \in \mathcal{I}_j} x_{ij}^{\omega} = 1 \quad \forall j \in \mathcal{J}^{\omega}, \tag{13.66}$$

$$x_{ij}^{\omega} \leq y_{i,m(j)}^{\omega} \quad \forall i \in \mathcal{I}_j,\ j \in \mathcal{J}^{\omega}, \tag{13.67}$$

$$\sum_{j \in \mathcal{J}^{\omega} \cap \mathcal{J}_i} p_{ij}^{\omega} x_{ij}^{\omega} + \sum_{m \in \mathcal{M}^{\omega} \cap \mathcal{M}_i} d_{im} y_{im}^{\omega} \leq O_i \rho_i^{\omega} \quad \forall i \in \mathcal{I}, \tag{13.68}$$

$$x_{ij}^{\omega} \in \{0,1\} \quad \forall i \in \mathcal{I}_j,\ j \in \mathcal{J}^{\omega}. \tag{13.69}$$

The special constraint sets (13.66) force each job to be processed by only one machine. The VUB constraints (13.67) force the variable y_{im}^{ω} to have value 1 if any of the jobs of type m is to be processed by machine i. The knapsack constraints (13.68) prevent using more time (capacity) at each machine than the amount available there. Given the type of constraints and objective function, the y-variables automatically take 0–1 values in the optimal solution.

In a manner similar to that of the other application cases, the compact representation (13.62)–(13.69) can be transformed into a splitting variable representation such that the γ-variables are replaced by their siblings, say the γ^{ω}-variables, for $\omega \in \Omega$. Additionally, the nonanticipativity constraints (13.70) are appended to the model for $\omega, \omega' \in \Omega : \omega \neq \omega'$.

$$\gamma_i^{\omega} - \gamma_i^{\omega'} = 0 \qquad \forall i \in \mathcal{I}. \tag{13.70}$$

13.5.3 Machine assignment. Pure 0–1 DEM

The second problem to be addressed consists of assigning a given set of jobs to machines such that the expected maximum workload (i.e., the makespan) is minimized. The DEM is a mixed 0–1 model for a given set $\mathcal{J}$ of jobs to be processed.

The uncertain parameter is the processing time of the jobs in the machines. This uncertainty is also treated via scenario analysis, but the scenario (i.e., the processing time realization) occurs after the assignment is performed (i.e., it occurs during the job processing). Thus, the assignment must consider all scenarios, without being subordinated to any of them. For this purpose a simple recourse model is used. The notation for the sets and parameters is as above.

Variables:

x_{ij} = 0–1 variable such that its value is 1 if job j is processed in machine i, and 0 otherwise, for $i \in \mathcal{I}_j$, $j \in \mathcal{J}$;

y_{im} = 0–1 variable such that its value is 1 if job type m is processed in machine i, and 0 otherwise, for $i \in \mathcal{I}^m$, $m \in \mathcal{M}$;

z_i^{ω} = workload of machine i under scenario ω for $i \in \mathcal{I}$, $\omega \in \Omega$;

z^{ω} = maximum workload under scenario ω for $\omega \in \Omega$;

u^ω = over workload vector in the machines under scenario ω, for $\omega \in \Omega$. Note that a positive element, say u_i^ω of the vector u^ω, gives the time capacity constraint violation for machine i for $i \in \mathcal{I}$.

The following is a compact representation of the DEM for the stochastic problem with simple recourse.

Objective

Determine the machine assignment for job processing to minimize the expected makespan over the scenarios plus a penalty function on the machine time capacity constraint violation, subject to the constraints (13.72)–(13.78),

$$\min \sum_{\omega \in \Omega} w_\omega \left[z^\omega + H^\omega(u^\omega) \right], \tag{13.71}$$

where $H^\omega(\cdot)$ is a constraint infeasibility function; see, e.g., [30] for several types of linear and nonlinear H-functions.

The constraints are

$$\sum_{i \in \mathcal{I}_j} x_{ij} = 1 \quad \forall j \in \mathcal{J}, \tag{13.72}$$

$$x_{ij} \le y_{i,m(j)} \quad \forall i \in \mathcal{I}_j,\ j \in \mathcal{J}, \tag{13.73}$$

$$\sum_{j \in \mathcal{J}_i} p_{ij}^\omega x_{ij} + \sum_{m \in \mathcal{M}_i} d_{im} y_{im} = z_i^\omega + u_i^\omega \quad \forall i \in \mathcal{I},\ \omega \in \Omega, \tag{13.74}$$

$$z_i^\omega \le O_i \quad \forall i \in \mathcal{I},\ \omega \in \Omega, \tag{13.75}$$

$$z_i^\omega \le z^\omega \quad \forall i \in \mathcal{I},\ \omega \in \Omega, \tag{13.76}$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in \mathcal{I}_j,\ j \in \mathcal{J}, \tag{13.77}$$

$$u_i^\omega \ge 0 \quad \forall i \in \mathcal{I},\ \omega \in \Omega. \tag{13.78}$$

Constraints (13.72) and (13.73) are as (13.66) and (13.67), respectively, for a given set $\mathcal{J}$ of jobs. Constraints (13.74) and (13.75) define the workload of the machines as well as the workload capacity violation for the scenarios. The function (13.71) together with the constraints (13.76) force z^ω to be the workload of the busiest machine (i.e., the makespan) under scenario ω.

Alternatively to the minimization of the expected makespan $\sum_{\omega \in \Omega} w_\omega z^\omega$, the objective could be the minimization of the makespan under any scenario, i.e., $\min \max_{\omega \in \Omega} \{z^\omega\}$.

Another objective function replaces the makespan by the expected setup and processing costs, in a manner similar to that of the objective of the model presented in the previous section. The model is a pure 0–1 DEM.

13.6 Production sequencing and scheduling under uncertainty

13.6.1 Introduction

Scheduling problems arise in many practical circumstances when planning the utilization of a production or manufacturing system. Many are basically optimization problems having

the following form: given a set of operations to be executed along a time horizon, subject to various constraints, find a feasible schedule that minimizes the value of a given objective function. Typical elements are limited availability of the resources, multiperiod operations, subsets of jobs with exclusivity constraints, and precedence relationships in the execution of the operations.

This type of problem can be formulated as 0–1 models and falls into the category of $\mathcal{NP}$-hard problems. Thus, traditional branch-and-bound methods have proved to be very inefficient for solving them. Instead, heuristic and metaheuristic approaches have been found to obtain satisfactory solutions for special classes of these problems, such as problems with single-period operations and special objective functions (e.g., makespan minimization). See [8, 38, 57], among others.

On the other hand, there is a vast literature on polyhedral analysis of the problem and, then, on tight 0–1 formulations and facet defining inequalities identification; see, e.g., [66, 72, 73], among others.

However, very frequently the resource availability as well as the resource consumption by the operations' execution and, as a consequence, their execution costs are uncertain parameters. Then, it is a very interesting application case of stochastic programming.

We start with a problem description whose deterministic version is broadly given in [28]. Next, the uncertain parameters will be discussed and the 0–1 DEM for the multistage stochastic problem with full recourse will be presented. Finally, some ideas about problem solving will be presented.

13.6.2 Problem statement

Assume a set of jobs, each of which includes a set of operations to be executed along a given time horizon. Each operation has a time window for its execution. The operations must be executed during a given number of (consecutive and integer) so-called production time periods without preemption. Some jobs are alternative in the sense that one and only one of these jobs can be executed. Let us call a set of alternative jobs a class. If a job is executed, the operations of the other jobs that belong to the same class cannot be executed (i.e., they cannot be assigned).

It is assumed that each operation has assigned a single dedicated machine (or working station) for its execution. Let us say that the operations with the same dedicated machine belong to the same type, such that the simultaneous execution of the operations is not allowed. A setup in a dedicated machine is required between the consecutive execution of two operations.

There are precedence relationships in the execution of the operations. They can be expressed by a directed acyclic graph, where the nodes are associated with the operations and the arcs refer to the existence of precedences between the execution of the operations represented by the from-nodes and the to-nodes of the arcs. Two types of precedences are considered, such that a minimum number (type 1) and a maximum number (type 2) of time periods are required between the beginning of the executions.

A set of resources with uncertain availability along the time horizon is allowed. The operation execution can require resource consumption in each of its production periods. The resource amount to be utilized depends on several factors and is also an uncertain parameter. Although the resource availability is uncertain at the planning time period, it is

known at the period where the resource is required. However, the resource consumption by the operations execution is known only at the consumption real time, which means that at a given time period the occurrence of the resource consumption scenario is not known in advance.

The goal consists of determining the time period at which each operation will begin its assignment, if any, such that a set of constraints is satisfied at a minimum expected execution cost over the scenarios.

Application cases of the sequencing and scheduling problem can be found in investment planning selection (see [35]) and production units maintenance planning (see [27], among others), besides the proper application in production and manufacturing (see [28, 66, 72, 73], among others). All of these papers consider only the deterministic version.

13.6.3 Pure 0-1 DEM

The following is additional notation for the sets and parameters to be used in the production sequencing and scheduling model.

Sets:

$\mathcal{J}$ = set of products;

$\mathcal{R}$ = set of resources;

$\mathcal{F}$ = set of potential scenarios for the resource consumption by the operations;

$\mathcal{I}$ = set of operations;

$\mathcal{C}$ = set of classes of jobs;

$\mathcal{T}_i$ = set of feasible periods to begin the execution of operation i for $i \in \mathcal{I}$ ($\mathcal{T}_i \subseteq \mathcal{T}$);

$\mathcal{I}_j$ = set of operations included in job j for $j \in \mathcal{J}$ ($\mathcal{I}_j \subseteq \mathcal{I}$);

$\mathcal{J}_c$ = set of jobs that belong to class c for $c \in \mathcal{C}$ ($\mathcal{J}_c \subseteq \mathcal{J}$);

$\mathcal{M}$ = set of types of operations;

$\mathcal{I}^m$ = set of operations that belong to type m for $m \in \mathcal{M}$ ($\mathcal{I}^m \subseteq \mathcal{I}$);

$\mathcal{A}^1$ (respectively, $\mathcal{A}^2$) = set of ordered pairs of operations with precedence relationship type 1 (respectively, type 2).

Deterministic parameters:

e_i, l_i = earliest and latest release dates, respectively, for beginning the execution of operation i for $i \in \mathcal{I}$; note that $\mathcal{T}_i \subseteq \{e_i, e_{i+1}, \ldots, l_i\}$;

d_i = number of time periods for the execution of operation i for $i \in \mathcal{I}$; note that $t \in \mathcal{T}_i$ implies that $1 \leq t \leq |\mathcal{T}| - d_i + 1$;

d^m = setup time between the ending and the beginning of the execution of two consecutive operations that belong to type m for $m \in \mathcal{M}$;

p_{ab} = minimum (respectively, maximum) number of time periods between the beginning of the execution of the operations a and b for $(a, b) \in \mathcal{A}^1$ (respectively, $(a, b) \in \mathcal{A}^2$).

Note 1: It is required that $\exists j \in \mathcal{J} : a, b \in \mathcal{I}_j$, and so $(a, b) \notin \mathcal{A}^1 \bigcup \mathcal{A}^2$ for $a \in \mathcal{I}_j$ and $b \in \mathcal{I}_{j'}$, $j, j' \in \mathcal{J} : j \neq j'$.

Note 2: $e_b \geq e_a + p_{ab}$ for $(a, b) \in \mathcal{A}^1$, $l_a \leq l_b - p_{ab}$ for $(a, b) \in \mathcal{A}^1$, $l_b \leq l_a + p_{ab}$ for $(a, b) \in \mathcal{A}^2$.

Uncertain parameters and cost coefficients:

w_f^g = weight factor assigned to the mixture of resource availability scenario group g and resource consumption scenario f for $g \in \mathcal{G}$, $f \in \mathcal{F}$;

o_{rhi}^f = amount of resource r that is required by operation i during its hth production time period under the resource consumption scenario f for $r \in \mathcal{R}$, $h = 1, 2, \ldots, d_i$, $i \in \mathcal{I}$, $f \in \mathcal{F}$; note that the resource consumption is an uncertain parameter that is known only during the time period where it is used but not at the beginning of the operation's execution;

O_r^g = available capacity of resource r at time period $t(g)$ under scenario group g for $r \in \mathcal{R}$, $g \in \mathcal{G}$;

c_i^{gf} = expected execution cost of operation i under the mixture of resource availability scenario group g and resource consumption scenario f for $i \in \mathcal{I}$, $g \in \mathcal{G}$, $f \in \mathcal{F}$. Note that the scenario group g and its associated time period $t(g)$ both influence the cost of the resource consumption as well as the cost of some other operation execution items.

Strategic variables

These are 0–1 variables such that

$y_j = 1$ if job j is selected for execution, and 0 otherwise, for $j \in \mathcal{J}$.

Sequencing and scheduling variables

These are 0–1 variables such that

$x_i^g = 1$ if operation i begins its execution at time period $t(g)$ under scenario group g, and 0 otherwise, for $g \in \mathcal{G}_t$, $t \in \mathcal{T}_i$, $i \in \mathcal{I}$.

The execution time interval for operation i is $t(g), t(g) + 1, \ldots, t(g) + d_i - 1$ for $x_i^g = 1$.

The following is a compact representation of the DEM for the multistage stochastic problem with a mixture of full recourse and simple recourse.

Objective

Determine the operations sequencing and scheduling to minimize the expected cost of the operations execution over the scenarios along a time horizon, subject to the constraints (13.81)–(13.88).

$$\min \sum_{g \in \mathcal{G}} \sum_{f \in \mathcal{F}} w_f^g \sum_{i \in \mathcal{I}} c_i^{gf} x_i^g. \tag{13.79}$$

Alternatively, the objective could be minimizing the expected makespan, among other objectives,

$$\min \left\{ \max_{g \in \mathcal{G}_t : t \in \mathcal{T}_a} \{(t(g) + d_a - 1) x_a^g \ \forall a \in \mathcal{I} : \nexists b \in \mathcal{I} : (a, b) \in \mathcal{A}^1\} \right\}. \tag{13.80}$$

The constraints are

$$\sum_{j \in \mathcal{J}_c} y_j = 1 \quad \forall c \in \mathcal{C}, \tag{13.81}$$

$$\sum_{g \in \mathcal{G}_t : t \in \mathcal{T}_i} x_i^g = y_j \quad \forall i \in \mathcal{I}_j,\ j \in \mathcal{J}, \tag{13.82}$$

$$\sum_{i \in \mathcal{I}^m} \sum_{k \in F_i^1} x_i^k \leq 1 \quad \forall m \in \mathcal{M},\ g \in \mathcal{G}_t,\ t \in \mathcal{T}, \tag{13.83}$$

$$\text{where } F_i^1 \equiv \left\{k \in \mathcal{N}^g \cup \{g\} : t(k) \in \mathcal{T}_i, t - d_i - d^m + 1 \leq t(k) \leq t\right\},$$

$$\sum_{k \in F_a^2} x_a^k \geq \sum_{k \in F_b^3} x_b^k \quad \forall g \in \mathcal{G}_t,\ t \in \mathcal{T}_b,\ (a, b) \in \mathcal{A}^1, \tag{13.84}$$

$$\text{where } F_a^2 \equiv \left\{k \in \mathcal{N}^g \cup \{g\} : t(k) \in \mathcal{T}_a, t(k) \leq t - p_{ab}\right\},$$

$$F_b^3 \equiv \left\{k \in \mathcal{N}^g \cup \{g\} : t(k) \in \mathcal{T}_b, t(k) \leq t\right\},$$

$$x_a^g \leq \sum_{k \in F_b^4 \cup F_b^5} x_b^k \quad \forall g \in \mathcal{G}_t,\ t \in \mathcal{T}_a : t < |\mathcal{T}| - p_{ab},\ (a, b) \in \mathcal{A}^2, \tag{13.85}$$

$$\text{where } F_b^4 \equiv \{k \in \mathcal{N}^g : t(k) \in \mathcal{T}_b, t(k) < t\},$$

$$F_b^5 \equiv \{k \in \mathcal{N}_g : t(k) \in \mathcal{T}_b, t(k) \leq t + p_{ab}\},$$

$$\sum_{i \in \mathcal{I}} \sum_{k \in F_i^6} o_{rhi}^f x_i^k \leq O_{rt}^g \quad \forall r \in \mathcal{R},\ g \in \mathcal{G}_t,\ t \in \mathcal{T},\ f \in \mathcal{F}, \tag{13.86}$$

$$\text{where } F_i^6 \equiv \left\{k \in \mathcal{N}^g \cup \{g\} : t(k) \in \mathcal{T}_i, t - d_i + 1 \leq t(k) \leq t\right\},$$

$$h \equiv t - t(k) + 1,$$

$$x_i^g \in \{0, 1\} \quad \forall g \in \mathcal{G}_t,\ t \in \mathcal{T}_i,\ i \in \mathcal{I}, \tag{13.87}$$

$$y_j \in \{0, 1\} \quad \forall j \in \mathcal{J}. \tag{13.88}$$

The special constraint sets (13.81) force the assignment (i.e., the execution) of one and only one job for each class. This assignment takes into account all scenarios without being subordinated to any of them.

Constraints (13.82) force the execution of all operations that are required by the selected jobs and prevent the execution of the operations that are required by the jobs that have not been selected in a given class. Additionally, these constraints select the scenario groups and then the time periods for beginning the execution of the operations that have been selected.

Constraints (13.83) prevent the assignment (i.e., the execution) of more than one operation of a given type at the same time period. Note that the scenario group for each time period to begin the execution of the operations belongs to the predecessor path from

the given scenario group, say g down to time period 1. See [66, 73] for a deterministic version.

Constraints (13.84) and (13.85) ensure that the precedence relationships types 1 and 2, respectively, are not violated. Note that the constraints (13.85) force that the scenario group for each time period to begin the execution of operation, say b, belong to either the predecessor path from group g to period 1 or the successor paths from group g to the scenario groups in time period $t(g) + p_{ab}$. Note that t is the time period to begin the execution of operation, say a under scenario group g.

The knapsack constraints (13.86) ensure that the availability of the resources is not violated for each pair resource availability scenario group g and resource consumption scenario f. The predecessor path from group g to period 1 is used. The constraints are based on simple recourse, since the value of variable x_i^k takes into account all the resource consumption scenarios but without being subordinated to any of them, for $k \in \mathcal{N}^g$.

The compact representation (13.79)–(13.88) can also be transformed into a splitting variable representation such that the x-variables and y-variables can be replaced by their respective siblings, where x_i^g is replaced by $x_{it}^\omega \; \forall \omega \in \Omega_g$ for $t = t(g)$ and y_j is replaced by $y_j^\omega \; \forall \omega \in \Omega$, so that there is an independent model for each scenario ω for $\omega \in \Omega$. The nonanticipativity constraints (13.89)–(13.90) are appended to the model.

$$x_{it}^\omega - x_{it}^{\omega'} = 0 \qquad \forall \omega, \omega' \in \Omega_g : \omega \neq \omega', \; g \in \mathcal{G}_t, \; t \in \mathcal{T}_i, \; i \in \mathcal{I}, \tag{13.89}$$

$$y_j^\omega - y_j^{\omega'} = 0 \qquad \forall \omega, \omega' \in \Omega : \omega \neq \omega', \; j \in \mathcal{J}. \tag{13.90}$$

From a practical point of view, and due to its combinatorial nature, even the problem related to a scenario cannot be solved up to optimality in affordable computing time except for moderate-size instances. So, efficient heuristic approaches should be used. In [4], we consider heuristics based on a mixture of a fix-and-relax approach for providing good solutions to the scenario-related sequencing and scheduling problem (see [25, 35]) and a branch-and-fix coordination scheme (see [5]) for coordinating the branching phase in the scenario-related branch-and-bound trees, so that the constraints (13.89)–(13.90) are satisfied.

13.7 Conclusions

Production planning and scheduling (PP&S) is one of the most broadly studied application fields in deterministic optimization. In this paper we have presented a representative set of PP&S application cases, where the treatment of the uncertainty plays a central role in the strategic, tactical, and operational decision making in many corporations. Scenario analysis has proved to be a useful mechanism for representing the uncertainty. Its treatment considers several types of recourse, although full recourse is more frequently used. Interestingly, most of the application cases require 0–1 variables in the modeling schemes. Important PP&S structures such as raw material and product supply time-indexed ordering with delivering time lag, advanced production without explicitly considering stock variables, and multi-period execution operations, among others, have been represented by modeling objects that make use of the scenario tree device. The large-scale instances of the PP&S problems require new algorithmic approaches to benefit from the polyhedral results on the deterministic version of the problems and the stochastic programming results for dealing with the uncertainty. New schemes to increase the performance of the standard Benders and Lagrangian

decomposition approaches are needed. In this sense branch-and-fix coordination schemes for coordinating the execution of the scenario-related branch-and-bound phases for the 0–1 model solving are one of the potential venues for problem solving. This type of scheme aims to accelerate the speed of the nonanticipativity constraint satisfaction in the splitting variable representations of the DEM for stochastic 0–1 problems.

Bibliography

[1] S. AHMED, A. KING, AND G. PARIJA, *A multi-stage stochastic integer programming approach for capacity expansion under uncertainty*, J. Global Optim., 26 (2003), pp. 3–29.

[2] A. ALONSO-AYUSO, L. F. ESCUDERO, A. GARÍN, M. ORTUÑO, AND G. PÉREZ, *An approach for strategic supply chain planning under uncertainty based on stochastic* 0–1 *programming*, J. Global Optim., 26 (2003), pp. 97–124.

[3] A. ALONSO-AYUSO, L. F. ESCUDERO, AND M. ORTUÑO, *A stochastic* 0-1 *program based approach for air traffic management*, Eur. J. Oper. Res., 120 (2000), pp. 47–62.

[4] A. ALONSO-AYUSO, M. F. CLEMENT, L. F. ESCUDERO, M. L. GIL AND M. ORTUÑO, *FRC-S3, A Fix-and-Relax Coordination Scheme for Stochastic Sequencing and Scheduling*, Technical Report I-2003-10, Centro de Investigación-Operativa, Universidad Miguel Hernández, Elche, Spain, 2003.

[5] A. ALONSO-AYUSO, L. F. ESCUDERO, AND M. ORTUÑO, *BFC, a branch-and-fix coordination algorithmic framework for solving some types of stochastic pure and mixed* 0-1 *programs*, Eur. J. Oper. Res., 151 (2003), pp. 503–519.

[6] R. ANTHONY, *Planning and Control Systems: A Framework for Analysis*, Technical Report, Graduate School of Business Administration, Harvard University, Cambridge, MA, 1965.

[7] A. ATAMTÜRK, G. NEMHAUSER, AND M. SAVELSBERGH, *Valid inequalities for problems with additive variable upper bounds*, Math. Program., 91 (2001), pp. 145–162.

[8] P. BAPTISTE, C. LE PAPE, AND W. NUIJTEN, *Constraint-Based Scheduling*, Kluwer Academic Publishers, Norwell, MA, 2001.

[9] P. BARICELLI, C. LUCAS, AND G. MITRA, *A model for strategic planning under uncertainty*, TOP, 4 (1996), pp. 361–384.

[10] E. BEALE, *On minimizing a convex function subject to linear inequalities*, J. Roy. Statist. Soc. Ser. B, 17 (1955), pp. 173–184.

[11] G. BELVAUX AND L. WOLSEY, *Modelling practical lot sizing problems as mixed-integer programs*, Management Sci., 47 (2001), pp. 993–1007.

[12] J. BENDERS, *Partitioning procedures for solving mixed variables programming problems*, Numer. Math., 4 (1962), pp. 238–252.

[13] J. Birge, *Decomposition and partitioning methods for multistage stochastic linear programs*, Oper. Res., 33 (1985), pp. 1089–1107.

[14] J. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer-Verlag, Berlin, 1997.

[15] G. Bitran and D. Tirupati, *Hierarchical production planning*, in Logistics of Production and Inventory, A. R.-K. S. C. Graves and P. Zipkin, eds., North-Holland, Amsterdam, 1993, pp. 523–568.

[16] C. Carøe and R. Schultz, *Dual decomposition in stochastic integer programming*, Oper. Res. Lett., 24 (1999), pp. 37–45.

[17] C. Carøe and J. Tind, *L-shaped decomposition of two-stage stochastic programs with integer recource*, Math. Program., 83 (1998), pp. 451–464.

[18] R. Cheung and W. Powell, *Models and algorithms for distribution problems with uncertain demands*, Transportation Sci., 39 (1996), pp. 43–59.

[19] M. Cohen and H. Lee, *Resource deployment analysis of global manufacturing and distribution networks*, J. Manufacturing Oper. Management, 2 (1989), pp. 81–104.

[20] G. Dantzig, *Linear programming under uncertainty*, Management Sci., 1 (1955), pp. 197–206.

[21] M. Dempster and R. Thompson, *Parallelization and aggregation of nested benders decomposition*, Ann. Oper. Res., 81 (1998), pp. 163–187.

[22] C. Dert, *A dynamic model for asset liability management for defined benefit pension funds*, in Worldwide Asset and Liability Modeling, W. Ziemba and J. Mulvey, eds., Cambridge University Press, Cambridge, UK, 1998, pp. 501–536.

[23] B. Dietrich, *A two phase heuristic for scheduling on parallel unrelated machines with setups*, in Flexible Manufacturing Systems, K. Stecke and R. Suri, eds., Elsevier Science, New York, 1989, pp. 187–192.

[24] B. Dietrich and L. F. Escudero, *On modelling the maximum workload allocation for parallel unrelated machines with setups*, Ann. Oper. Res., 43 (1993), pp. 359–377.

[25] C. Dillenberger, L. F. Escudero, A. Wollensak, and W. Zhang, *On practical resource allocation for production planning and scheduling with period overlapping setups*, Eur. J. Oper. Res., 75 (1994), pp. 275–286.

[26] G. Eppen, R. Martin, and L. Schrage, *A scenario approach to capacity planning*, Oper. Res., 37 (1989), pp. 517–527.

[27] L. F. Escudero, *On maintenance scheduling of production units*, Eur. J. Oper. Res., 9 (1982), pp. 264–274.

[28] L. F. Escudero, *S3 sets. An extension of the Beale-Tomlin special ordered sets*, Math. Program., 42 (1988), pp. 113–123.

[29] L. F. ESCUDERO, *Robust decision making as a decision making aid under uncertainty*, in Decision Theory and Decision Analysis, S. Rios, ed., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994, pp. 127–138.

[30] L. F. ESCUDERO, *CMIT: A capacitated multi-level implosion tool for production planning*, Eur. J. Oper. Res., 76 (1994), pp. 511–528.

[31] L. F. ESCUDERO, J. FUENTE, C. GARCÍA, AND F. PRIETO, *A parallel computation approach for solving multistage stochastic network problems*, Ann. Oper. Res., 90 (1999), pp. 131–160.

[32] L. F. ESCUDERO, E. GALINDO, C. GARCÍA, E. GÓMEZ, AND V. SABAU, *SCHUMANN: A modelling framework for supply chain management under uncertainty*, Eur. J. Oper. Res., 119 (1999), pp. 13–34.

[33] L. F. ESCUDERO, P. KAMESAM, A. KING, AND R. WETS, *Production planning via scenario modelling*, Ann. Oper. Res., 43 (1993), pp. 311–335.

[34] L. F. ESCUDERO, F. QUINTANA, AND J. SALMERÓN, *CORO: A modelling and algorithm framework for oil supply, transformation and distribution optimization under uncertainty*, Eur. J. Oper. Res., 114 (1999), pp. 638–656.

[35] L. F. ESCUDERO AND J. SALMERÓN, *Fix-and-Relax Partitioning. An Algorithmic Framework for Large-Scale Resource Constrained Project Selection and Scheduling*, Technical Report I-2000-02, Centro de Investigación-Operativa, Universidad Miguel Hernández, Elche, Spain, 2002.

[36] M. GASSMANN, *Mslip: A computer code for the multistage linear programming problem*, Math. Program., 47 (1990), pp. 407–423.

[37] C. HAHN, E. DUPLAGA, AND J. HARTLEY, *Supply-chain synchronization: Lessons from Hyundai motor company*, Interfaces, 30 (2000), pp. 32–45.

[38] S. HARTMANN, *Project Scheduling under Limited Resources. Models, Methods and Applications*, Springer-Verlag, Berlin, 1999.

[39] J. HIGLE AND S. SEN, *Stochastic Decomposition. A Statistical Method for Large-Scale Stochastic Linear Programming*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[40] P. KALL AND S. WALLACE, *Stochastic Programming*, John Wiley, New York, 1994.

[41] U. KARMARKAR, *Capacity loading and release planning with work in progress and lead times*, J. Manufacturing Oper. Management, 2 (1989), pp. 105–123.

[42] W. KLEIN HANØVELD AND M. VLERK, *Stochastic integer programming: General models and algorithms*, Ann. Oper. Res., 85 (1999), pp. 39–57.

[43] J. KRARUP AND O. BILDE, *Plant location, set covering and economic lot size: An o(mn) algorithm for structured problems*, in Optimierung bei Graphentheoretischen und gauzzahligen Problemen, Birkhäuser-Verlag, Basel, Switzerland, 1977, pp. 155–180.

[44] G. Laporte and F. Louveaux, *The integer l-shaped method for stochastic integer programs with complete recourse*, Oper. Res. Lett., 13 (1993), pp. 133–142.

[45] C. Lucas, S. Mirhassani, G. Mitra, and C. Poojari, *An application of lagrangian relaxation to a capacity planning problem under uncertainty*, J. Oper. Res. Soc., 52 (2001), pp. 1256–1266.

[46] G. Lulli and S. Sen, *A Branch-and-Price Algorithm for Multi-Stage Stochastic Integer Programming with Application to Stochastic Batch-Sizing Problems*, Technical Report, Stochastic Programming E-Print Series, 2002.

[47] A. Maatan, C. Schweigman, A. Ruijs, and M. van der Vlerk, *Modeling farmers' response to uncertain rainfall in Burkina Faso: A stochastic programming approach*, Oper. Res., 50 (2002), pp. 399–414.

[48] A. Miller, G. Nemhauser, and M. Savelsbergh, *On capacitated lot-sizing and continuous* 0–1 *knapsack polyhedra*, Eur. J. Oper. Res., 125 (2000), pp. 298–315.

[49] S. MirHassani, C. Lucas, G. Mitra, and C. Poojari, *Computational solution of capacity planning model under uncertainty*, Parallel Comput. J., 26 (2000), pp. 511–538.

[50] G. Mitra, *Investigation of some branch and bound strategies for the solution of mixed integer linear programming*, Math. Program., 4 (1973), pp. 155–170.

[51] G. Mitra, M. Hajian, and I. Hai, *A distributed processing algorithm for solving integer programs using a cluster of workstations*, Parallel Comput. J., 23 (1977), pp. 733–753.

[52] R. Möhring, *Stochastic optimization methods in scheduling*, in Proceedings of the Ninth International Conference on Stochastic Programming, Humboldt University, Berlin, 2001.

[53] J. Mulvey and A. Ruszczynski, *A diagonal quadratic approximation method for large-scale linear programs*, Oper. Res. Lett., 12 (1992), pp. 205–221.

[54] M. Novak, R. Schultz, and M. Westphalen, *Optimization of Simultaneous Power Production and Trading by Stochastic Integer Programming*, Technical Report, Stochastic Programming E-Print Series, 2002.

[55] R. Nürnberg and W. Römisch, *A two-stage planning model for power scheduling in a hydro-thermal system under uncertainty*, Optim. Eng., 3 (2002), pp. 355–378.

[56] W. Ogryczak and A. Ruszczynski, *From stochastic dominance to mean-risk models: Semi-deviations as risk measures*, Eur. J. Oper. Res., 116 (1999), pp. 33–50.

[57] M. Pinedo, *Scheduling Theory, Algorithms and Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[58] Y. Pocket and L. Wolsey, *Solving multi-item lot sizing problems using strong cutting planes*, Management Sci., 37 (1991), pp. 53–67.

[59] A. PREKOPA, *Stochastic Programming*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.

[60] R. ROCKAFELLAR AND R. J.-B. WETS, *Scenario and policy aggregation in optimization under uncertainty*, Math. Oper. Res., 16 (1991), pp. 119–147.

[61] W. RÖMISCH AND R. SCHULTZ, *Multi-stage stochastic integer programs: An introduction*, in Online Optimization of Large Scale Systems, M. Grötschel, S. Krumke, and J. Rambau, eds., Springer-Verlag, Berlin, 2001, pp. 581–600.

[62] R. SCHULTZ, *Stochastic programming with integer variables*, Math. Program. Ser. B, 97 (2003), pp. 285–309.

[63] R. SCHULTZ, L. STOUGIE, AND M. VAN DER VLERK, *Solving stochastic programs with integer recourse by enumeration: A framework using Gröbner basis reductions*, Math. Program., 83 (1998), pp. 229–352.

[64] R. SCHULTZ AND S. TIEDEMANN, *Risk aversion via excess probabilities in stochastic programs witn mixed-integer recourse*, SIAM J. Optim., 14 (2003), pp. 115–138.

[65] J. SHAPIRO, *Mathematical programming models and methods for production planning and scheduling*, in Logistics of Production and Inventory, S. Graves, A. R. Kan, and E. Zipkin, eds., North–Holland, Amsterdam, 1993, pp. 371–443.

[66] J. SOUSA AND L. WOLSEY, *A time indexed formulation of non-preemptive single machine scheduling problems*, Eur. J. Oper. Res., 54 (1992), pp. 353–357.

[67] S. TAKRITI AND J. BIRGE, *Lagrangean solution techniques and bounds for loosely coupled mixed-integer stochastic programs*, Oper. Res., 48 (2000), pp. 91–98.

[68] A. TOMASGARD AND E. HØEG, *A supply chain optimization model for the Norwegian Meat Cooperative*, 2005, Chapter 14 in this volume.

[69] P. VALENTE, *Software Tools for the Investigation of Stochastic Programming Problems*, Ph.D. thesis, Department of Mathematics and Computation, Brunel University, London, 2002.

[70] H. WAGNER AND O. BEMAN, *Models for planning capacity expansion of convenience store under uncertain demand and the value of information*, Ann. Oper. Res., 44 (1995), pp. 19–44.

[71] H. WAGNER AND T. WITHIN, *A dynamic version of the economic lot size model*, Management Sci., 5 (1958), pp. 89–96.

[72] L. WOLSEY, *Valid inequalities for mixed integer programs with generalized and variable upper bounds*, Discrete Appl. Math., 25 (1990), pp. 251–261.

[73] L. WOLSEY, *Mip modeling of changeovers in production planning and scheduling problems*, Eur. J. Oper. Res., 99 (1997), pp. 154–165.

[74] W. ZIEMBA AND J. MULVEY, EDS., *Worldwide Asset and Liability Modeling*, Cambridge University Press, Cambridge, UK, 1998.

Chapter 14

A Supply Chain Optimization Model for the Norwegian Meat Cooperative

A. Tomasgard and E. Høeg†*

14.1 Introduction

The food industry is changing rapidly as there is increasing focus on efficiency at all levels in the food production value chain. Producers face requirements to become more responsive to fast changes in the markets, and main customers, like the supermarket chains, require more influence on logistic processes and product development. At the same time there is pressure to reduce both cost and lead times. Typical operations in the supply chain for meat products include receiving raw materials, coordinating production and distribution, deciding on inventory levels for raw materials and finished products, and sales of intermediate and finished products. The main difference from traditional goods production is that the raw material and the products are usually fresh and have limited durability.

One keyword in supply chain optimization is *coordination*. It is a challenge to provide customers simultaneously with higher service level and lower costs. In most industries it is important to coordinate plans between different regions and between the different levels in the supply chain to meet these goals. By enforcing better coordination and thereby capacity utilization, the companies try to avoid suboptimal decisions at every stage in the production and distribution processes.

Coordination is essential because demands in different regions are not perfectly correlated. Different regions can share safety inventories and buffers, and the available resources and raw material can be utilized from a global perspective rather than with focus on local capacities and local demand. Also, coordination makes it possible to specialize production capabilities and to share production capacities for peak production between different regions and different products.

*SINTEF Industrial Management and Norwegian University of Science and Technology, Trondheim, Norway (Asgeir.Tomasgard@sintef.no).

†The Norwegian Meat Cooperative, Trondheim, Norway (erik.hoeg@gilde.no).

Another important aspect is *smoothing*. In a value chain producing food from fresh raw materials, balancing production and inventories is essential to being able to meet variations in demand. This is true both in the short term, capturing normal variations in demand, and in the medium term, capturing seasonal demand variations. Inventories of frozen raw materials can be seen as seasonal smoothing, balancing the supply side of raw material and the demand side for finished products with the seasonal cycles. Safety inventories are an insurance directed toward short-term variations, smoothing out weekly variations. Planning tools that addresses smoothing and stochastic demand can make instant improvements to the coordination of inventories of different regions and the balancing of inventories and production planning.

To achieve synergy effects between regions from uncorrelated demand and smoothing, it is important that the planning in the chain be coordinated. It is also important that the plans take advantage of all production flexibility present in the production system. This brings us to a third aspect of supply chain optimization, namely, how to deal with *uncertainty* through decisions. When planning inventory and production levels, it is important to capture the dynamics of the real-world decision process. We obtain information as uncertainty is gradually resolved. We change our decisions periodically, reacting to the new information. Still, many decision support tools choose a deterministic approach, where this option to change a decision is not captured as plans are made. Using an approach where planning is carried out in a deterministic model and the deterministic model is reoptimized when new information comes along will not repair this planning error. Such a scheme will never choose flexible solutions, unless flexibility is free. This is rarely the case. Making decisions that keep options open will seldom be of value in a model that assumes a deterministic future. There are no incentives to buy insurance if you know what will happen. Only a stochastic model will capture the option value of flexible decisions that can be changed later, and only a stochastic model reflects that the decision maker foresees that the world can change.

Dealing with raw material that is fresh and where the final use of the material is not decided when the animal is slaughtered, it is critical that the decision support tools capture both the uncertainty and the decision flexibility. We model coordination between the different regions and different levels in the chain of production and inventories under uncertain demand. The model is tailored for the Norwegian Meat Cooperative, but it addresses aspects relevant both for other companies in the meat industry and for production companies in other industries. The model presented has an operational time horizon, but several of the characteristics of the model would be similar in the tactical horizon. We also discuss the relation between the two planning horizons.

There is a clear need for research in this direction. As enterprise resource planning (ERP) systems of different kinds become standard in industry, much of the data needed to utilize advanced optimization-based planning tools are already available. These tools have therefore been seen by many companies as possible remedies to help meet the new customer requirements and the need for efficiency and coordination. The problem is that while these tools are good at storing transactions and managing data and thereby represent a foundation for planning, the planning process itself is often neglected. The data needed to do advanced planning are present, but the planning methodology or optimization functionality needed to refine the data into plans is often lacking. Also, in the cases where optimization functionality is included in commercially available ERP systems, they are (to our knowledge

without exception) based on deterministic models.

The research literature in the field of quantitative methods for supply chain management includes contributions from many disciplines like operations research/management science, operations management, strategic analysis, economics, management accounting, and many more. A good review of the literature focusing on both historical development and taxonomy can be found in [10]. In the literature of mathematical programming and operations research we distinguish approaches that have a global (company) view and approaches with an agent view of the supply chain. In the first case the whole supply chain is owned by one company, or one participant in the chain is so dominant it can coordinate all levels of the supply chain. In the second view the supply chain is coordinated through cooperation between agents.

The major stream of operations research literature takes the first view, for example in traditional models on distribution and inventory planning [19], production coordination [12], and location analysis [16]. Examples of the agent view on production coordination and inventory management can be found in [14] and [6]. In this article we focus on operational coordination of a supply chain owned by one company, the Norwegian Meat Cooperative, and therefore choose the global view and describe a decision support tool that optimizes the overall performance of the value chain.

Most of the literature (as most of the literature cited above) deals with deterministic decision support models, but we will give a brief review of some of the directions of research for stochastic supply chain models. In [4] one can find a survey of global supply chain management with focus on strategic models that capture both operational flexibility and financial risk, in particular exchange risk. These models are typically based on option pricing theory, stochastic dynamic programming, stochastic integer programming, or combinations of these. Examples of other stochastic programming models dealing with design of supply chains and capacity planning are [1, 5, 17, 18]. Two-stage stochastic programming models have also been useful within goods distribution [3], transportation planning, and vehicle routing [9, 11]. Examples of operational production planning with stochastic demand can be found in [7, 8]; see also Chapter 13 of this volume.

This article focus on resource management, production, inventories, distribution, and sales with stochastic demand. We present a mathematical model and also show how to find and represent the data needed. In section 14.2 we describe in more detail the supply chain of the Norwegian Meat Cooperative. In section 14.3 we present our modeling assumptions and the mathematical model for operational supply chain optimization. The modeling framework comes from stochastic programming as discussed in [2, 13] and in Part I of this volume, as we need to address both decision flexibility and uncertain demand. In section 14.4 we link forecasting of demand to scenario generation. In section 14.5 we discuss practical experience, and finally we present conclusions and further work.

14.2 Description of the value chain of the Norwegian Meat Cooperative

The Norwegian Meat Cooperative is a cooperative owned by a majority (37,000) of Norwegian farmers. It is the largest meat producing company in Norway. In the Norwegian food industry there has been a concentration of power to a few chains of supermarkets at

the customer end of the supply chain. The Norwegian Meat Cooperative dominates the production side. The company is divided into five regions, North, West, South, East, and Central. The annual turnover is about 1200 million Euro. In 2000, the market shares for the company were 76% for slaughtering, 53% for cutting, 50% for processing, and 50% on sales of meat products. There are also some private enterprises of varying sizes. Supply chain management in its widest consequence has lead to Norwegian supermarket chains entering into meat product processing and to direct contact between the farmers and private processing enterprises.

That the Norwegian Meat Cooperative is owned by its suppliers makes the business special in some respects. First, in many industries companies are free to buy the input they need to maximize profit based on market possibilities and production capabilities. For the Norwegian Meat Cooperative, the situation is the opposite. The input drives the process, as they cannot refuse to receive an animal for slaughtering from one of the member farmers. Then the goal is to transform the input into the products with the highest market value. This clearly is not the same task as maximizing profit by choosing the optimal input and output levels, as most other companies do. Even if all input has to be accepted, the Norwegian Meat Cooperative is free to build up inventories of intermediate products. Throwing away raw material or intermediate products, however, is politically unacceptable (which is reasonable considering the high prices of food in Norway).

Second, the company has to be competitive in several markets—for whole carcasses, cut parts, and finished products. Third, because of the high market shares, the company also is responsible for maintaining the balance in the Norwegian meat market. Together, these three characteristics will make a great challenge for any company, because each decision may affect the goals in opposite directions.

The production process starts with the arrival of live animals at the slaughterhouse—*slaughtering*. The next step in the process is to split the carcasses into smaller standardized pieces, and in some cases further slice the pieces into sirloin, T-bone steaks, and so on—*cutting*. This is done in accordance with some predecided cutting and slicing prescriptions. In some cases the results of this step are ready to market. In other cases they can be used as input in further refined mixed meat products like sausages and meatballs in the *processing* stage of the supply chain. At all stages the capacities for production and inventory capacities are limited. There is also a difference between fresh and frozen raw materials, and there is a raw material market for both. The final step is *distribution and sales*.

The flow of slaughtered animals, cut parts, processed products, and marketable products in the Norwegian Meat Cooperative may be classified in three flows:

- within one region,
- between regions,
- between the Norwegian Meat Cooperative and competitors.

These material flows are shown in Figure 14.1, together with the processes described above. Export and import exist, but those flows are less significant.

We now give short descriptions of the four main processes in the supply chain: slaughtering, cutting, processing, and sales. Then we discuss the link between tactical and operational decisions.

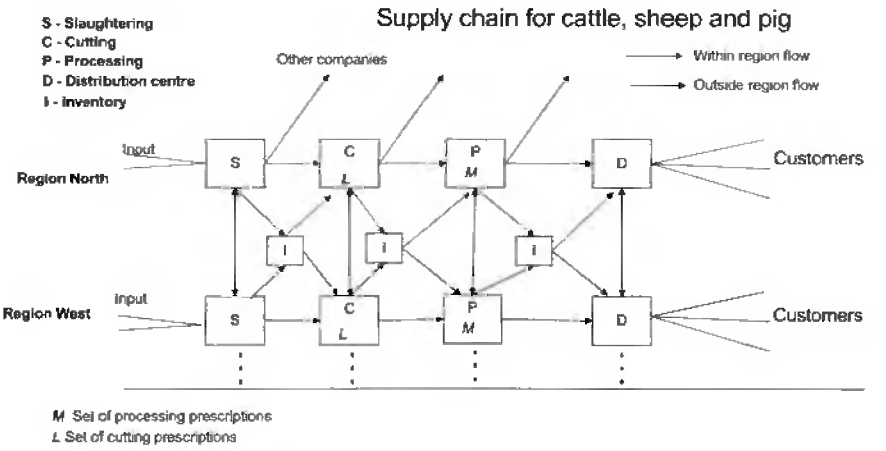


Figure 14.1. *The supply chain of the Norwegian Meat Cooperative.*

14.2.1 Slaughtering

The slaughtering process consists of bringing the animals to the slaughterhouses and thereafter the slaughtering itself, including stalling, killing, bleeding, skinning/dehairing, removal of rumen and intestines, veterinary examination, grading, weighing, and cooling.

In each region, the Norwegian Meat Cooperative has many slaughterhouses. In recent years these have become more specialized, implying that the different locations have comparative advantages for slaughtering particular animal types. Using specialized slaughterhouses will increase the transport costs and make planning and coordination even more important.

Because of its high market share, the Norwegian Meat Cooperative is responsible for maintaining balance in the slaughtering market. The idea is that every competitor shall have supply at the same level as their demand. Some of the slaughtered animals are therefore sold directly, before they reach the cutting process. The flow of products from slaughtering is managed from the headquarters in Oslo. Sales of whole carcasses are mainly made to the competitors of the Norwegian Meat Cooperative as part of regulating the market. There are also some internal sales of slaughtered animals between the different regions within the company, before the animals reach the cutting process. These sales are smaller in volume and less important.

14.2.2 Cutting

The main task in the cutting process is to split the carcass into a set of smaller pieces and to further cut these into smaller pieces suitable for further processing or sales. There exists a large set of cutting patterns, all tailored to serve different situations for further processing. Some patterns will produce parts that are better for producing specific products later. Demands for finished products and production recipes for further processing are

therefore critical issues when one decides which cutting patterns should be chosen.

Each cutting pattern has a calculated yield, giving the cutting plant incentives for choosing among them. Typically the calculated yields from the cutting patterns are not updated often enough to reflect the real market value of the different cutting choices. For example, it is difficult to make the yields reflect market price for the whole carcass, market price for the cut parts, and market price for the different final products that can be made from the cut parts. The yield should also depend on information about regional demands, inventory levels, production capacities in different regions, etc. The possibility to transport cutting parts between regions and to coordinate production between regions also implies that the decisions are not purely local. The real yield connected to a cutting pattern is dynamic and related to the global market situation, the global availability of raw material, and the choice of production recipes in the processing stage.

Decision support models that will help in choosing an optimal set of cutting patterns will be of great value to the company's performance. Then there is a need for changes both in planning methodology and in supporting planning tools. A typical output from such tools will be shadow prices for cut parts giving better indications of more market oriented yields for the cutting patterns. There is a hope that decision support tools like the one described in this article will be helpful in the future, facilitating better comprehension of the value and yields of the different cutting patterns.

At this stage in the process, the Norwegian Meat Cooperative no longer has a responsibility for maintaining market balance. Industrial sales of cut parts are made at market price.

14.2.3 Processing

The processing units produce finished products for sales to the market. The internal market of the processing units are the sales units within the company. In addition the company produces for inventory build-up because of fluctuating prices, short-term demand variations, and seasonal demand variations. In rare cases, industrial sales to competitors occur. In practice this is for companies that have their own brand but outsource production.

To obtain rational production, certain plants specialize in the production of different articles. Production of tinned meat, cured meat, sausages, sandwich meat, and hamburgers occurs at different locations. At this stage it is not only a possibility to exchange goods between regions, it is a necessity. No region produces all products demanded by end users within the region.

The most important decisions are what to produce, where to produce it, and which recipes to use. Much of the production flexibility comes from using different recipes. Every product has at least one recipe, and some maybe as many as 20. The prices of raw material fluctuate during the year. The choice of recipe that carries the minimal cost will therefore change. More important, the possibility of coordinating the use of recipes in different regions makes it possible to utilize the available resources in the best possible way. This may include choosing recipes that are not cost optimal in certain regions. The goal is to choose recipes that maximize overall company profit and to make sure that demand is met. This often means that the cut parts of the cost optimal recipe in one region would have better use in other products either locally or in other regions. Alternative uses exist that will give

higher local costs but also higher global profit outweighing this. To utilize the resources one needs a global view on available cutting parts, the production capacities, and the demand. The performance of the Norwegian Meat Cooperative from a company view includes the idea that unsatisfied demand or surplus of input material also has a cost.

At this point in the process, the decisions made at the cutting departments earlier will now affect which intermediate products are available for processing. The value of the cut parts will rely on which recipes are chosen and which products are demanded. Coordination of planning is the keyword for success.

14.2.4 Sales and distribution

One of the main tasks of the sales and distribution process is to forecast demand. Market signals should then be sent backward in the supply chain to the production plants and cutting plants to make sure the correct decisions concerning recipes and cut parts are made. Another task is to activate mechanisms that lead to higher or lower demand, to better suit the production capabilities within a period and the available raw material. Typically this is achieved by introducing market activities to stimulate demand in case of excess capacity or surplus of raw material and to reduce marketing activities or increase prices when demand is too high or production capacity too low.

The Norwegian Meat Cooperative markets its products under two labels, Gilde and Goman. In many cases the two labels have the same products (made from the same recipes), but both profile and customers differ. In our models we differentiate between these as independent marketable products. Two different products with different prices at the sales level for marketable products may in fact correspond to the same product at the processing level as they are produced using the same raw material and the same recipe.

The Norwegian Meat Cooperative does 90% of its distribution itself, serving shops, supermarket chains, hotels, restaurants, and cafeterias. The company sells thousands of different products. With this complexity and diversity, it is important that the forecasts for demand be precise. In the meat industry both too-low and too-high forecasts will imply inefficient production. Undersupply will lead to unhappy customers, overtime, pressure on machines and workers, cost of offering substitute products at a lower price, etc. Oversupply may lead to wasting products because meat products have limited durability. Some actions to prevent waste will be to offer surplus products at a lower price or to grow inventories when possible (possibly reducing the value of these and substitute products in the future). Also it is likely that when the company produces too much of one product, this steals important material and production resources from other products that are demanded in the market.

Figure 14.2 shows the effect on income due to oversupply or undersupply of a product in the market when demand is d_p^{rt} for product p in period t in region r. The graph shows income as a function of sales, when demand is fixed. In this figure γ is the price that will be paid for all sold units when you sell at least 80% of the demanded volume, but not more than the demand d. Here η is a unit penalty cost for all demand under 80% of d that is not met. For all sales between 100% and 120% of the demand, the price paid is reduced to ζ. For all sales over 120% of d, there is no income. In fact this can be viewed as waste management, and most likely there will be a cost related to this. The numbers used here are of course a rough estimate of the income from different sales volume as a function of demand, but they

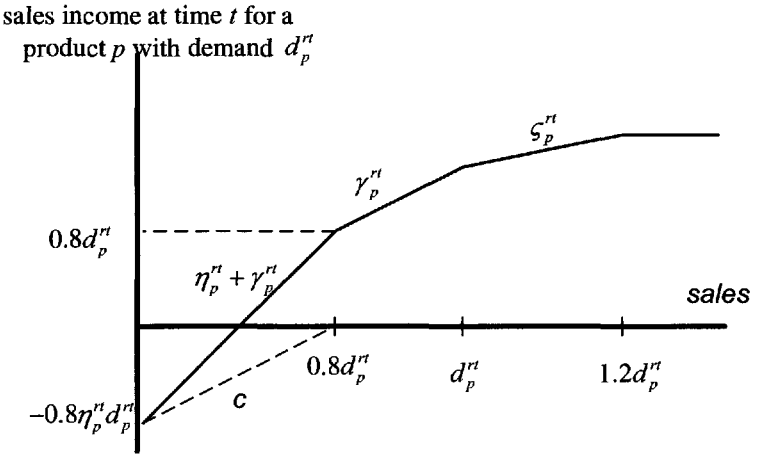


Figure 14.2. *Sales income for a product as a function of sold kilos and market demand.*

reflect quite well the typical situation for the Norwegian Meat Cooperative today.

14.2.5 Time horizons and decisions

The tactical planning horizon for production and distribution in the Norwegian Meat Cooperative is about 12 months. This is the horizon when politically decided price targets for the average price paid to farmers for the animals should be met. Seasonal variations in demand can be handled by tactical inventories for seasonal smoothing. In addition there are restrictions on how long food can be deep frozen before it goes into the market, limiting the length of the tactical planning horizon.

The Norwegian Meat Cooperative is owned by its suppliers and is therefore responsible for selling all the farmers' animals, but still they have some influence. The most effective way to increase or reduce supply is by increasing or lowering prices for animals delivered from the farmers. Another way is to use price to make average weights go up or down. Motivating farmers to sell heavier animals will increase supply. The opposite effect will come from paying a higher price per kilo for light animals. One should have in mind that the cost of slaughtering an animal would be very much the same, regardless of weight. Decisions made to optimize supply in the tactical horizon may therefore increase cost in the slaughtering process at the operational level.

Since most significant changes in the food market can be predicted, it is critical that this knowledge be used to optimize supply. Easter, summer, autumn, and Christmas are all events or seasons that have well-known patterns in the food market. Other seasonal variations also follow well-known patterns. At the same time, supply will consist of some combination of cattle, sheep, and pigs. Failing to have the right kind of availability of animals or cut parts could lead to a situation where demand cannot be satisfied while the company holds goods and raw material which nobody wants.

Decisions that are made in the tactical decision horizon are typically as follows:

- Tactical inventory targets are inventory levels that one should try to achieve for different products at different times of the year to be able to meet demand variations (smoothing) and thereby utilize the available resources to maximize profit over the year.

- Inventory locations for safety inventories or seasonal smoothing can be shared between different regions, as the demands in different regions are not perfectly correlated.

- Target prices influence the price profiles for the different products over the year. Price can be used to stimulate demand. The average price over the year for whole carcasses is politically decided within an upper and a lower bound. Still, prices can be used to give incentives to the farmers to deliver animals at preferable times.

- Marketing and similar activities can be used to stimulate demand.

- Production configuration is important to achieve efficiency. Production is specialized in schemes where certain products are allocated to certain plants for periods of the planning horizon.

In a tactical model, one may wish to focus on medium-term problems, such as inventory build-up, marketing strategy, important production allocations, and material flows. The decisions from the tactical model should place restrictions on operational decisions, where the focus is shorter and may concern decisions for daily or weekly planning.

The model we present in this article is an operational model. At the operational level the goal is to maximize profit within the framework laid out by tactical decisions, responding to short-term variations in demand and supply. This means that inventory targets are set and price profiles are fixed in these models. The model horizon should be 6 to 8 weeks, capturing variations from marketing activities and special holidays. In this horizon also price profiles on animals delivered to the slaughterhouses are fixed. This leads to two important observations. First, the Norwegian Meat Cooperative cannot in general use price paid to the farmers per animal as a mechanism to control inflow in this period. Second, the slaughterhouses have very exact estimates of the inflow of animals to the slaughterhouses within the period. Demand, on the other hand, is highly stochastic, based on natural variation, weather, marketing activities of the supermarket chains, etc.

14.3 The linear stochastic programming model

The model is at the operational level, focusing on production planning and inventory build-up under uncertain demand. We describe coordination between regions and coordination between cutting, processing, and sales. The planning horizon is typically 6 to 8 weeks. We first present the assumptions and simplifications and then provide the notation and the mathematical model. In section 14.4 we present how to find and represent the necessary data for the stochastic programming models in terms of a scenario tree.

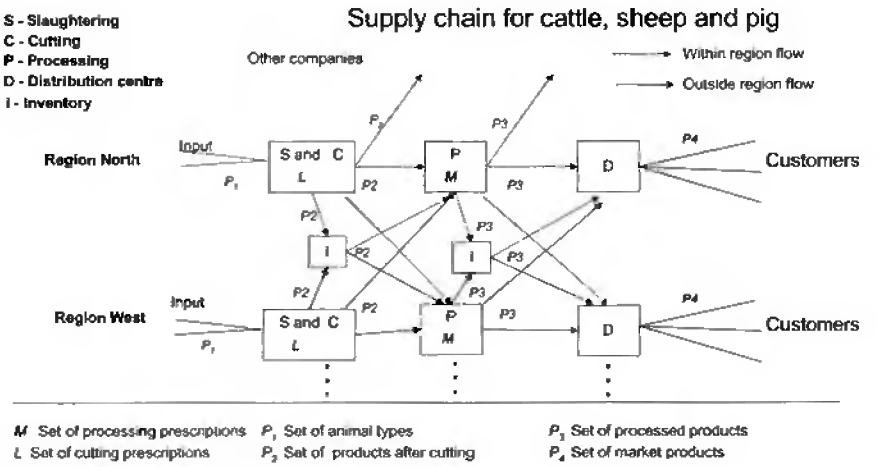


Figure 14.3. *The supply chain.*

14.3.1 Modeling assumptions

We have made some assumptions to be able to model the supply chain of the Norwegian Meat Cooperative as a stochastic linear program. We also made some simplifications where we exclude practical details because we think this will clarify the presentation of the model. We summarize the most important assumptions here. The supply chain we model and the flows of different material types (cut parts, processed products, and marketable products) are shown in Figure 14.3.

First, we make some assumptions on market regulation and material flows of animals before they are cut into pieces. We do not model flow of whole animals to external customers. We omit flow to external customers because we keep market regulation out of the model. It would be politically controversial to include market regulation decisions in a model where the purpose is to maximize the profit of the Norwegian Meat Cooperative.

Also we do not model flows of whole carcasses between different regions internally in the Norwegian Meat Cooperative or inventories of whole carcasses. This makes it possible to collapse the slaughtering and the cutting processes in the model and to eliminate some variables for material flows; see Figure 14.3. Transportation is more efficient for cut parts than for whole animals. The only remaining reason to separate the two processes in the model would then be the possibility of modeling inventories of carcasses. This modeling choice would introduce flexibility, as the choice of cutting patterns could be delayed in time. It would unfortunately also complicate the model dramatically, as the choice of cutting patterns would now be related to an inventory variable and not a constant supply. We would need to introduce binary variables or quadratic terms for modeling this. In practice carcasses are not frozen and stored in inventories from the motivation to maintain this flexibility, so we have chosen to omit it in the current version of the model, as a trade-off between cost and benefit.

In the model we have also eliminated the other material flows between regions when

they are at the same level in the supply chain. Hence, in the model if cut parts are sent to another region, they are sent directly to the processing level. Products that are processed in one region are sent directly to a distribution center if transported to another region. This does not eliminate important parts of the real material flow. It means that sales to the external market happen only within a region. There is no trading between regions for the purpose of external sales at the same level of the supply chain. This would be straightforward to include in the model if necessary.

We aggregate all plants within a region, so that the planning is done on a regional basis, not at plant level. This is done because there normally are only a few major plants in each region. It would be possible to present the model at the plant level instead. We also aggregate products into product groups. This is not critical for the model as it is presented, but this is what is done in practice in running the model. If not, it would become too large.

We include capacity restrictions on the processing plants but not on the cutting. This would be a trivial extension, but adding it here would not contribute further to the clarity of the model. Slaughtering capacities should be resolved by preprocessing. This is not part of the planning today, as one cannot in practice choose not to slaughter an animal when it is delivered. The maximum allowed transport distance for live animals is very limited, so there is little flexibility.

In this article we focus on the material flows that come from the animals. We have assumed unlimited availability of water, milk, spices, wrappers, additives, labels, etc., that are also input to the production. Also we have set no upper limit for inventories. All these extensions are easy to model.

We set no limits on transportation capacities within and between regions. In some respects this is reasonable, because one could rent transportation capacity in the market. One may consider a piecewise linear transportation cost, however, instead of the linear cost used here.

We assume that inflow of animals, prices, and costs are deterministic in a 6- to 8-week time horizon. Demand is considered as the only stochastic variable in the model. The Norwegian Meat Cooperative has much more success in predicting inflows on a 6-week horizon than they have for prediction of demand.

One should notice that processed products are not further refined before they become a marketable product. Still, a single processed product could be marketed under different labels as Gilde, Goman, or others. These products have different demands and different prices. We have allowed this in the model, making a difference between products in the sales link in the chain and in the production. This makes it possible to model price discrimination.

We keep decisons around marketing activities out of the model. Naturally, marketing activities must still be considered in the forecasting methods for estimating demand.

In the objective function we model only cash flows into the Norwegian Meat Cooperative and out from the Norwegian Meat Cooperative. No internal cash flows are modeled, as the model seeks to optimize the overall profit of the Norwegian Meat Cooperative. We maximize expected profit net of shortfall costs from Figure 14.2.

The current version of the application is based on a two-stage model. In section 14.4, where we discuss scenario generation and forecasting, we argue that there is much value in modeling the problem as multistage. The formulation given here can be extended to a multistage model.

14.3.2 Notation

We will first introduce the notation of sets, indices, constants, and variables in the model.

Sets

The most important observation to make here is that we have chosen to model all products and intermediate products as the set $\mathcal{P}$. We have further split this into subsets to reflect the different product categories animal, cutting parts, finished products, and marketable products.

We have modeled the set of time periods as $\mathcal{T}$ and split it into time periods that are in the first stage $\mathcal{T}_1$ of the stochastic program and in the stochastic second stage $\mathcal{T}_2$.

We use the following sets:

$\mathcal{P}_1$ set of animal types (cattle, sheep, and pig);

$\mathcal{P}_2$ set of intermediate products resulting from the cutting process;

$\mathcal{P}_3$ set of finished products resulting from the processing;

$\mathcal{P}_4$ set of marketable products sold from the sales department; we have introduced the possibility of price diversification so that a product from $\mathcal{P}_3$ can be sold as two different products in $\mathcal{P}_4$;

$\mathcal{P}$ set of intermediate and finished products $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2 \cup \mathcal{P}_3 \cup \mathcal{P}_4$;

$\mathcal{M}(p)$ set of recipes for products $p \in \mathcal{P}_3$;

$\mathcal{L}(p)$ set of cutting patterns for products $p \in \mathcal{P}_2$;

$\mathcal{R}$ set of regions;

$\mathcal{T}_1$ time periods with deterministic demands, $t = 1, \ldots, T_1$;

$\mathcal{T}_2$ time periods with stochastic demand, $t = T_{1+1}, \ldots, T_2$.

Indices

We use the following indices to identify elements of the sets defined above:

r region involved in production or in a material flow, normally used for the originating end of the flow;

n region involved in production or in a material flow, normally used for the receiving end of the flow;

l cutting pattern used to cut material into the products $p \in \mathcal{P}_2$;

p finished or intermediate product; also used for animal type when $p \in \mathcal{P}_1$;

o finished or intermediate product, used when two product indices are needed;

m recipe used for products $p \in \mathcal{P}_3$;

t time period.

Constants

α_p^{rt} kilos supplied of animal type $p \in \mathcal{P}_1$ in region r in time period t;

β_p^{rt} cost per kilo (price to farmer) for supply of animal type $p \in \mathcal{P}_1$ in region r in time period t;

γ_p^{rt} price per kilo of product $p \in \mathcal{P}$ in region r in time period t;

ζ_p^{rt} reduced price per kilo of product $p \in \mathcal{P}$ in region r in time period t because of oversupply;

η_p^{rt} penalty for extensive undersupply per kilo of product $p \in \mathcal{P}$ in region r in time period t;

ϕ_{pmo} share of the total weight of product $p \in \mathcal{P}_3$ that comes from product $o \in \mathcal{P}_2$ in recipe m;

ψ_{plo} kilos of cut product $p \in \mathcal{P}_2$ that comes from a kilo of animal $o \in \mathcal{P}_1$ using cutting pattern l;

φ_{po} constant that takes the value 1 if finished product $o \in \mathcal{P}_3$ can be marketed as product $p \in \mathcal{P}_4$ and 0 otherwise;

κ_{pm} extra cost induced by choosing recipe m to produce product $p \in \mathcal{P}_3$ (spices, additives);

λ_{pr}^{nt} transportation cost per kilo product transported from region r to region n of product $p \in \mathcal{P}$ in time period t;

μ_p^{rt} inventory cost per kilo in time period t for product $p \in \mathcal{P}_2 \cup \mathcal{P}_3$ in region r;

ι_p^{rt} target for tactical inventory level for product $p \in \mathcal{P}_2 \cup \mathcal{P}_3$ in region r in time period t;

ς_p^{r0} initial inventory in the beginning of the planning horizon for product $p \in \mathcal{P}_2 \cup \mathcal{P}_3$ in region r;

ν_p^{rt} processing capacity for product $p \in \mathcal{P}_3$ in region r in time period t.

Decision variables

u_{pl}^{rt} share of the animals of type $p \in \mathcal{P}_1$ that are cut according to cutting pattern l in time period t in region r;

v_{pm}^{rt} kilos of product $p \in \mathcal{P}_3$ produced by recipe m in time period t in region r;

w_{po}^{rt} kilos of marketable product $p \in \mathcal{P}_4$ that comes from processed product $o \in \mathcal{P}_3$ in time period t in region r;

x_{pr}^{nt} kilos of product $p \in \mathcal{P}_2 \cup \mathcal{P}_3$ transported to inventory of region n from region r in time period t;

z_{pr}^{nt} kilos of product $p \in \mathcal{P}_2 \cup \mathcal{P}_3$ transported from inventory in region r to region n in time period t;

y_{pr}^{nt} kilos of product $p \in \mathcal{P}$ transported directly from region r to region n in time period t;

i_p^{rt} inventory of product $p \in \mathcal{P}_2 \cup \mathcal{P}_3$ in region r in time period t; initial inventory $(i_p^{r0} = \varsigma_p^{r0})$;

b_p^{rt} total sales of product $p \in \mathcal{P}$ in region r in time period t;

c_p^{rt} sales of product $p \in \mathcal{P}$ in region r in time period t in the interval between 80% of d_p^{rt} and d_p^{rt} (sold at full price);

e_p^{rt} oversupply of product $p \in \mathcal{P}$ that can be sold at reduced price in region r in time period t;

f_p^{rt} oversupply of product $p \in \mathcal{P}$ that cannot be sold in region r in time period t;

g_p^{rt} sales up to 80% of the demanded volume of product $p \in \mathcal{P}$ in region r in time period t. If this variable is not at its upper limit, it will lead to penalties (or loss of good will).

For simplicity we denote the vector of all inventory levels i_p^{rt} as i^t.

Stochastic demand

Demands denoted by $\tilde{d}$ for the different products are stochastic. When uncertainty is resolved and in the deterministic time periods, we denote the deterministic demand by d.

$\tilde{d}_{pr}^{t}$ stochastic demand in region r for product p in time period t seen from a point in time $t' \in \mathcal{T}_1$;

d_{pr}^{t} deterministic demand in region r for product p in time period $t \in \mathcal{T}_1$ or demand in region r for product p in time period $t \in \mathcal{T}_2$ after uncertainty is resolved (seen from a point in time t' where $t \in \mathcal{T}_2$).

For ease of notation the vector of all elements $\tilde{d}_p^{rt}$ of time period t is $\tilde{d}^t$, and similarly the vector of all elements d_p^{rt} of time period t is d^t.

14.3.3 The mathematical formulation

We describe the supply chain optimization model as a two-stage stochastic linear program with relatively complete recourse. The only stochasticity that is present is in the right-hand side. The typical length of a time period is 1 week, and the planning horizon would typically be 8 weeks, where, for example, the first 4 weeks would belong to $\mathcal{T}_1$ and the last 4 weeks to $\mathcal{T}_2$. The objective is to maximize expected profit taking into consideration the shortfall costs. Hence the objective can be described by summarizing the expected cash flow of the

time periods. The cash flow of each time period t can be described as a function $\Pi^t(d^t)$ of demand d^t (or $\tilde{d}^t$ if stochastic):

$$\Pi^t(i^{t-1}; d^t) = \sum_{r\in\mathcal{R}}\sum_{p\in\mathcal{P}} \gamma_p^{rt} c_p^{rt} + \sum_{r\in\mathcal{R}}\sum_{p\in\mathcal{P}} \zeta_p^{rt} e_p^{rt} \tag{14.1}$$

$$+ \sum_{r\in\mathcal{R}}\sum_{p\in\mathcal{P}} (\eta_p^{rt} + \gamma_p^{rt}) g_p^{rt} - \sum_{r\in\mathcal{R}}\sum_{p\in\mathcal{P}} 0.8\, \eta_p^{rt} d_p^{rt} \tag{14.2}$$

$$- \sum_{r\in\mathcal{R}}\sum_{p\in\mathcal{P}_1} \alpha_p^{rt} \beta_p^{rt} - \sum_{r\in\mathcal{R}}\sum_{p\in\mathcal{P}_3}\sum_{m\in\mathcal{M}(p)} \kappa_{pm} v_{pm}^{rt} \tag{14.3}$$

$$- \sum_{r\in\mathcal{R}}\sum_{p\in\mathcal{P}} \mu_p^{rt} i_p^{rt} - \sum_{r\in\mathcal{R}}\sum_{n\in\mathcal{R}}\sum_{p\in\mathcal{P}} \lambda_{pr}^{nt} (y_{pr}^{nt} + x_{pr}^{nt} + z_{pr}^{nt}). \tag{14.4}$$

There are two constant terms, one in the end of the second line and one in the beginning of the third. The first is the penalty paid for the first 80% of d if no demand is met. The second is the price paid to the farmers for receiving the animals. One cannot choose not to receive an animal.

The two-stage stochastic program with fixed relatively complete recourse is

$$\max \sum_{t\in\mathcal{T}_1} \Pi^t(i^{t-1}; d^t) + \mathcal{Q}(i^{T_1}),$$

where

$$\mathcal{Q}(i^{T_1}) = \max E_{\tilde{d}} \left[\sum_{t\in\mathcal{T}_2} \Pi^t(i^{t-1}; \tilde{d}^t) \right],$$

subject to the following constraints which we have split into the categories *slaughtering and cutting, processing, distribution, income, and inventories*. The constraint sets are identical for all time periods $t \in \mathcal{T}_1 \cup \mathcal{T}_2$.

Slaughtering and cutting. This constraint says that the cut parts p that are sold in addition to the ones that are transported to inventories or to the processing plants should equal the amount that is actually cut.

$$b_p^{rt} + \sum_{n\in\mathcal{R}} (x_{pr}^{nt} + y_{pr}^{nt}) - \sum_{o\in\mathcal{P}_1}\sum_{l\in\mathcal{L}(p)} \alpha_o^{rt} \psi_{plo} u_{pl}^{rt} = 0, \quad p \in \mathcal{P}_2, r \in \mathcal{R}. \tag{14.5}$$

Processing. The first constraint is related to a specific cut part p. It says that what comes from the cutting plants and from inventories in regions r into a processing plant in region n should equal what is needed of this cut part in the recipes to produce different finished products o in region n. The second constraint says that what is sold from the processing plant of product p in addition to what is sent to a distribution center or an inventory equals what is produced. The last constraint gives production capacities for

processing plants.

$$\sum_{r\in\mathcal{R}}(y_{pr}^{nt}+z_{pr}^{nt})-\sum_{o\in\mathcal{P}_3}\sum_{m\in\mathcal{M}(o)}\phi_{omp}v_{om}^{nt}=0,\quad p\in\mathcal{P}_2,n\in\mathcal{R},\tag{14.6}$$

$$b_p^{rt}+\sum_{n\in\mathcal{R}}(y_{pr}^{nt}+x_{pr}^{nt})-\sum_{m\in\mathcal{M}(p)}v_{pm}^{rt}=0,\quad p\in\mathcal{P}_3,r\in\mathcal{R},\tag{14.7}$$

$$\sum_{m\in\mathcal{M}(p)}v_{pm}^{rt}\le v_p^{rt},\quad p\in\mathcal{P}_3,r\in\mathcal{R}.\tag{14.8}$$

Distribution. This constraint says that what is received for sales in a region n either from the processing departments or inventories of processed product p is sold through products o. The second equation is included to calculate the sold volume of every marketable product.

$$\sum_{r\in\mathcal{R}}(y_{pr}^{nt}+z_{pr}^{nt})-\sum_{o\in\mathcal{P}_4|\varphi_{op}=1}w_{op}^{nt}=0,\quad p\in\mathcal{P}_3,n\in\mathcal{R},\tag{14.9}$$

$$b_p^{rt}-\sum_{o\in\mathcal{P}_3}w_{po}^{rt}=0,\quad p\in\mathcal{P}_4,r\in\mathcal{R}.\tag{14.10}$$

Income. These constraints model the piecewise linear income as a function of sales.

$$b_p^{rt}-g_p^{rt}-c_p^{rt}-e_p^{rt}-f_p^{rt}=0,\quad p\in\mathcal{P},r\in\mathcal{R},\tag{14.11}$$

$$c_p^{rt}\le 0.2d_p^{rt},\quad p\in\mathcal{P}\backslash\mathcal{P}_1,r\in\mathcal{R},\tag{14.12}$$

$$g_p^{rt}\le 0.8d_p^{rt},\quad p\in\mathcal{P}\backslash\mathcal{P}_1,r\in\mathcal{R},\tag{14.13}$$

$$e_p^{rt}\le 0.2d_p^{rt},\quad p\in\mathcal{P}\backslash\mathcal{P}_1,r\in\mathcal{R}.\tag{14.14}$$

Inventories. These constraints model the inventory balance and the tactical inventory targets.

$$i_p^{rt-1}+\sum_{n\in\mathcal{R}}x_{pn}^{rt}-\sum_{n\in\mathcal{R}}z_{pr}^{nt}-i_p^{rt}=0,\quad p\in\mathcal{P}_2\cup\mathcal{P}_3,r\in\mathcal{R},\tag{14.15}$$

$$i_p^{rt}\ge\iota_p^{rt},\quad p\in\mathcal{P}_2\cup\mathcal{P}_3,r\in\mathcal{R}.\tag{14.16}$$

14.4 Scenario generation and forecasting

In the previous sections we discussed the supply chain optimization model and its mathematical formulation. We have argued that it is important to include stochasticity in operational planning, and we have shown how to do it in a stochastic programming model. Now we discuss how to find and represent the data for stochastic demand. As most other production companies have done, the Norwegian Meat Cooperative has based its production on common forecasting methods that predict static estimates for future sales based on expectations, medians, or other one-dimensional measures. A stochastic programming method requires the dynamic description of a scenario tree. In our scenario trees, we would like to approximate the distribution of demand for a given product, based on historical data. This

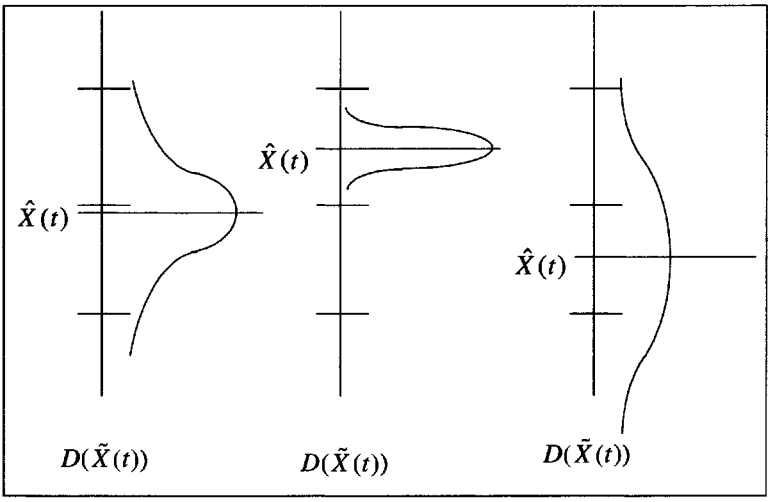


Figure 14.4. *Symmetric distributions from normally distributed error terms.*

is of course closely related to forecasting. For a further description of forecasting methods developed for the Norwegian Meat Cooperative, see [21]. Here we will not discuss the forecasting methods, but rather we discuss the links between forecasting and scenario generation.

We use the forecasting model

$$\hat{X}(t) = \alpha + \beta_1 X(t-1) + \beta_2 X(t-2).$$

The notation $\tilde{X}(t)$ denotes the stochastic variable that we will forecast. Furthermore we let $X(t)$ be an observation and $\hat{X}(t)$ the forecast. In light of the forecasting model we have chosen as an example, the assumed data generating process is

$$\tilde{X}(t) = \alpha + \beta_1 X(t-1) + \beta_2 X(t-2) + \tilde{\epsilon}.$$

Here $\tilde{\epsilon}$ is the stochastic error term related to the forecasting model, and we assume it is identically and independently distributed for all time periods.

A common method to produce scenario trees from forecasts is to utilize the empirical distribution of the error term to describe the uncertainty around the forecasted value. One possible method is to assume that the error term is normally distributed with distribution $\tilde{N}(0, \sigma^2)$, where σ is the observed empirical standard deviation of the error term. This gives a symmetrical distribution for $\tilde{X}(t)$ based on the forecasted value $\hat{X}(t)$, as shown in Figure 14.4. When using these forecasts to build scenario trees, we discretize the distribution represented by branches in the trees. Naturally, the forecasted value itself will vary in different parts of the scenario tree and in different time periods. The forecasting method takes care of this. But the spread around the forecasted value will be symmetrical and equal in every time period and in every node in the scenario tree. This is a consequence of assuming an error term that is normally distributed. So whatever level we observe of $X(t-1)$ and $X(t-2)$, the shape of the distribution around the forecast $\hat{X}(t)$ will be equal.

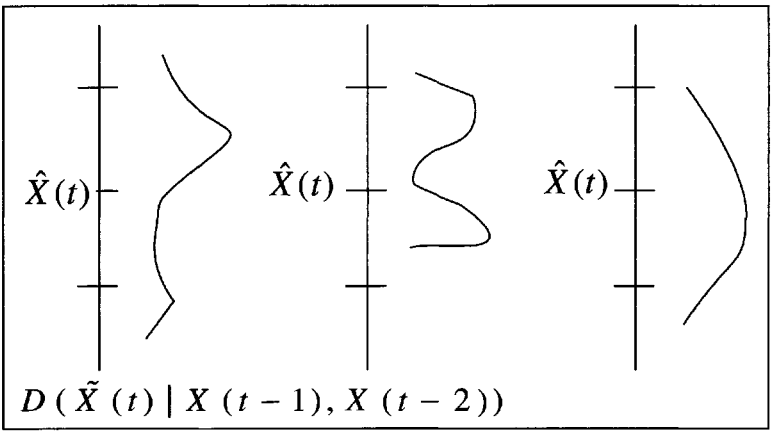


Figure 14.5. *Different shapes of probability distributions.*

This last point is a weakness of the method for practical use in the meat industry. It is not natural that demand distributions are symmetrical around a forecasted value independent of the forecasted value. This is particularly clear when the forecasted value is close to an upper or lower limit for credible demands. Likewise it is unnatural that the spread of the distribution is independent of the forecasted value. Generating scenario trees based on this method will thus lead to an erroneous representation of uncertainty.

Another way to create a scenario tree from a forecasting method assuming equally and independently distributed error terms would be a simple heuristic: use the observations of the error term as a discrete empirical distribution and create scenarios directly from it (possibly by collapsing some of the observations). In contrast to assuming a normal distribution of the error term, this method will manage to handle asymmetric distributions shown in Figure 14.5. Still the spread around the forecasted value will be equal in every time period and independent of the forecasted value. As before, this is a weakness. It is not credible that a forecast close to the lower limit of sales and a forecast close to the upper limit of sales should generate distributions in a scenario tree with the same spread and skewness around the forecasted value. Again this is a result of representing uncertainty through the error term.

Both methods mentioned above suffer from the fact that they are not suitable for modeling situations where the shape of the distribution of demand depends on the forecasted level. The normal distribution approach in addition suffers from the assumption of symmetry.

It is easy to imagine numerous situations where the shape of a distribution depends on the level of demand. We believe that this is the case in the market for meat products, and therefore we choose an approach that allows for such differences, namely, quantile regression as in [15]. This method does not use the error term to represent the uncertainty, but instead uses regression analysis to forecast quantiles of the distribution. The model

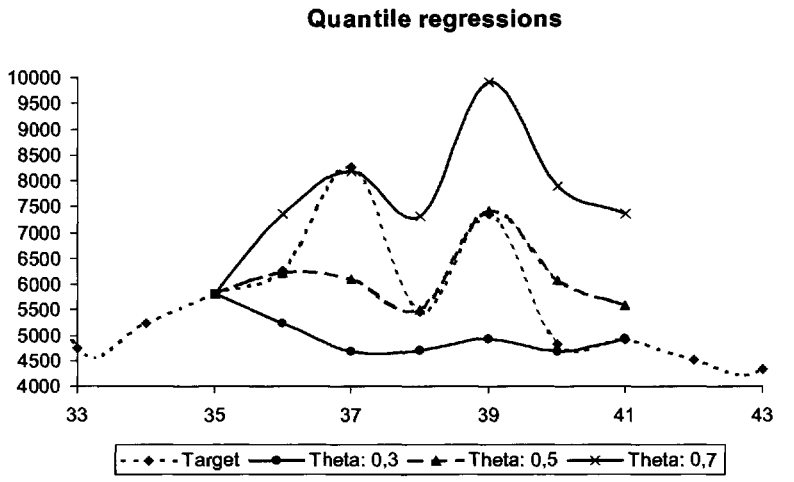


Figure 14.6. *Representation of quantile forecasts.*

suggested by Koenker and Basset in [15] would for our choice of forecasting model

$$\hat{X}(t) = \alpha + \beta_1 X(t-1) + \beta_2 X(t-2)$$

be represented as the problem of finding an $\hat{X}(t)$ such that the probability $P(\tilde{X}(t) < \hat{X}(t)) = \theta$ when forecasting the θ quantile. This can be done (at least asymptotically) by solving the optimization problem

$$\min_{\alpha,\beta_1,\beta_2,b} \sum_{t|X(t)\geq b} \theta|X(t) - \alpha - \beta_1 X(t-1) - \beta_2 X(t-2)| + \sum_{t|X(t)<b} (1-\theta)|X(t) - \alpha - \beta_1 X(t-1) - \beta_2 X(t-2)|.$$

This problem can be formulated as a linear program and is easily solved. We will now give a short description of how to use this to generate scenarios. First let us look at how we could use this method to give a better view of the uncertainty. Figure 14.6 presents the forecasts for θ equal to 30%, 50%, and 70% for barbecue sausage for the next 4 weeks seen from the end of week 35 in year 2000. The parameters α, β_1, and β_2 are reestimated for every week so the forecasting model used is

$$\hat{X}(t+\tau) = \alpha(\tau) + \beta_1(\tau)X(t-1) + \beta_2(\tau)X(t-2), \quad \tau = 1, \ldots, T.$$

This is typically how the method would be used to create input for a two-stage stochastic model. The presentation in Figure 14.6 cannot yet be viewed as scenarios, as the figure presents the quantiles. Hence the 70% quantile should naturally be understood as the forecast of which historically 30% of the observations have been above the forecasted value.

Assume that you would like to have a tree with three branches with equal probability weight to estimate demand of barbecue sausages. Then it would be natural to divide the

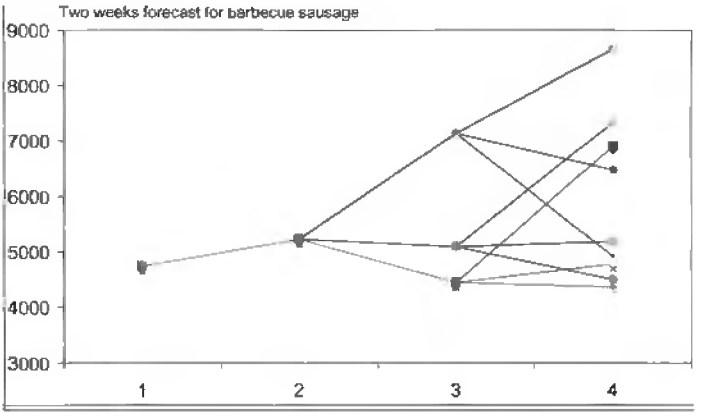
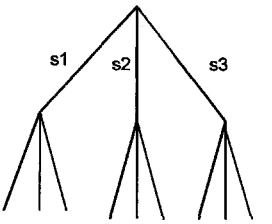


Figure 14.7. *Scenario tree for sausages based on quantile regression.*

	α	β_1	β_2
s1	2527,776	0,207691	0,17398
s2	53,82816	0,629098	0,369824
s3	894,9632	0,650654	0,597599



$$(0, \theta_{0.33}) \quad (\theta_{0.33}, \theta_{0.67}) \quad (\theta_{0.67}, \infty)$$

$$\theta_{0.16} \qquad \theta_{0.5} \qquad \theta_{0.84}$$

Figure 14.8. *Scenario tree and parameter estimation based on quantile regression.*

support of the demand into three intervals with the same probability weight: $(0, \theta_{1/3})$, $(\theta_{1/3}, \theta_{2/3})$, and $(\theta_{2/3}, \infty)$. One would then forecast the quantiles representing the medians of each interval, namely, $\theta_{1/6}, \theta_{1/2}$, and $\theta_{5/6}$ to represent the three demands in the scenario tree (all with equal weight 1/3). This is illustrated in a three-stage scenario tree in Figure 14.7. Here we have used the same parameter sets for both forecasting periods; see Figure 14.8. One could instead have chosen an approach similar to the one presented first in Figure 14.6 and then updated the parameters in each time period.

The example in Figure 14.7 shows that after an increase in sales, it is likely that demand will continue to rise, while after a reduction in sales, it is not likely that the reduction will continue. The first observation to make is that this would have been hard to realize using the

error term to estimate the uncertainty around a forecasted value. The fundamental difference of the quantile regression from the other methods mentioned is that it does not use the error term to represent uncertainty around the forecasted value; it forecasts the quantiles directly. This makes it possible to represent the shape of the distribution depending on the demand level and depending on the time period. As in Figure 14.8, there is a parameter set for each quantile, allowing for different representations of the forecasting method for different levels of sales.

The other observation is the effect that the previous demand observations have on the shape of the demand distribution. This effect and in particular the dynamics of the development would have been hard to see in a two-stage tree. This indicates that using three-stage or multistage models would be advantageous for planning by the Norwegian Meat Cooperative. This kind of information about the dynamic development of demand is critical, both for production planning and for the people in the marketing departments.

There are not many references in the literature to the use of quantile regression for forecasting. There is still theoretical exploration to do, but the method seems to work well in practice. For a further discussion on quantile regression in forecasting and a more detailed description of this work, see [20].

Correlations between products are not handled in the current approach. One has to assume independently distributed demand for the different products and make combined scenario trees for all the products one would like to represent with uncertain demand. Clearly, if there are many stochastic demands the tree size will explode.

14.5 Experience from practical use

The current version of the model has been tested in a pilot test. The data were taken from the production data in the ERP system of the Norwegian Meat Cooperative. The pilot test gave three important results. First, it verified that the Norwegian Meat Cooperative already stores the necessary data to use a model like this. Second, it verified that the problem sizes we need to solve are within the limits of state-of-the-art solution technologies. Third, it indicated an economical potential for using such a tool for coordinating the supply chain.

So far we have not worked with solution methodology to solve the problem efficiently. We have used Xpress-MP to solve the stochastic linear programs deterministic equivalent, without tuning parameters or utilizing structure. On a Pentium III 733 Mhz with 756 MB memory it is then possible to solve problems with around 2 million variables and 1 million constraints within 12 hours. Table 14.1 illustrates some typical problem sizes for relevant problem instances. The problems we need to solve typically have around 100 end products, 40 to 75 intermediate products (cutting parts), 3 animal types, and 5 regions. This means that we are able to solve problems limited to between 10 and 25 scenarios using off-the-shelf software. This limits the description of uncertainty to focus on the demand of a few important products.

Clearly, it is interesting to give a richer description of uncertainty, in terms of including demand for additional products and extending the model to multistage. To be able to do this we will in future work concentrate in two directions. First, we will exploit how large-scale optimization methods for stochastic programming apply to this problem. Second, we will investigate adequate ways to aggregate products to reduce problem dimension.

Table 14.1. *Examples of problem sizes for typical planning situations.*

Scenarios	$\lvert\mathcal{P}_1\rvert$	$\lvert\mathcal{P}_2\rvert$	$\lvert\mathcal{P}_3\rvert$	$\lvert\mathcal{P}_4\rvert$	$\lvert\mathcal{R}\rvert$	$\lvert\mathcal{T}_1\rvert$	$\lvert\mathcal{T}_2\rvert$	Variables	Constraints
10	3	40	100	100	5	4	4	961664	379060
10	3	75	100	100	5	4	4	1123364	440660
25	3	40	100	100	5	4	4	2273024	895960
25	3	75	100	100	5	4	4	2655224	1041560
50	3	40	100	100	5	4	4	4458624	1757460
50	3	75	100	100	5	4	4	5208324	2043060
100	3	40	100	100	5	4	4	8829824	3480460
100	3	75	100	100	5	4	4	10314524	4046060

Because of confidentiality, economical results we can report from our tests are limited. Still we can report on the progress of the project. The forecasting methods developed, including the scenario generation, have been implemented commercially in the organization. New business processes for forecasting and coordination of the production departments and sales departments are developing in parallel. The first results from the first year of testing show dramatically improved quality of forecasts, but it is always difficult to judge whether this comes from the new focus or from the methods. Most likely it comes from both.

The supply chain optimization model itself is in testing and further development in the company during the winter and spring of 2003. The purpose of this testing is to develop more efficient solution routines and to find the appropriate level of aggregation. A commercial implementation of the stochastic programming model in the organization will of course imply a reorganization of some business processes. In parallel we therefore start the work of developing business processes for production, inventory, and sales coordination together with the company.

14.6 Conclusions

We have discussed the importance of modeling uncertainty and dynamics in supply chain coordination models. We have presented a two-stage stochastic linear program with recourse, designed for operational coordination and planning in the supply chain of a meat producer. The ability to model stochastic demand opens up the possibility of modeling shared inventories and better utilizing raw material and production capacities. One of the main reasons for this is that demand is not correlated for different products and in different regions of the market.

We argue that results from traditional forecasting methods representing uncertainty by a single forecasted value for demand are not adequate as the basis for planning production or building inventories. Deterministic models do not allow for the proper coordination and promote neither sharing of inventories for smoothing purposes nor flexible decisions in cutting plants and processing plants.

One usual way to create scenarios from existing forecasting methods is to use the empirical distribution of the error term of the forecasting model as a basis. We have argued

that this is often not adequate, as the shape of the error term is then normally assumed to be independent of the demand level and the time period, leading to symmetric trees or trees where the demand distributions have the same shapes in every node. There are many cases where this does not resemble the reality we try to model.

We suggest instead the use of quantile regression as a basis for generating scenario trees. Then we forecast the quantiles directly and do not have to go through the error terms. Examples show that this gives important information about the demand distributions and their shape at different points in time and in different parts in the tree.

This work also shows that it would be advantageous to model the problem as a multistage stochastic program. This would lead to larger scenario trees, but as Part I of this volume shows, problems with millions of variables can be solved in acceptable times.

Another problem with scenario tree generation is how to model the correlations between different products and different regions. In the current approach we assume independence of demand distributions and combine the outcomes of all stochastic variables. This dramatically reduces the number of stochastic variables we are able to work with when solving the model, as the tree sizes become too large. Future work includes methods for reducing the number of uncertain variables through factor analysis and principal component analysis, so correlations can be found for a smaller set of factors than today's product groups.

Bibliography

[1] A. Alonso-Ayuso, L. F. Escudero, A. Garín, M. T. Ortuño, and G. Pérez, *An approach for strategic supply chain planning under uncertainty based on stochastic 0-1 programming*, J. Global Optim., 26 (2003), pp. 97–124.

[2] J. R. Birge and F. V. Loveaux, *Introduction to Stochastic Programming*, Springer-Verlag, Berlin, 1997.

[3] R. K. M. Cheung and W. B. Powell, *Models and algorithms for distribution planning with uncertain demands*, Transport. Sci., 30 (1996), pp. 43–59.

[4] M. A. Cohen and A. Huchzermeyer, *Global supply chain management: A survey of research and applications*, in Quantitiative Models for Supply Chain Management, S. Tayur, R. Ganeshan, and M. J. Magazine, eds., Kluwer Academic Publishers, Norwell, MA, 1999, pp. 840–879.

[5] B. Dominguez-Ballesteros, C. Lucas, and G. Mitra, *SCHUMANN, Strategic Planning under Uncertainty in the Manufacturing and Processing Industry*, Technical Report, Department of Mathematical Sciences, Uxbridge, Middlesex, UK, 2001.

[6] R. Ernst and S. G. Powell, *Manufacturer incentives to improve retail service levels*, Eur. J. Oper. Res., 104 (1998), pp. 437–450.

[7] L. F. Escudero, E. Galindo, G. Garcia, E. Gomez, and V. Sabau, *Schumann, A modelling framework for supply chain management under uncertainty*, Eur. J. Oper. Res., 119 (1999), pp. 14–34.

[8] L. F. ESCUDERO, P. K. KAMESAM, A. J. KING, AND R. J. B. WETS, *Production planning via scenario modelling*, Ann. Oper. Res., 43 (1993), pp. 311–335.

[9] A. FEDERGRUEN AND P. ZIPKIN, *A combined vehicle routing and inventory allocation problem*, Oper. Res., 32 (1984), pp. 1019–1037.

[10] R. GANESHAN, E. JACK, M. J. MAGAZINE, AND P. STEPHENS, *A taxonomic review of supply chain management research*, in Quantitiative Models for Supply Chain Management, S. Tayur, R. Ganeshan, and M. J. Magazine, eds., Kluwer Academic Publishers, Norwell, MA, 1999, pp. 840–879.

[11] M. GENDREAU, G. LAPORTE, AND R. SÉGUIN, *Stochastic vehicle routing*, Eur. J. Oper. Res., 88 (1996), pp. 3–12.

[12] A. C. HAX AND D. CANDEA, *Production and Inventory Management*, Prentice–Hall, Englewood Cliffs, NJ, 1984.

[13] P. KALL AND S. W. WALLACE, *Stochastic Programming*, John Wiley, Chichester, UK, 1994.

[14] D. KJENSTAD, *Coordinated Supply Chain Scheduling*, Ph.D. thesis, Norwegian University of Science and Technology, Trondheim, Norway, 1998.

[15] R. KOENKER AND G. BASSET, Regression quantiles, *Econometrica*, 46 (1978), pp. 33–50.

[16] M. LABBÉ AND F. V. LOVEAUX, *Location problems*, in Annotated Bibliographies in Combinatorial Optimization, M. Dell'Amico, F. Maffioli, and S. Martello, eds., John Wiley, New York, 1997, pp. 261–281.

[17] S. A. MIRHASSANI, C. A. LUCAS, AND G. MITRA, *An Application of Lagrangean Relaxation to a Capacity Planning Problem under Uncertainty*, Technical Report, Brunel University, Uxbridge, UK, 1999.

[18] S. A. MIRHASSANI, C. A. LUCAS, G. MITRA, E. MESSINA, AND C. POOJARI, *Computational solution of capacity planning models under uncertainty*, Parallel Comput., 26 (2000), pp. 511–538.

[19] D. SIMCHI-LEVI, P. KAMINSKY, AND E. SIMCHI-LEVI, *Designing and Managing the Supply Chain: Concepts, Strategies and Case Studies*, McGraw–Hill, New York, 1999.

[20] L. H. VIK, A. TOMASGARD, AND M. P. NOWAK, *Quantile Regression in Forecasting*, Technical Report, SINTEF Industrial Management, Trondheim, Norway.

[21] L. H. VIK, A. TOMASGARD, AND M. P. NOWAK, *Sales Forecasts for Meat Products*, Technical Report, SINTEF Industrial Management, Trondheim, Norway.

Chapter 15

Melt Control: Charge Optimization via Stochastic Programming

Jitka Dupačová and Pavel Popela†*

15.1 Motivation

Melt control problems belong to the broad field of production control applications. They are studied as one of the production steps in iron and steel works. Melt control problems may be fully separated from other foundry optimization problems, which simplifies the model building and its solution. Their importance arises because foundries usually have high overheads, and hence even small percentage savings may recover a significant amount of money. In addition, material inputs represent the biggest part of the total melting costs.

The produced alloys and input materials are composed of basic elements (iron, carbon, etc.). The production process consists of several steps (e.g., charge, alloying). During the alloying process, the hot melt in the furnace is enriched with input materials (return materials, scrap, ferroalloys, etc.), and the new mixture is melted again. Hence, the problem has a natural multistage decision structure. Whereas the unit costs of the inputs are known at the time of decision making, the composition of input materials is not known precisely, and it was modeled as random in [6]. In each step of the process, the melt composition changes and, particularly, random losses of elements in the melt must be considered. During heating of the melt the amounts of elements change randomly, e.g., due to a rise of slag and oxidation. These losses depend on the composition of the melted materials. In some cases, they may be influenced also by the amounts of these materials used, which will not be considered here. The remaining amount of an element is expressed as a linear function in the input quantities of all considered elements, and the coefficients are called *utilizations* of the considered element related to the amount of other elements in the melt.

*Department of Probability and Mathematical Statistics, Charles University Prague, Sokolovská 83, CZ-186 75 Prague, Czech Republic (dupacova@karlin.mff.cuni.cz).

†Department of Mathematics, Brno University of Technology, Technická 2, CZ-616 69 Brno, Czech Republic (popela@um.fme.vutbr.cz).

The goal is to find amounts of the input materials in the lowest cost so that the prescribed output alloy composition is achieved. We use scenario-based two- and three-stage stochastic linear programs to illustrate basic modeling ideas for charge optimization of induced and electric-arc furnaces. For a general approach to melt control, developed for any alloy, furnace, and technology, see [9, 10].

15.2 Examples

In the following two simplified examples, the random losses and hence the related utilizations of elements are taken as the only random variables. This assumption can be accepted, for example, for melting ferroalloys of a guaranteed composition, and it will be relaxed in the last example. Historical melt reports are available and may be used to construct scenarios or scenario trees of utilizations for the melt control problems; see section 15.3.

15.2.1 Two-stage induced furnace charge optimization

We begin with a simple model for charge optimization of iron production in an induced furnace—a model with a common two-stage structure. Through the initial charge decision the final cost of the melt is minimized, taking into account also the consequences of possible random losses and the requirements on the final composition of the melt. The problem in extensive form (see [1]) is

$$\text{minimize} \sum_{j\in J_1} c_j x_j^1 + \sum_{k_2\in K_2} p_{k_2} \sum_{j\in J_2} c_j x_j^{k_2} \tag{15.1}$$

subject to

$$l_{i1} \le \sum_{l=1}^{m_1} \tau_{il}^E \sum_{j\in J_1} a_{lj} x_j^1 \le u_{i1}, \quad i = 1,\dots,m_1, \tag{15.2}$$

$$l_{i2} \le \sum_{l=1}^{m_1} \tau_{il}^{k_2} \sum_{j\in J_1} a_{lj} x_j^1 + \sum_{j\in J_2} a_{ij} x_j^{k_2} \le u_{i2}, \quad i = 1,\dots,m_2, \quad k_2 \in K_2, \tag{15.3}$$

$$x_j^1 \ge 0, \quad j \in J_1, \qquad x_j^{k_2} \ge 0, \quad j \in J_2, \quad k_2 \in K_2, \tag{15.4}$$

where

- $t = 1, 2$ are stages;
- J_t is the set of indices of input materials available at stage t, m_t is the number of elements at stage t, and indices i and l specify them;
- $c_j \ge 0$, $j \in J_t$ are known unit costs of the jth input material;
- $l_{it}, u_{it} \ge 0$ are prescribed lower and upper goal bounds for the amount of the ith element in the melt composition at stage t;
- $a_{ij} \ge 0, \sum_i a_{ij} \le 1 \,\forall j$ denote the amount of the ith element in the unit amount of the jth input material;

- $x_j^1 \geq 0$ denote the first-stage decision variables, the amount of the jth input material at the beginning of the melt process (charge);
- $x_j^{k_2}$, $k_2 \in \mathcal{K}_2$ denote the second-stage decision variables, which stay for the additional amount of the jth input material assigned under scenario k_2 (alloying).

The only random elements are utilizations. Let

- $\tau_{il}^{k_2}$ be the utilization of the ith element related to the amount of the lth element in the melt when scenario k_2 occurs, $0 \leq \tau_{il}^{k_2} \leq 1$, and
- k_2 be indices of scenarios with probabilities $p_{k_2} \geq 0$.

The frequently considered case $\tau_{il}^{k_2} = 0$ when $i \neq l$ means that interactions of random losses are ignored.

In the first-stage constraints (15.2), τ_{il}^E stands for an experience-based "standard" utilization which applies to the first stage. Usually, $\tau_{il}^E = \sum_{k_2 \in \mathcal{K}_2} p_{k_2} \tau_{il}^{k_2}$ $\forall i, l$, i.e., the average utilizations. These constraints reflect the metallurgical rules which aim at the process control stability and, in general, they cannot be neglected. On the other hand, possible losses of the materials added in the second stage of the melting process are negligible and are not considered.

The model has a common two-stage structure with fixed recourse. The charge decisions x_j^1 take into account consequences of the random utilizations at the second stage, making the final alloy composition reachable.

15.2.2 Three-stage electric-arc furnace charge optimization

The situation is more complicated with a steel production in an electric-arc furnace. Because of two alloying phases, the whole process must be modeled as a three-stage problem. To simplify the model description, we mostly utilize the notation of the example in section 15.2.1 and the arborescent form introduced, e.g., in [1] or in [5, Part II], with indices k_t corresponding to nodes of the tth stage (i.e., to stage t scenarios), $t = 2, 3$, and $a(k_t)$ denoting the index of the ancestor node to the node k_t.

To obtain the Markovian structure of the model constraints, the melt composition is described explicitly by additional auxiliary variables $h_i^{k_t}$ describing the state of the decision process—the amount of ith melt element at node k_t of stage t before a subsequent decision was taken. We assume an empty furnace at the beginning of the process, hence, $h_i^1 = 0 \, \forall i$ at stage 1, then $h_i^{k_2} = \sum_{l=1}^{m_1} \tau_{il}^{k_2} \sum_{j \in J_1} a_{lj} x_j^1$, etc. Hence, for $t > 1$ and all elements considered at stage t,

$$h_i^{k_t} = \sum_{l=1}^{m_{t-1}} \tau_{il}^{k_t} \left(h_l^{a(k_t)} + \sum_{j \in J_{t-1}} a_{lj} x_j^{a(k_t)} \right) \tag{15.5}$$

The model is

$$\text{minimize} \sum_{j \in J_1} c_j x_j^1 + \sum_{t=2}^{3} \sum_{k_t \in \mathcal{K}_t} p_{k_t} \sum_{j \in J_t} c_j x_j^{k_t} \tag{15.6}$$

subject to

$$\sum_{l=1}^{m_{t-1}} \tau_{il}^{k_t} \left(h_l^{a(k_t)} + \sum_{j \in J_{t-1}} a_{lj} x_j^{a(k_t)} \right) - h_i^{k_t} = 0, \quad i = 1, \ldots, m_{t-1}, \quad k_t \in \mathcal{K}_t, \quad t = 2, 3, \tag{15.7}$$

$$l_{it-1} \leq \sum_{k_t \in \mathcal{K}_t} p_{k_t} h_i^{k_t} \leq u_{it-1}, \quad i = 1, \ldots, m_{t-1}, \quad t = 2, 3, \tag{15.8}$$

$$l_{i3} \leq h_i^{k_3} + \sum_{j \in J_3} a_{ij} x_j^{k_3} \leq u_{i3}, \quad i = 1, \ldots, m_3, \tag{15.9}$$

$$x_j^1 \geq 0, \quad j \in J_1, \qquad x_j^{k_t} \geq 0, \quad k_t \in \mathcal{K}_t, \quad j \in J_t, \quad t = 2, 3. \tag{15.10}$$

The expected cost of melt is minimized subject to constraints requiring that during the whole melt process the average melt composition satisfies the given bounds and that the final product satisfies these bounds for all scenarios. In the last stage, full utilization of added materials is assumed. For $t = 2$, the constraints (15.8) are in agreement with (15.2) based on average utilizations $\tau_{il}^E = \sum_{k_2 \in \mathcal{K}_2} p_{k_2} \tau_{il}^{k_2}$.

An extension to more that three stages is evident. However, it is clear that the two introduced models must be further generalized and significantly extended (e.g., involving additional linear technological and inventory constraints, uncertain scrap composition, or for more than three stages) before they become applicable in real-world foundries; see [10] for a discussion and suggestions.

15.2.3 Random input composition

Theoretically, the probability distribution of the jth input composition $a_{ij}\ \forall j$ may be estimated from data obtained by a repeated chemical analysis of the input. In practice, normative values (based on such measurements) are used and/or intervals $\underline{a}_{ij} \leq a_{ij} \leq \overline{a}_{ij}$ are built separately for each of the input materials. This information is exploited in the scenario generation procedure which will be explained in section 15.3 and which provides scenarios $s \in \mathcal{S}$ of input matrices $A = (a_{ij}, i = 1, \ldots, m_1, j \in J_1)$ and their probabilities π_s; we index by superscripts s the corresponding elements of matrices A and the second-stage variables. These scenarios and their probabilities are supposed to be known before the melting process starts, and the first-stage decisions depend on this probabilistic specification. In addition, suppose that the composition of the (high-quality) input added in the second stage (alloying) is known and that the initial charge is based on expert estimates $a_{ij}^E, i = 1, \ldots, m_1$, of the composition of input materials $j \in J_1$. Assuming independence of the random input composition and utilizations, we rewrite the example in section 15.2.1 as follows:

$$\text{minimize} \sum_{j \in J_1} c_j x_j^1 + \sum_{s \in \mathcal{S}} \sum_{k_2 \in \mathcal{K}_2} p_{k_2} \pi_s \sum_{j_2 \in J_2} c_{j_2} x_{j_2}^{k_2 s} \tag{15.11}$$

subject to

$$l_{i1} \leq \sum_{l=1}^{m_1} \tau_{il}^E \sum_{j \in J_1} a_{lj}^E x_j^1 \leq u_{i1}, \quad i = 1, \ldots, m_1, \tag{15.12}$$

$$l_{i2} \le \sum_{l=1}^{m_1} \tau_{il}^{k_2} \sum_{j \in J_1} a_{lj}^{s} x_j^1 + \sum_{j \in J_2} a_{ij} x_j^{k_2 s} \le u_{i2}, \quad i = 1, \ldots, m_2, \quad k_2 \in \mathcal{K}_2, \quad s \in \mathcal{S}, \tag{15.13}$$

$$x_j^1 \ge 0, \quad j \in J_1, \qquad x_{j_2}^{k_2 s} \ge 0, \quad j_2 \in J_2, \quad k_2 \in \mathcal{K}_2, \quad s \in \mathcal{S}. \tag{15.14}$$

15.3 Scenario generation

When building a melt control program, scenario generation is one of the most important tasks. There are a large number of melt control reports; however, these reports contain only indirect information about scenarios. Usually only measurements $h_i^{k_t}\ \forall i, k_t$ and the inputs $x_j^{k_t}\ \forall j, k_t$ are specified together with full or partial information about the composition $\boldsymbol{A} := (a_{ij}, \forall i, j)$ of the input materials. Although the ideas are also valid for multistage problems, we will explain them for the simple two-stage melt control program formulated in the example in section 15.2.1. In this case the melt control reports list most of all the inputs amounts x_j^1, $j \in J_1$, and the resulting melt composition

$$h_i^{k_2} = \sum_{l=1}^{m_1} \tau_{il}^{k_2} \sum_{j \in J_1} a_{lj} x_j^1, \quad i = 1, \ldots, m_1, \ k_2 \in \mathcal{K}_2. \tag{15.15}$$

15.3.1 Scenarios of diagonal utilization matrices

In the simplest case, with diagonal utilization matrices $\boldsymbol{T}$ and a known composition $\boldsymbol{A}$ of input materials, the nonzero utilizations $\tau_{ii}^{k_2}$ are obtained as the solution of the trivial system of equations

$$h_i^{k_2} = \tau_{ii}^{k_2} \sum_{j \in J_1} a_{ij} x_j^1, \quad i = 1, \ldots, m_1, \ k_2 \in \mathcal{K}_2. \tag{15.16}$$

Utilizations obtained in this way may be used directly as *measurement-based* scenarios in the second-stage constraints

$$l_{i2} \ \le \ \tau_{ii}^{k_2} \sum_{j \in J_1} a_{lj} x_j^1 + \sum_{j \in J_2} a_{ij} x_j^{k_2} \ \le \ u_{i2}, \quad i = 1, \ldots, m_2, \ k_2 \in \mathcal{K}_2;$$

compare with (15.3).

The melt control reports contain measurements on the auxiliary state variables and report the applied decisions for all stages of the production process. The information related to the tth stage is composed from the initial measurement $h_i^{k_{t-1}}$, stage-related inputs $x_j^{k_t}$, and the final measurement $h_i^{k_t}$. Assuming an empty furnace at the beginning, the result is a "fan" of measurement-based scenarios of utilizations $(\tau_{ii}^{k_1}, \ldots, \tau_{ii}^{k_T})\ \forall i$ which branch only at the root and have equal probabilities.

Another possibility is to generate a limited number, say K, of diagonal utilization matrices, taking into account specific statistical properties of marginal probability distributions of their diagonal elements $\tau_{ii}\ \forall i$, e.g., their expectations μ_i and variances σ_i^2, and also covariances ρ_{il} of couples τ_{ii}, τ_{ll}. This means to estimate these moment values, e.g., using

the past melt reports and to find a feasible solution T^k, p_k, $k = 1, \ldots, K$, of the *moment fitting* problem

$$\sum_{k=1}^{K} p_k \tau_{ii}^k = \mu_i, \qquad i = 1, \ldots, m_1,$$

$$\sum_{k=1}^{K} p_k (\tau_{ii}^k - \mu_i)^2 = \sigma_i^2, \qquad i = 1, \ldots, m_1,$$

$$\sum_{k=1}^{K} p_k (\tau_{ii}^k - \mu_i)(\tau_{ll}^k - \mu_l) = \rho_{il}, \quad i, l = 1, \ldots, m_1,$$

$$\sum_{k=1}^{K} p_k = 1, \qquad p_k \geq 0, \quad k = 1, \ldots, K.$$

To get the *fitted scenarios* and their probabilities means to solve this nonlinear system with respect to p_k and $\tau_{ii}^k \; \forall i, k$. The system can be further extended for constraints on the ranges of the utilization values, on higher-order moments, etc. For a consistent statistical specification, the general results of the moment problem imply that a solution of such a system exists for a modest number of scenarios. Otherwise one may try to get an approximate solution, e.g., by minimizing the weighted sum of squares of differences

$$\sum_{i=1}^{m_1} \left[\alpha_i \left(\sum_{k=1}^{K} p_k \tau_{ii}^k - \mu_i \right)^2 + \beta_i \left(\sum_{k=1}^{K} p_k (\tau_{ii}^k - \mu_i)^2 - \sigma_i^2 \right)^2 \right] + \sum_{i,l=1}^{m_1} \gamma_{il} \left(\sum_{k=1}^{K} p_k (\tau_{ii}^k - \mu_i)(\tau_{ll}^k - \mu_l) - \rho_{il} \right)^2 \tag{15.17}$$

subject to $\sum_{k=1}^{K} p_k = 1$, $p_k \geq 0$, $k = 1, \ldots, K$. Parameters α_i, β_i, γ_{il} can be used to reflect importance and quality of data; see [4, 7] for a detailed discussion.

The assumption of diagonal utilization matrices facilitates scenario generation for the three- and multistage models; it is frequently used in practice and will be accepted also in the numerical illustration in section 15.4. Still, an extension to general utilization matrices and to random composition of input material is important. System (15.15) cannot identify the utilizations in a unique way even if there is an experience-based benchmark for their values, expressed, for instance, in the form of simple box constraints, lower and upper bounds on $\tau_{il}^{k_2}$ valid for all k_2. Allowing for random coefficients a_{lj} independent of utilizations may help; see section 15.3.4.

15.3.2 Scenario tree generation

The pathwise input by scenarios as described above does not display the information structure given by the technological process, and a scenario tree should be built. The number of branching points is linked with the stages of the corresponding production process, as in the example in section 15.2.2.

A simple case of a scenario tree refers to the *interstage independence* with tth stage utilizations $\tau_{ii}^{k_t}$ independent of utilizations in the preceding stages. This means that all melt reports concerning stage t may be used to get utilizations $\tau_{ii}^{k_t}$ for all nodes identified by $\tau_{ii}^{k_{t-1}}$. This approach carries over the equal scenario probabilities at all stages so that the probabilities of all paths from the root to leaves of the scenario tree are equal, too.

Accepting interstage independence means a simplification whose disadvantage is that the number of nodes of such a tree grows rapidly. There are other ways to construct the scenario tree; see, e.g., [4, 5]. In section 15.4, we apply the moment fitting approach of [7] explained briefly in subsection 15.3.1 to generate a tree which mimics the statistical properties of the joint probability distribution, including the interstage dependence.

15.3.3 Scenarios of input composition

It is natural to build independent scenarios of composition for individual input materials. On the other hand, statistical dependence of the contents of the considered elements in the given input material should not be, in general, disregarded.

As the first possibility assume that the composition of the jth input material, $\boldsymbol{a}^j := (a_{ij}\ \forall i)$, is a multinormal vector, $\mathcal{N}(\boldsymbol{a}^{jE}, \boldsymbol{V}^j)$, with expectations $a_{ij}^E\ \forall i$ and the variance matrix estimated from experimental data obtained by chemical analysis. Scenarios $\boldsymbol{a}^{js}$ may then be sampled from this distribution. Another possibility is to sample scenarios from one-dimensional marginal distributions $\mathcal{N}(a_{ij}^E, \sigma_{ij}^2)$ or to discretize these marginal distributions independently for all i and to accept all m_1-tuples of these independent marginal scenarios as scenarios of $\boldsymbol{a}^j$. The next step is to select representative scenarios and their probabilities so that the moment values $\boldsymbol{a}^{jE}$, $\boldsymbol{V}^j$ are retained; see, e.g., [2] or [7].

If the intervals $\underline{a}_{ij} \le a_{ij} \le \overline{a}_{ij}$ listed separately for each element i provide the only available information, we may accept the corresponding uniform distributions as the model of the random composition and to approximate these distributions by the nearest (in the sense of a selected probability metric) discrete uniform one-dimensional distributions. Such approximation depends on the chosen probability metric.

Let $F(t)$ denote the distribution function of a random variable and $\hat{F}(t)$ the distribution function obtained by an approximation. For instance, think of distributions carried by three scenarios $a_{ij}^E - \delta_{ij}$, a_{ij}^E, $a_{ij}^E + \delta_{ij}$, with equal probabilities 1/3. The optimal values of δ_{ij} are equal to $1/3(\overline{a}_{ij} - a_{ij}^E)$ for the Kolmogorov metric

$$\varrho_K(F, \hat{F}) = \sup_t |F(t) - \hat{F}(t)|$$

and to $2/3(\overline{a}_{ij} - a_{ij}^E)$ for the Wasserstein metric

$$\varrho_W(F, \hat{F}) = \int |F(t) - \hat{F}(t)|dt.$$

Another recommendation is to put expectations $a_{ij}^E = 1/2(\underline{a}_{ij} + \overline{a}_{ij})$ and $\sigma_{ij}^2 = \frac{1}{16}(\underline{a}_{ij} - \overline{a}_{ij})^2$, to accept normal marginal distributions with these parameters, and to approximate them by symmetric discrete probability distributions concentrated, say, again in three atoms, $a_{ij}^E - \delta_{ij}$, a_{ij}^E, $a_{ij}^E + \delta_{ij}$, with equal probabilities 1/3. This time the optimal

values of δ_{ij} obtained by minimization of the Wasserstein metric are equal to $1.225\sigma_{ij}$, i.e., to $0.612(\overline{a}_{ij} - a_{ij}^E)$. See [8] for further examples of approximations by discrete distributions.

Concerning covariances, there are typically no records, and only experts' knowledge may be used. Hence, once more, one may accept all m_1-tuples of the independent marginal scenarios obtained by discretization and select representative scenarios and their probabilities to fit the moment values $\boldsymbol{a}^{jE}$, V^j.

Finally, observe that the random coefficients appear only on the right-hand sides of constraints (15.13) that may be rewritten as

$$l_{i2} - \sum_{l=1}^{m_1} \tau_{il} \sum_{j \in J_1} a_{lj} x_j^1 \le \sum_{j \in J_2} a_{ij} x_j^2 \le u_{i2} - \sum_{l=1}^{m_1} \tau_{il} \sum_{j \in J_1} a_{lj} x_j^1, \quad i = 1, \ldots, m_1. \quad (15.18)$$

Hence for a given charge x_j^1, $j \in J_1$, observed utilizations $\tau_{il}\, \forall i$, l, and compositions $\boldsymbol{a}^j$, $j \in J_1$, the minimum cost additional input x_j^2, $j \in J_2$, solves the second-stage linear program

$$\min_{x_j^2 \ge 0 \forall j} \sum_{j \in J_2} c_j x_j^2 \quad \text{subject to (15.18)}$$

whose optimal value is a convex function of $h_i := \sum_{l=1}^{m_1} \tau_{il} \sum_{j \in J_1} a_{lj} x_j^1\, \forall i$. For $a_{ij} \in [\underline{a}_{ij}, \overline{a}_{ij}]$ with expectation $a_{ij}^E = 1/2(\underline{a}_{ij} + \overline{a}_{ij})$ we have

$$h_i \in I_i := [\underline{h}_i, \overline{h}_i] \quad \text{and} \quad h_i^E = 1/2(\underline{h}_i + \overline{h}_i), \quad (15.19)$$

where $\underline{h}_i = \sum_{l=1}^{m_1} \tau_{il} \sum_{j \in J_1} \underline{a}_{lj} x_j^1\ \forall i$ and $\overline{h}_i = \sum_{l=1}^{m_1} \tau_{il} \sum_{j \in J_1} \overline{a}_{lj} x_j^1\ \forall i$. Hence, the *minimum expected cost of the additional input is attained for the average values* h_i^E by Jensen's inequality, i.e., by solving program (15.1)–(15.4) with $a_{ij} = a_{ij}^E \forall i$, $j \in J_1$ in (15.3). In this optimistic case, one uses the most favorable, degenerated distribution of state variables implied by the assumed intervals and expectations of the random composition $a_{ij} \forall i$, $j \in J_1$ of the input materials. The pessimistic, worst-case distribution corresponding to (15.19) is discrete, concentrated on vertices of the Cartesian product of intervals $I_i\, \forall i$; cf. the Edmundson–Madansky bound. In the multidimensional case, an explicit formula may be given only under special assumptions such as the stochastic independence or separability of the second-stage optimal value with respect to h_i, $\forall i$ (see [1]), which is not realistic in our context. One obtains then the worst-case probability distribution for which the marginal distributions of $h_i\, \forall i$, are concentrated on $\underline{h}_i$, $\overline{h}_i$ with equal probabilities $1/2$. As a consequence (compare with (15.13)), for each $j \in J_2$ there are nonnegative second-stage variables $\underline{x}_j^{k_2}$ and $\overline{x}_j^{k_2}\, \forall k_2$, and inequalities (15.3) split into

$$l_{i2} \le \sum_{l=1}^{m_1} \tau_{il}^{k_2} \sum_{j \in J_1} \underline{a}_{lj} x_j^1 + \sum_{j \in J_2} a_{ij} \underline{x}_j^{k_2} \le u_{i2}, \quad i = 1, \ldots, m_2, \quad k_2 \in \mathcal{K}_2,$$

and

$$l_{i2} \le \sum_{l=1}^{m_1} \tau_{il}^{k_2} \sum_{j \in J_1} \overline{a}_{lj} x_j^1 + \sum_{j \in J_2} a_{ij} \overline{x}_j^{k_2} \le u_{i2}, \quad i = 1, \ldots, m_2, \quad k_2 \in \mathcal{K}_2.$$

In the objective function (15.1), $x_j^{k_2}$ is replaced by the average $1/2(\underline{x}_j^{k_2} + \overline{x}_j^{k_2})\ \forall k_2 \in \mathcal{K}_2,\ j \in J_2$. This is in agreement with (15.11).

An additional assumption of *unimodal* marginal distributions of h_i separately for each i results in one-dimensional worst-case marginal distributions *uniform* over the intervals I_i. This conclusion and other extensions may be found among results of the minimax approach and moment problems; see, e.g., [3].

Scenarios of matrices of the input composition may then be created by combining all possibilities taken into account for individual input materials.

Let $s_j,\ s_j \in \mathcal{S}_j$ be scenarios representing the random composition of the jth input and π_{s_j} their probabilities. Combining all possible outcomes for each of the input materials leads to $S = \prod_{j \in J_1} (\#\mathcal{S}_j)$ scenarios of the technological matrices $\boldsymbol{A} = (a_{ij})$. Their probabilities π_s are equal to the product of the corresponding probabilities $\pi_{s_j}\ \forall j$.

15.3.4 Scenarios of nondiagonal utilization matrices

Assume that *independently* of utilizations the composition of individual inputs is indicated in the melt reports. To simplify the notation we index scenarios of the matrices $\boldsymbol{A} = (a_{ij})$ by a superscript s as in the example in section 15.2.3. Assume that an expert is able to select groups of melting reports, say $\mathcal{S}(k_2)$, for which the same utilizations $\tau_{il}^{k_2}$ are likely. It means that for each k_2 *separately* utilizations $\tau_{il}^{k_2}$ satisfy the system

$$h_i^{k_2 s} = \sum_{l=1}^{m_1} \tau_{il}^{k_2} \sum_{j \in J_1} a_{ij}^s x_j^{1s},\ i = 1, \ldots, m_1,\ s \in \mathcal{S}(k_2). \tag{15.20}$$

The $a_{ij}^s\ \forall i, j$ and $s \in \mathcal{S}(k_2)$ in (15.20) are known coefficients. The chance to identify the utilizations $\tau_{il}^{k_2}\ \forall i, l$ by solving system (15.20) depends on scenarios $\boldsymbol{A}^s,\ s \in \mathcal{S}(k_2)$; the system may have more than one solution as well as be inconsistent, and the linear model (15.20) may be used to estimate utilizations $\tau_{il}^{k_2}$. The probability of the resulting scenario—the (nondiagonal) matrix $\boldsymbol{T}^{k_2}$—is proportional to the cardinality $\#S(k_2)$ of $S(k_2)$. If $\#S(k_2)$ is large in comparison with m_1, the set $S(k_2)$ may be replaced by a union of disjoint sets $S'(k_2)$ each consisting of m_1 elements so that the matrix of the corresponding subsystem of (15.20) is nonsingular. By solving these subsystems of equations separately, one gets several equiprobable scenarios of matrices of utilizations for each k_2.

Theoretically, the moment fitting procedure (see subsection 15.3.1) may be used again. However, its numerical tractability—numerical solution of a large nonconvex weighted least squares problem—is an open question.

The *scenario tuning procedure* by [9] assumes that there are at disposal experts' scenarios $\hat{\boldsymbol{T}}^{k_2},\ k_2 \in \mathcal{K}_2$. Hence, the corresponding number of second-stage decision vectors $\boldsymbol{x}_2^{k_2}$ that satisfy (15.3) for $k_2 \in \mathcal{K}_2$ is introduced. One expects also that the already recorded decisions, say $(\hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2)$, should be feasible, or nearly feasible, for each of the experts' scenarios. If it is not the case, using the past recorded experience the input experts' scenarios are tuned for the sake of feasibility of the production process. One allows small perturbations $\boldsymbol{\Delta}^{k_2},\ k_2 \in \mathcal{K}_2$, of the matrices of utilizations $\hat{\boldsymbol{T}}^{k_2}$ and applies the perturbations which provide the best fit. This can be done by solving *separately* for each $k_2 \in \mathcal{K}_2$ the quadratic program

$$\text{minimize } \|\Delta^{k_2}\|^2$$

with respect to $\Delta_{il}^{k_2}\ \forall i, l, k_2$, subject to

$$l_{i2} \le \sum_{l=1}^{m_1}(\hat{\tau}_{il}^{k_2} + \Delta_{il}^{k_2}) \sum_{j\in J_1} a_{lj}\hat{x}_j^1 + \sum_{j\in J_2} a_{ij}\hat{x}_j^{k_2} \le u_{i2}, \quad i = 1, \ldots, m_2,$$

for all pairs $(\hat{x}^1, \hat{x}^2)$.

The matrices Δ^{k_2*} of minimal perturbances are used to "tune" or update for each k_2 the initial experts' scenarios $\hat{T}^{k_2}$ to $\hat{T}^{k_2} + \Delta^{k_2*}$, which then provide the sought input for the stochastic program in question.

15.4 Model implementation

In this section, we illustrate discussions about melt control modeling principles introduced in previous sections by examples based on real-life data. We consider steel production in one Czech foundry with a furnace capacity of 20 tons. We focus on a specific technology combining production steps realized in an electric-arc furnace (EAF) with a ladle furnace (LF) finishing. We restrict ourselves to the steel denoted 42CrMo. See Table 15.1 for the required composition of the produced alloy. The goal intervals $[l_{iT}, u_{iT}]\ \forall i$ are specified by decimal numbers between 0 and 1 defining how many kilograms of the ith element should be contained in 1 kilogram of the produced liquid metal. The factory-defined goal intervals are tighter than those given by Czech standards to achieve the higher product quality. The required amount of the produced steel is 14.9 tons. The input materials, prices, their compositions, and elements are given in Table 15.1.

Table 15.1. *Produced alloy and input materials.*

	c_j	t		l_{iT} and u_{iT}									
Material	Price	Stage	L/U	Fe	C	Mn	Si	P	S	Cr	Mo	Al	Cu
42CrMo		final	L	0.9500	0.0038	0.0060	0.0010			0.0100	0.0020	0.0001	
			U	0.9800	0.0050	0.0085	0.0035	0.0003	0.0003	0.0150	0.0028	0.0004	0.0001
Scrap	3.0	1	L	0.9600	0.0010	0.0040	0.0020	0.0001	0.0001	0.0020			0.0001
			U	0.9900	0.0040	0.0080	0.0040	0.0003	0.0004	0.0040			0.0003
RM955	3.9	1		0.9804	0.0025	0.0035		0.0003	0.0003	0.0110	0.0020		
T951	3.7	1		0.9829	0.0025	0.0035		0.0003	0.0003	0.0070	0.0035		
Coke	1.5	1 2			1.0000								
FeMo	525.0	2		0.3490	0.0010						0.6500		
FeSiMn	18.9	2 3		0.0850	0.0100	0.7000	0.2000			0.0050			
AlInput	50.0	2 3					0.0500					0.9500	
FeCr6	21.2	3		0.2350	0.0650					0.7000			
FeSi45	10.1	3		0.5497			0.4500	0.0002	0.0001				

The whole production process is a multistage one. A two-stage model is applied first in subsection 15.4.1. In this case, we may obtain it either by aggregation of all stages following the first stage or by modeling only the end of the production process. We begin with the second possibility. It may be interpreted as the model for the initial and final alloying phases.

Frequently, the results have to be obtained under real-time control restrictions, so a sensible scenario generation becomes very important.

Then, in subsection 15.4.2, a three-stage model adds the first stage called a charge to the previously discussed two-stage model. In addition, with the three-stage model, the interstage dependence and the random composition of the scrap must be considered.

Both subsections have a similar structure and use the same set of data. First, available data are presented and discussed. Then, models already defined in section 15.2 are slightly extended to allow realistic computations. Special attention is devoted to the collection of the model input data and hence to the generation of scenarios in different ways. Therefore, computations are realized for various deterministic reformulations starting from simple ones. The models are implemented in GAMS and solved using solvers implementing either standard (CPLEX and OSL) or decomposition-based solution algorithms (OSLSE). The most interesting results are further analyzed.

15.4.1 Two-stage melt control

The EAF with ladle teeming is chosen. The important general information can be found in Table 15.1, i.e., for $t = 1, 2, 3$ we find there sets of input materials and the number of considered elements $m = m_t = 10 \forall t$, coefficients c_j, $j \in J_t$, a_{ij}, $j \in J_t, i = 1, \ldots, m$, and l_{it}, u_{it} $\forall i = 1, \ldots, m$. Specifically, we are interested now only in rows 2 and 3 in the Stage column of Table 15.1. To continue, we need more information about the process flow. It is important that all melts are documented in melt reports. Hence, together with general information contained in Table 15.1 that is valid for all technologies developed for 42CrMo steel production in combination of EAF and LF, we have several tables, such as Table 15.2, storing information just about one melt.

Table 15.2. (a) *Amounts of input materials in kg.* (b) *Measurements in* 100%.

	x_j^t		
Material	Stage 1	Stage 2	Stage 3
Scrap	5400		
RM955	6000		
T951	5000		
Coke	100	20	
FeMo		11	
FeSiMn		100	50
AlInput		8	2
FeCr6			130
FeSi45			45
Sum [kg]	16500	139	227

(a)

m	h_i^{T+1}/w^{T+1}		
Elements	Stage 1	Stage 2	Stage 3
Fe	0.9722	0.9634	0.9697
C	0.0029	0.0043	0.0049
Mn	0.0011	0.0056	0.0079
Si	0.0001	0.0001	0.0020
P	0.0002	0.0002	0.0002
S	0.0002	0.0002	0.0002
Cr	0.0066	0.0064	0.0124
Mo	0.0019	0.0024	0.0024
Al		0.0002	0.0003
Cu	0.0000	0.0000	0.0000
Melt weight w^{T+1} in kg	15299	14960	14928

(b)

Tables 15.1 and 15.2 fully identify one run of the steel production process for 42CrMo. At first, the furnace is empty ($h_i^1 = 0 \forall i$). During the charge stage the amounts of input materials (see the second column of Table 15.2(a) for x_j^1) must be taken into account. Then,

the mixture is melted and the composition is measured (multiply proportions and melt weight from the first column of Table 15.2(b) to obtain h_i^2). The alloying cost for the melt related with Table 15.2 is $z_{HB}^2 = 12{,}351$ and the total melt cost equals $z_{HB}^1 = 70{,}601$.

The idea may be applied repeatedly and at the end all x_j^t and h_i^{t+1} are specified. In addition, $h_i^{T+1} \in [l_{iT}, u_{iT}]$ so the required alloy is made. As in (15.5), we introduce $h_i^{k_{T+1}} = \sum_{l=1}^{m_T} (h_l^{a(k_{T+1})} + \sum_{j \in J_T} a_{lj} x_j^{a(k_{T+1})})$. For simplicity we replace $h_i^{k_t}$ by h_i^t when only one fixed scenario, completely specified by stage t and index i, is taken into account.

With the two-stage model, we are asked to continue the melting process, knowing the weight and chemical composition of the melt before alloying begins (see the first column of Table 15.2(b)). Although Tables 15.1 and 15.2 contain all necessary information for initial deterministic optimization, we still need to modify the model (15.1)–(15.4). We begin our computations in stage 2, i.e., after the charge and before alloying, having the first measurement results at our disposal. Thus, we must recognize that the furnace already contains the melt. We also see that the composition of input materials for stages 2 and 3 is known. The goal intervals are specified by relative proportions; hence, l_{it} and u_{it} represent fractions from unity. It is useful to introduce auxiliary variables $w^t, t = 1, \ldots, T+1$, denoting the amounts of melt at considered stages $w^{k_t} = \sum_{i=1}^{m_t} h_i^{k_t}$ for $t = 1, \ldots, T+1$ and $k_t \in \mathcal{K}_t$. Then, the bounds for *amounts* of elements are specified by $l_{it} w^{k_{t+1}}$ and $u_{it} w^{k_{t+1}}$. It is also necessary to define the lower bound w_L^{T+1} for the minimum amount of the produced steel and the upper bounds $w_U^t, t = 1, \ldots, T+1$, derived from the furnace capacity. The updated two-stage model (15.1)–(15.4) is

$$\text{minimize} \sum_{j \in J_2} c_j x_j^2 + \sum_{k_3 \in \mathcal{K}_3} p_{k_3} \sum_{j \in J_3} c_j x_j^{k_3} \tag{15.21}$$

subject to

$$l_{i2} w^{3E} \le \sum_{l=1}^{m_1} \tau_{il}^E \left(h_i^2 + \sum_{j \in J_2} a_{lj} x_j^2 \right) \le u_{i2} w^{3E}, \quad i = 1, \ldots, m_2, \tag{15.22}$$

$$w^{3E} = \sum_{i=1}^{m_2} \sum_{l=1}^{m_1} \tau_{il}^E \left(h_i^2 + \sum_{j \in J_2} a_{lj} x_j^2 \right), \tag{15.23}$$

$$l_{i3} w^{k_3} \le \sum_{l=1}^{m_2} \tau_{il}^{k_3} \left(h_l^2 + \sum_{j \in J_2} a_{lj} x_j^2 \right) + \sum_{j \in J_3} a_{ij} x_j^{k_3} \le u_{i3} w^{k_3}, \quad i = 1, \ldots, m_3, \quad k_3 \in \mathcal{K}_3, \tag{15.24}$$

$$w^{k_3} = \sum_{i=1}^{m_3} \left(\sum_{l=1}^{m_2} \tau_{il}^{k_3} \left(h_i^2 + \sum_{j \in J_2} a_{lj} x_j^2 \right) + \sum_{j \in J_3} a_{ij} x_j^{k_3} \right), \quad k_3 \in \mathcal{K}_3, \tag{15.25}$$

$$x_j^2 \ge 0, \quad j \in J_2, \quad x_j^{k_3} \ge 0, \quad j \in J_3, \quad k_3 \in \mathcal{K}_3, \quad w^{k_3} \ge w_L^4, \quad w^{k_t} \le w_U^{t+1}, \quad t = 2, 3. \tag{15.26}$$

This is again a fixed recourse problem based on scenarios.

Table 15.3. *Melt composition, goal intervals, expert- and measurement-based utilizations.*

	Fe	C	Mn	Si	P	S	Cr	Mo	Al	Cu
$\hat{\tau}_{ii}^2$	0.94750	0.31750	0.25000	0.05950	0.60250	0.60250	0.88250	1.00000	0.00575	0.01750
τ_{ii}^2	0.94000	0.32000	0.24000	0.05600	0.60000	0.60000	0.88000	1.00000	0.00400	0.01000
h_i^2/w^2	0.97219	0.00291	0.00110	0.00006	0.00017	0.00018	0.00664	0.00190		0.00000
l_i^1		0.0018								
u_i^1	1.0000	0.0033	0.0038	1.0000	0.0004	0.0004	1.0000	0.0025	1.0000	0.0001
$\hat{\tau}_{ii}^3$	0.98300	1.00000	0.97625	0.04250	0.98750	0.98750	0.96000	1.00000	0.42625	1.00000
τ_{ii}^3	0.98500	1.00000	0.99500	0.04700	0.98000	0.97000	0.98000	1.00000	0.44000	1.00000
h_i^3/w^3	0.96336	0.00428	0.00558	0.00005	0.00017	0.00018	0.00644	0.00237	0.00022	0.00000
l_i^2		0.00350	0.00520	0.00005			0.00500	0.00200	0.00010	
u_i^2	1.00000	0.00440	0.00900	0.00350	0.00035	0.00035	0.01300	0.00280	0.00035	0.00010
l_i^3	0.95000	0.00380	0.00600	0.00100			0.01000	0.00200	0.00010	
u_i^3	0.98000	0.00500	0.00850	0.00350	0.00030	0.00030	0.01500	0.00280	0.00040	0.00010

Before we may begin computations, we have to complete our input data set; see Table 15.3 for measurements h_i^2, bounds l_{it}, u_{it}, $t = 1, 2, 3$, and expert- and measurement-based utilizations $\hat{\tau}_{ii}^t, \tau_{ii}^t$, $i = 1, \ldots, m_t$, $t = 2, 3$. (Notice the distinction between the expert-based utilizations $\hat{\tau}_{ii}^t$ and the standard or average utilizations τ_{il}^E introduced earlier.)

Interstage goal intervals l_{it}, u_{it} $\forall i$, $t = 1, 2$ have been derived by metallurgical rules, the experience of previous melts, and the goal interval relaxation. In a foundry, $\tau_{ij}^{k_3}$ are often specified by experts. Usually, they consider only diagonal utilization matrices and keep in view one expert-based scenario, $\hat{\tau}_{ii}^t$, $i = 1, \ldots, m_t$, $t = 2, 3$, for one stage. We may try to verify expert-based utilizations, e.g., for our two-stage model, computing $\hat{\boldsymbol{T}}^3(\boldsymbol{h}^2 + \boldsymbol{A}^2\boldsymbol{x}^2) + \boldsymbol{A}^3\boldsymbol{x}^3$ to get the final composition of melt. Boldface letters $\boldsymbol{x}^t$, $\boldsymbol{h}^t$, $\boldsymbol{A}^t$, and $\hat{\boldsymbol{T}}^t$ denote vectors and matrices having components x_j^t, h_i^t, a_{ij}^t, and $\hat{\tau}_{ij}^t$, respectively.

It can be found easily that in our example with the expert utilizations $\hat{\tau}_{ii}$, the goal requirements for 42CrMo are not satisfied (see Table 15.1) and even the final measurements cannot be obtained by the described matrix multiplications (cf. Table 15.2(b)). It seems that experts' suggestions are useless and even meaningless. This motivates a more careful scenario generation. By solving the separated system of linear equations (15.16) for unknown $\tau_{ii}^{k_3}$, with the known compositions a_{ij}, different amounts of inputs $x_j^{a(k_3)}$, and varying results of measurements $h_i^{a(k_3)}$ and $h_i^{k_3}$,

$$h_i^{k_3} = \tau_{ii}^{k_3}\left(h_i^{a(k_3)} + \sum_{j \in J_2} a_{ij} x_j^{a(k_3)}\right), \quad i = 1, \ldots, m_2, \quad k_3 \in \mathcal{K}_3, \tag{15.27}$$

we obtain a measurement-based utilization τ_{ii}^3 which corresponds to one of scenarios $\tau_{ii}^{k_3}$. In a similar way, we get also measurement-based utilizations $\tau_{ii}^{k_2}$. In this way, we may trace realized trajectories of the considered production process. Table 15.4 presents four different scenarios of $\tau_{ii}^{k_2}$ (indexed by superscripts $k_2 = 1, \ldots, 4$) and four different scenarios $\tau_{ii}^{k_3}$

Table 15.4. *Measurement-based scenarios of utilizations—sets $\mathcal{K}_2$ and $\mathcal{K}_3$.*

Scenario	Fe	C	Mn	Si	P	S	Cr	Mo	Al	Cu	z^1_{HB}	$z^1_{SB_1}$
τ^1_{ii}	0.937	0.35	0.192	0.061	0.58	0.66	0.88	1.00	0.005	1.00	70610	55065
τ^2_{ii}	0.941	0.37	0.199	0.057	0.55	0.69	0.86	1.00	0.004	1.00	70040	56508
τ^3_{ii}	0.952	0.40	0.21	0.047	0.57	0.55	0.89	1.00	0.003	1.00	69359	54149
τ^4_{ii}	0.946	0.43	0.183	0.058	0.59	0.64	0.87	1.00	0.005	1.00	69326	57728
τ^5_{ii}	0.989	1.00	0.98	0.041	0.98	0.96	0.978	1.00	0.55	1.00	12351	4766
τ^6_{ii}	0.983	1.00	0.985	0.037	0.96	0.95	0.965	1.00	0.55	1.00	12725	5193
τ^7_{ii}	0.992	1.00	0.981	0.040	0.97	0.955	0.981	1.00	0.55	1.00	11680	4469
τ^8_{ii}	0.987	1.00	0.983	0.038	0.95	0.94	0.969	1.00	0.55	1.00	11466	5868

Table 15.5. *Comparison of optimal solutions.*

Inputs	$x^2_{j,HB}$	$x^2_{j,SB_1,\min}$	$\hat{x}^2_{j,SB_1,\min}$	$x^2_{j,SB_4,\min}$	$x^2_{j,SB_{4096},\min}$	$x^2_{j,SB_{cov},\min}$
Coke	20.0	7.0	6.8	16.9	17.6	17.3
FeMo	11.0	1.6	1.5	2.1	2.1	2.1
FeSiMn	100.0	87.5	89.4	105.0	107.7	106.0
AlInput	8.0	3.6	3.7	4.0	4.1	4.0
Objective function	z^2_{HB} 12351	$z^2_{SB_1,\min}$ 4766	$\hat{z}^2_{j,SB_1,\min}$ 4813	$z^2_{SB_4,\min}$ 6140	$z^2_{j,SB_{4096},\min}$ 7896	$z^2_{j,SB_{cov},\min}$ 6920

(indexed by superscripts $k_3 = 5, \ldots, 8$) derived from four melt reports, along with the related costs of the alloying and of the whole melting process.

Because our goal is to find another, cheaper way to produce the same steel, we allow the "computer knowing our two-stage program" to choose other inputs arbitrarily feasible but with the lowest cost; see Table 15.5 for results.

We also know the cost of the whole realized process and its part related to our two-stage problem, i.e., $z^2_{HB} = 12{,}351$ (where HB is "history based" and the superscript 2 refers to the current initial alloying second stage). It may be compared with one-scenario-based (deterministic) optimization solution $z^2_{SB_1,\min} = 4766$ obtained from the (15.21)–(15.26) model for $\mathcal{K}_3$ set having the only element. (Here and in what follows, the index SB_s denotes scenario-based problems using s scenarios.) In this case, knowing losses in advance and utilizing our model, a melter could save more than 61% of alloying costs. This looks surprising, as the alloying stages could be considered as a source of minor changes regarding the initial charge. However, we must remember that we have obtained the best solution given complete foresight, whereas in reality, losses are uncertain and unknown in advance. Therefore, we will apply now the true scenario-based stochastic programming model. At first, we assume that only a few melt reports for steel 42CrMo are available, namely, those which were used to get the four scenarios of utilizations $\tau^{k_3}_{ii}$, $k_3 = 5, \ldots, 8$, from Table 15.4. Assume that probabilities of scenarios are equal, so $p_5 = \cdots = p_8 = 1/4$. Using GAMS/OSL again to solve the program (15.21)–(15.26) with the new data set, we obtain $z^2_{SB_4,\min} = 6140$ (savings 50.28%; see Table 15.5 for the optimal solution). We see

that $z^2_{SB_1,\min} \leq z^2{}_{SB_4,\min} \leq z^2{}_{HB,\min}$ in our example. However, computing the expected cost with respect to the four given scenarios for solution $x^2_{SB_1,\min}$, i.e., checking first the feasibility in (15.21)–(15.26) and then computing the objective function values for recourse actions $x_j^{k_3}$ obtained by the solution of separate linear programs, we get $z^2_{SB_4}(x^2_{SB_1,\min}) = 9421$. This 23.72% savings in comparison with the melter's decision (which was bad indeed in this case, but useful for illustration of our scenario-based approach) shows that even the deterministic approach may bring help in melt control, and it illustrates why the scenario approaches are advantageous. We may also return to expert-based utilizations $\hat{\tau}^t_{ii}$ which were marked as wrong for computations along individual scenarios. Using them for the two-stage problem in place of the measurement-based τ^t_{ii}, we get the related optimal value $\hat{z}^2_{SB_1,\min} = 4813$, less optimistic than $z^2_{SB_1}$ but the corresponding expected cost $z^2_{SB_4}(\hat{x}^2_{SB_1,\min}) = 9024$ means savings of 26.93%. Hence, we can understand now why metallurgists are overestimating their expert-based estimates of utilizations. They intuitively use the worst-case approach to avoid the surprise coming with the overfilled furnace. In a certain sense, their aggregated values for $\hat{\tau}^t_{ii}$ are chosen to find a robust decision. Although we understand the basic idea, we must say that the reasoning is wrong, as we have a better scenario-based model.

At this moment, we have only four melt reports. A reasonable question is whether we may significantly increase the number of scenarios before more information about further melts is available. The simplest step is to assume independent random losses within the alloying stage. Therefore, we may create new scenarios easily just by combining all elements utilizations. The number of derived scenarios becomes large, equaling $4^6 = 4096$ as four utilizations remain constant, independent of the scenario changes; see Table 15.4. Several solvers (CPLEX, OSL, OSLSE) have been tested with the GAMS source code. The results are the same: we obtain $z_{SB_{4096},\min} = 7896$; however, different amounts of computing time were needed. In this case, we can see even from the visual analysis of scenarios in Table 15.4 that there might be some nonzero correlations among utilizations. Thus, our instage-independence-based scenario set is built in a too-defensive way: we skipped available information, and hence we considered unrealistic scenarios and their recourse costs were taken into account. As a result we obtained a too-pessimistic solution. Still, the obtained solution may be implemented and used as it gives a good chance to decrease the melting costs. However, there is a bottleneck. This approach is useless for a large set of scenarios because computations for the alloying stage should be realized in real time, i.e., at most during tens of seconds. For this purpose, scenario set reduction techniques have been developed. (See [12] for the application of principal components and [11] for identification of so-called extreme scenario sets.) In this paper we try to remove the unrealistic independence assumption and to apply another approach—the moment fitting procedure from subsection 15.3.1. As the number of scenarios is quite small (four melt reports until now) to obtain reasonable values for covariances, we may exploit general metallurgical laws and experience. The idea is that the relationships among utilizations are quite general, and they do not vary too much when similar steels are produced and similar technologies are used. Therefore, we set the values of ρ_{il} analyzing similar steel melt reports, and we roughly estimate μ_i and σ_i^2 from 42CrMo steel melt reports. We have eight new scenarios minimizing the objective (15.17) from subsection 15.3.1 (with equal weights) under additional experience-based constraints saying that scenarios have nearly the same probabilities and that the utilizations are bounded below and above by the existing extremal cases of their values derived from melt reports.

Table 15.6. *Fitted scenarios and probabilities.*

Scenario	Fe	C	Mn	Si	P	S	Cr	Mo	Al	Cu	p_k
τ_{ii}^1	0.989	1.00	0.98	0.041	0.98	0.956	0.968	1.00	0.55	1.00	0.628
τ_{ii}^2	0.983	1.00	0.985	0.037	0.96	0.945	0.965	1.00	0.55	1.00	0.623
τ_{ii}^3	0.992	1.00	0.981	0.040	0.97	0.952	0.971	1.00	0.55	1.00	0.620
τ_{ii}^4	0.987	1.00	0.983	0.038	0.95	0.949	0.969	1.00	0.55	1.00	0.630
τ_{ii}^5	0.988	1.00	0.982	0.042	0.98	0.961	0.978	1.00	0.55	1.00	0.626
τ_{ii}^6	0.984	1.00	0.984	0.039	0.95	0.953	0.975	1.00	0.55	1.00	0.625
τ_{ii}^7	0.993	1.00	0.981	0.040	0.97	0.955	0.981	1.00	0.55	1.00	0.626
τ_{ii}^8	0.987	1.00	0.983	0.037	0.96	0.944	0.979	1.00	0.55	1.00	0.622

The obtained scenarios are listed in Table 15.6 and the optimal value is 6920; see the last column of Table 15.5.

At this moment we may consider what to do when more melt reports are available. At first glance, it seems that there is no necessity to build artificial scenarios as the number of scenarios is growing; they are related to historical melts and they may be considered as representative enough. However, after one year with several melts per day we face again the question of how to reduce the number of scenarios under real-time restrictions. Hence, the approaches of subsection 15.3.1 and those suggested in [12] or [11] remain useful. The list may be completed by methods based on random sampling from the given huge set of scenarios; see, e.g., [13].

15.4.2 Three-stage melt control

As the next step we continue with a three-stage model. We want to include optimization of the first step of 42CrMo production. One possibility is reduction of the number of stages to two, simply thinking about all alloying steps incorporated in one stage only. In this case, we optimize the charge under rather rough forecasting of all further alloying consequences. This is definitely better than the one-stage model, and it is comparable with another approximating two-stage model whose horizon is restricted to the end of the first alloying stage. As we have discussed before, because the charge can be computed in advance, we are not restricted by the real-time requirements and our model may be larger. Again we use the update of the previously introduced model (15.5)–(15.10) incorporating melt weights w^{k_t} and relative goal intervals l_{it} and u_{it}. At the beginning, we assume that the scrap composition is fixed to the midpoints of intervals.

The resulting form of the model is

$$\text{minimize} \sum_{j\in J_1} c_j x_j^1 + \sum_{t=2}^{3} \sum_{k_t \in K_t} p_{k_t} \sum_{j\in J_t} c_j x_j^{k_t} \tag{15.28}$$

subject to

$$\sum_{l=1}^{m_{t-1}} \tau_{il}^{k_t} \left(h_l^{a(k_t)} + \sum_{j \in J_{t-1}} a_{lj} x_j^{a(k_t)} \right) - h_i^{k_t} = 0, \quad i = 1, \ldots, m_{t-1}, \quad k_t \in \mathcal{K}_t, \quad t = 2, 3, \tag{15.29}$$

$$l_{it-1} \sum_{k_t \in \mathcal{K}_t} p_{k_t} w^{k_t} \le \sum_{k_t \in \mathcal{K}_t} p_{k_t} h_i^{k_t} \le u_{it-1} \sum_{k_t \in \mathcal{K}_t} p_{k_t} w^{k_t}, \quad i = 1, \ldots, m_{t-1}, \; t = 2, 3, \tag{15.30}$$

$$l_{i3} w^{k_4} \le h_i^{k_3} + \sum_{j \in J_3} a_{ij} x_j^{k_3} \le u_{i3} w^{k_4}, \quad i = 1, \ldots, m_3, \tag{15.31}$$

$$w^{k_t} = \sum_{i=1}^{m_t} h_i^{k_t}, \quad k_t \in \mathcal{K}_t, \quad t = 1, \ldots, 4, \tag{15.32}$$

$$x_j^1 \ge 0, \quad j \in J_1, \quad x_j^{k_t} \ge 0, \quad k_t \in \mathcal{K}_t, \quad j \in J_t, \quad t = 2, 3, \tag{15.33}$$

$$w^{k_3} \ge w_L^4, \quad w^{k_t} \le w_U^{t+1}, \quad t = 1, 2, 3. \tag{15.34}$$

With the charge, special metallurgical constraints may be added, from the simplest bounds on the scrap amount or on the sum of amounts of return materials to the additional constraints based on linear functions of the melt composition (restricted amount of the sum of phosphorus and sulfur, satisfactory C-equivalent, etc.; see [10]). In general, all these constraints can be modeled by a polyhedral set $\mathcal{X}$. Therefore, we do not present them explicitly, although they were utilized in computations.

We are ready now to present computational results. The input data are again taken from Tables 15.1–15.4. At first, we get $z^1_{HB,\min} = 70{,}601$ as a consequence of the melt report information. Then, knowing melt-related inputs $x_j^t, j \in J_t, t = 2, 3$, and measurement results $h_i^t, i = 1, \ldots, m_t, t = 2, 3$, solving two systems of separate linear equations (15.16), we can easily identify the measurement-based scenario of utilizations related to the melt report in question, which consists of *two* diagonal matrices with elements $\tau_{ii}^{k_t}, i = 1, \ldots, m_t, t = 2, 3$. In what follows, we solve program (15.28)–(15.34) for this scenario and obtain the best single scenario cost $z^1_{SB_1,\min} = 55{,}065$. Similarly, we may obtain results for the remaining three scenarios. The last two columns of Table 15.4 indicate that we may save between 16% and 22%. However, further computations are provided in Table 15.7, which is based on analysis analogous to that in Table 15.5 for the two-stage model. The conclusion is quite clear; real savings with just the one-scenario approach will be significantly lower. Therefore, we utilize approaches developed in section 15.3.

Using the four melt reports we may derive data for the four-scenario-based model. Exploitation of these melt reports on the complete melt implies that there is a one-to-one correspondence of the terminal nodes indexed by $5, \ldots, 8$ and their ancestors, $a(k) = k - 4$, $k = 5, \ldots, 8$, elements of $\mathcal{K}_2$ listed in Table 15.4. The computation based on the fan of these four scenarios will return the total expected cost of the production process (consisting now of the charge and alloying), $z^1_{SB_4,\min} = 59{,}601$. As in subsection 15.4.1, we may compute values $z^1_{SB_4}(x^1_{HB}) = 66{,}934$ for the historical first-stage solution, $z^1_{SB_4}(x^1_{SB_1,\min}) = 63{,}121$ for the optimal first-stage solution based on one scenario, and $z^1_{SB_4}(\hat{x}^1_{SB_1}) = 62{,}115$ for

Table 15.7. *Comparison of optimal solutions.*

Inputs	$x^1_{j,HB}$	$x^1_{j,SB_1,\min}$	$\hat{x}^1_{j,SB_1,\min}$	$x^1_{j,SB_4,\min}$	$x^1_{j,SB_{16},\min}$	$x^1_{j,SB_{cov},\min}$	$x^1_{j,SB_{ri,cov},\min}$
RM955	6000.0	5780.3	5830.6	5902.1	6172.6	5930.3	6010.6
T951	5000.0	4900.5	4930.5	4950.1	5120.1	5200.1	5150.5
Scrap	5400.0	5600.2	5809.7	5510.0	5107.7	4960.0	4820.3
Coke	210.0	180.4	200.3	190.0	198.1	192.0	193.0
Objective function	z^1_{HB} 70601	$z^1_{SB_1,\min}$ 55065	$\hat{z}^1_{j,SB_1,\min}$ 57112	$z^1_{SB_4,\min}$ 59601	$z^1_{j,SB_{16},\min}$ 61396	$z^1_{j,SB_{cov},\min}$ 60020	$z^1_{j,SB_{ri,cov},\min}$ 60997

the optimal first-stage solution based on one scenario of expert-based utilizations. For the related first-stage optimal solutions $\boldsymbol{x}^1$ and some additional results, see Table 15.7.

The conclusion is similar to that in section 15.4.1. The use of optimization, even for the deterministic (one-scenario) problem, is better then the melter's intuition and experience (9.18% savings in our example). The practitioners may achieve some improvement using the expert-based scenario (10.18% savings) to hedge against uncertainty. Nevertheless, the numerical results again provide evidence that the proposed scenario-based stochastic programming approach is the right way to hedge against uncertainty (15.58% savings for four scenarios).

The next questions are how to increase the number of scenarios and how to create a nontrivial scenario tree to capture the multistage decision structure of the problem. The first step could be simple. As discussed in subsection 15.3.2, we may assume interstage independence. Accordingly, we may generate a scenario tree with $4^2 = 16$ scenarios from Table 15.4. The result is $z^1_{SB_{16,\min}} = 61{,}396$ (12.42% savings).

In practice, the interstage dependence has not yet been analyzed, so it is hard to get reliable experts' opinions. Common sense says that if random utilizations of certain elements have been high for the first stage, this may be caused by a short time of heating, and the nonrealized loss may be realized during the alloying phase. Hence, negative correlations between stage-related utilizations may occur. We shall apply again the general idea of the moment fitting approach, assuming for simplicity that the instage structure of utilizations is fully specified by the related melt report and also that there is no dependence between losses of different elements belonging to different stages. As in subsection 15.4.1, we use the moment conditions on utilizations in both stages and on their interstage correlations. In the fitting objective function, such as (15.17), we put a lower weight (25%) on terms fitting interstage covariances. The results of our computations are listed in Table 15.8.

The next to last column of Table 15.7 indicates that some improvement of the objective function value may be expected. The question is the reliability of our estimates on interstage dependence characteristics.

The next step is to consider the random input. We know from Table 15.1 that this concerns only the scrap. We may extend our model (15.28)–(15.34) adding first-stage input scenarios using a modification of (15.11)–(15.14) to the three-stage problem.

We consider only random content of chromium (Cr), as it is the important element for the final composition, and iron (Fe), as it is the main part of the alloy and it indirectly

Table 15.8. *Fitted scenarios and probabilities for a* 4×2 *tree.*

	Fe	C	Mn	Si	P	S	Cr	Mo	Al	Cu	p_k
τ_{ii}^1	0.981	1.00	0.978	0.040	0.99	0.96	0.961	1.00	0.55	1.00	0.272
τ_{ii}^2	0.978	1.00	0.985	0.037	0.96	0.945	0.965	1.00	0.55	1.00	0.230
τ_{ii}^3	0.989	1.00	0.979	0.040	0.97	0.952	0.971	1.00	0.55	1.00	0.210
τ_{ii}^4	0.982	1.00	0.983	0.038	0.95	0.949	0.969	1.00	0.55	1.00	0.288
τ_{ii}^5	0.988	1.00	0.982	0.042	0.98	0.956	0.977	1.00	0.55	1.00	0.526
τ_{ii}^6	0.984	1.00	0.978	0.039	0.94	0.951	0.975	1.00	0.55	1.00	0.474
τ_{ii}^7	0.991	1.00	0.984	0.041	0.97	0.952	0.978	1.00	0.55	1.00	0.591
τ_{ii}^8	0.987	1.00	0.987	0.038	0.96	0.947	0.976	1.00	0.55	1.00	0.409
τ_{ii}^9	0.982	1.00	0.981	0.042	0.98	0.961	0.978	1.00	0.55	1.00	0.544
τ_{ii}^{10}	0.985	1.00	0.986	0.040	0.97	0.953	0.975	1.00	0.55	1.00	0.456
τ_{ii}^{11}	0.993	1.00	0.982	0.041	0.97	0.955	0.981	1.00	0.55	1.00	0.601
τ_{ii}^{12}	0.987	1.00	0.985	0.035	0.96	0.944	0.979	1.00	0.55	1.00	0.399

describes the variation of the remaining elements within the scrap. We create $3^2 = 9$ equiprobable scenarios for scrap composition using ideas of subsection 15.3.3 (scenarios specified by $a_{ij}^E - \delta_{ij}$, a_{ij}^E, $a_{ij}^E + \delta_{ij}$, with $\delta_{ij} = 2/3(\overline{a}_{ij} - a_{ij}^E)$). Finally, we combine these nine scenarios with eight scenarios of utilizations obtained by the moment fitting procedure listed in Table 15.8. The scrap composition data are utilized from Table 15.1. Bounds $\underline{a}_{ij}$ and $\overline{a}_{ij}$ are given by the related rows denoted by L and U. The choice of the coefficient 2/3 was preferred because it generates more distinct scenarios than other possibilities.

Analyzing results from Table 15.7, we may conclude that considering the random composition of the scrap we have significantly decreased the risk of its use (cf. inputs for other scenario-based models and notice the difference in the scrap input).

The results obtained based on 72 scenarios ($z^1_{SB_{ri,cov,\min}} = 60{,}997$ and savings 13.60%; see the last column of Table 15.7) are considered as the most realistic and numerically tractable representation of uncertainty achieved so far.

We emphasize that for illustration purposes we have chosen several melt reports of a beginning melter. Because he was significantly unsuccessful, we have the opportunity to explain many different aspects with just one data set. Although data from experienced melters are "more boring," numerical experience shows that also in these cases we may save a significant amount of money. The one-scenario models usually forecast savings of about 12%, whereas the real savings decrease to 3%. With the scenario models discussed above it is quite realistic to expect real savings between 5% and 10%.

Regarding the modeled technological process, the obtained solutions must be interpreted as suboptimal only. Nevertheless, they are significantly better than solutions obtained by the contemporary techniques used in melt control.

In this context there is also an interesting interpretation of EVSI (expected value of scenario information; see [1]) for the charge problem: we may compare whether the costs of sorting scrap are less than the attained profit. This means computing the model for unsorted

scrap and comparing the results for the case when the scrap composition is fully known. In our case this means distinguishing different types of input scraps with small intervals of uncertainty for a_{ij}.

15.5 Discussion and extensions

The scope of this paper has been mostly restricted to the iron production problem without inclusion of technological and storage constraints. Neither environmental aspects nor the quality of production were taken into account. In principle, without any problems the model may be extended for additional deterministic constraints and, following ideas of multiobjective decision making, the objective function may be augmented for an additional term related to pollution limitations or to metal quality.

The models and their implementation were based on several assumptions, such as diagonal utilization matrices and fully specified input composition, interstage independence of utilities, and their independence of the input quantities. Some of these assumptions were relaxed in the discussed scenario generation procedure, which in turn not only made use of historical measurements but also exploited experts' opinion. As for other applications of scenario-based stochastic programs, no recipe for the best scenario generation and selection procedure exists. We observed that simple, sound discretization procedures gave different scenarios. Sensitivity of the results (of the minimal costs and the best initial charge) with respect to the selected scenarios and their robustness related to various simplifying assumptions should be carefully analyzed. Bounds based on the best- and worst-case analysis delineated briefly in subsection 15.3.3 may help.

At the end, three specific problem and model properties of the stochastic program in question have to be underlined:

1. Stages are not defined by modeler's choice because they are given by the modeled production process.

2. Because the filled furnace cannot be enlarged or emptied during the process (contrary to the assumed unlimited borrowing and lending possibilities in financial applications, for example), the related hard constraints imply that relatively complete recourse cannot be assumed. Hence, feasibility of the first-stage solution must be analyzed.

3. Computations related to the alloying stages should be realized in real time, and this asks for a numerically tractable scenario generation procedure which results in a relatively small number of representative scenarios.

Acknowledgments

This work was partly supported by research projects Mathematical Methods in Stochastics MSM 113200008 and MSMT CEZ: J22/98:261100009 and by Grant Agency of the Czech Republic grants 106/01/1464, 201/02/0621, and 402/02/1015.

Bibliography

[1] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer, New York, 1997.

[2] D. R. Cariño, D. H. Myers, and W. T. Ziemba, *Concepts, technical issues and uses of the Russell-Yasuda Kasai financial planning model*, Oper. Res., 46 (1998), pp. 450–462.

[3] J. Dupačová, *The minimax approach to stochastic programming and an illustrative application*, Stochastics, 20 (1987), pp. 73–88.

[4] J. Dupačová, G. Consigli, and S. W. Wallace, *Scenarios for multistage stochastic programs*, Ann. Oper. Res., 100 (2000), pp. 25–53.

[5] J. Dupačová, J. Hurt, and J. Štěpán, *Stochastic Modeling in Economics and Finance*, Part II, Kluwer Academic Publishers, Norwell, MA, 2003.

[6] W. H. Evers, *A new model for stochastic linear programming*, Management Sci., 13 (1967), pp. 680–693.

[7] K. Høyland and S. W. Wallace, *Generating scenario trees for multistage decision problems*, Management Sci., 47 (2001), pp. 295–307.

[8] G. Pflug, *Scenario tree generation for multiperiod financial optimization by optimal discretization*, Math. Program. B, 89 (2001), pp. 251–271.

[9] P. Popela, *An Object Oriented Approach to Multistage Stochastic Programming: Models and Algorithms*, Ph.D. thesis, Charles University, Prague, 1998.

[10] P. Popela, *Application of stochastic programming in foundry*, Folia Fac. Sci. Natur. Univ. Masaryk. Brun. Math., 7 (1998), pp. 117–139.

[11] P. Popela and J. Roupec, *GA-based scenario set modification in two-stage melt control problems*, in Proceedings of the 8th International Conference MENDEL 99, Brno, Czech Republic, 1999, pp. 112–117.

[12] P. Popela and J. Zeman, *Principal component-based scenario reduction in two-stage melt control problems*, in Proceedings of the 8th International Conference MENDEL 99, Brno, Czech Republic, 1999, pp. 191–198.

[13] A. Shapiro and T. Homem-de-Mello, *A simulation based approach to two-stage stochastic programming with recourse*, Math. Program., 81 (1998), pp. 301–325.

This page intentionally left blank

Chapter 16

A Stochastic Programming Model for Network Resource Utilization in the Presence of Multiclass Demand Uncertainty

Julia L. Higle and Suvrajeet Sen**

16.1 Introduction

There are numerous applications in which revenues are generated by the use of resources that are distributed over a network. In some cases, these networks are spatial, while in others they are temporal. Nodes in a spatial network, such as those in air transportation and telecommunications industries, correspond to locations on the network, and arcs correspond to the ability to transport goods or provide services between nodes. On the other hand, temporal networks are formed by discretizing time and are commonly used for yield management models for automobile rental companies, hotels, etc. In these models, nodes are often associated with points in time, and arcs correspond to bookings over time. In either case, it is important to recognize that demand is often served by using resources associated with multiple arcs of the network. Airline customers may use multiple flights to complete their itineraries, calls may be routed across multiple links in a telecommunication network, and rental car and hotel customers may retain facilities for multiple days. Furthermore, these networks typically serve multiple classes of customers, some of whom pay higher rates than others. For example, if a television network has a "breaking" story for which video conferencing is necessary immediately, they may be willing to pay at a higher rate than a university that has paid in advance to transmit lectures over the same network. Similarly, customers in the airline industry are categorized by fare classes, as are hotel and car rental customers. In any of these applications, the revenue generated by the network depends, in large measure, on the admission control policy used for network management. Intuitively, good control policies will result in a system that serves as many high-paying customers as possible, while maintaining a high level of resource utilization.

This paper introduces models that may be used to facilitate the efficient management

*SIE Department, University of Arizona, Tucson, AZ 85721 (julie@sie.arizona.edu, sen@sie.arizona.edu).

of resources distributed over a network. Demand flows between origination-destination (O-D) pairs and forecast uncertainty are incorporated. Our stochastic programming (SP) model has many of the advantages of linear programming (LP) models. Our primary model is a two-stage model, in which the initial allocation of resources is revised based on the nature of the demands observed in the second stage. The latter model is a linear program. Thus, unlike leg-based approaches (see [3, 5]), our model incorporates network effects in a manner reminiscent of a deterministic LP (see [9]). We use a sampling-based method which allows the randomness to be represented in a general form, including histograms, simulations, etc. [10]. Finally, we discuss approximations that reflect the differing arrival patterns of the alternate customer classes.

The paper begins with a review of two widely referenced network models, namely, the deterministic LP (DLP) and a probabilistic nonlinear program (PNLP), which is a simple recourse problem. While PNLP poses severe restrictions, it helps to set the stage for our model development. In section 16.3, we propose a stochastic optimization approach for generating bid prices. The approaches are compared via a simulation in section 16.4. The SP-based bid prices yield higher returns than the LP model. The advantages of the SP model apparently diminish as the number of fare classes increases.

16.2 Related models for network resource utilization

In this section we discuss two networkwide formulations that have been proposed in the literature. Such models are used to capture the revenue impact of several O-D pairs. The marginal values associated with the resources are then used to determine whether a particular customer is to be accepted or rejected. In the yield management literature, such a policy is referred to as a "bid-price" control policy, the marginal values being the bid-prices. Despite the wide applicability of the class of models under consideration, it is the airline industry that has led the development and deployment of these models. Consequently, our discussion borrows terms from the airlines. Various classes of customers request service along paths through the network. We use the term "itinerary" to specify a path and class combination. A path is composed of a sequence of "legs," each of which has a specified capacity which may be allocated among the various customer classes.

In the DLP formulation, demand for each itinerary is assumed to be known. With each itinerary (indexed by i), we associate an allocation x_i which denotes the quantity of resource allocated to itinerary i. The path associated with itinerary i is specified by an incidence vector A_i consisting of as many elements as there are legs in the network. If leg ℓ belongs to the path associated with itinerary i, then the corresponding element $a_{\ell,i}$ is 1; otherwise it is 0. With each itinerary we also associate unit revenues, denoted v_i. Finally, the vector of available leg capacities is denoted by C, with each element denoted c_ℓ. Letting $\bar{d}_i$ denote the demand forecast for itinerary i, the DLP formulation is

$$\begin{aligned} \text{Max} \sum_i v_i x_i \\ \text{s.t.} \quad \sum_i A_i x_i \le C, \qquad & \text{(DLP)} \\ 0 \le x_i \le \bar{d}_i \quad \forall i. \end{aligned}$$

Let λ denote the Lagrange multiplier associated with the capacity constraints. Given the incidence vector for itinerary i, A_i, $\lambda^\top A_i$ denotes the total marginal value associated with the path. A bid-price control policy accepts all those customers who are willing to pay greater than $\lambda^\top A_i$, provided capacity is available. When applied with frequent reoptimization, this simple rule has outperformed a variety of other policies [18, 25].

The demands $\bar{d}_i$ used in the DLP model are often expected values of a random variable $\tilde{d}_i$. When demand is a random variable, one might consider the PNLP model which has appeared previously in various forms [18]. The PNLP model is

$$\begin{aligned} \text{Max} \sum_i v_i E[\text{Min}\{x_i, \tilde{d}_i\}] \\ \text{s.t.} \quad \sum_i A_i x_i \le C, \qquad \text{(PNLP)} \\ x_i \ge 0 \quad \forall i. \end{aligned}$$

Here $E[\ ]$ denotes the expectation operator. If $\tilde{d}_i = \bar{d}_i$ for all itineraries i, then solutions to DLP also solve PNLP. When demands are random (so that $\tilde{d}_i \neq \bar{d}_i$ in general), the model fails to permit an adaptive response to the resolution of uncertainty. That is, in cases where $\tilde{d}_i < x_i$, the model does not capture the opportunity to reallocate the unused capacity to other itineraries. In this case, the propensity toward unused capacity is clear, so that this model may be too rigid for revenue maximization. In fact, [25] has demonstrated that the DLP model provides better bid-prices than the PNLP model.

To examine this inadequacy of the PNLP model, and to make the transition to a more general class of stochastic programming formulations, we rewrite PNLP as a simple recourse problem (see [23]),

$$\begin{aligned} \text{Max} \sum_i \{v_i x_i - E[h_i(x_i, \tilde{d}_i)]\} \\ \text{s.t.} \quad \sum_i A_i x_i \le C, \qquad \text{(SRSP)} \\ x_i \ge 0 \quad \forall i, \end{aligned}$$

where

$$\begin{aligned} h_i(x_i, d_i) = \text{Min}\, v_i\, z_i^- \\ \text{s.t.} \quad z_i^+ - z_i^- = d_i - x_i, \\ z_i^+, z_i^- \ge 0. \end{aligned}$$

Note that $h_i(x_i, d_i) = v_i\,\text{Max}(0, x_i - d_i)$ and

$$\begin{aligned} v_i x_i - E[h_i(x_i, d_i)] &= v_i x_i + v_i E[\text{Min}(0, \tilde{d}_i - x_i)] \\ &= v_i E[\text{Min}(x_i, \tilde{d}_i)]. \end{aligned}$$

Thus the SRSP model is equivalent to PNLP. This formulation of PNLP has a great deal in common with one of the oldest SLP formulations, which appeared in [7].

The value function of an LP (in "Min" form) is convex, and consequently, PNLP/SRSP is a convex two-stage program with separable subproblems. While this is an attractive property, the model itself is inadequate. The excess capacity when $d_i - x_i < 0$ is not appropriately reallocated to other itineraries. The discrepancy caused by this approximation leads to poor revenue approximations and poor bid-prices. Talluri and van Ryzin [18] provide an interesting example of inconsistencies that might arise by using PNLP for bid-price calculations. We now proceed to extend these formulations in an effort to identify one without excessive computational demands that is more faithful to operating realities.

16.3 An SP model

We now propose an SP model that overcomes some of the shortcomings of the previously proposed approaches, DLP and PNLP. Our model provides a more logical representation of the allocation of unused capacity. It is a two-stage model in which the first stage allocates capacity to various customer classes, and the second stage provides a way to model capacity utilization. The objective function of the first stage represents expected revenues, and the constraints associated with this (first) stage represent leg capacities for the network. The Lagrange multipliers associated with these first-stage constraints represent marginal values associated with each leg and are used as bid-prices. The second-stage model of capacity utilization is an LP that has a lot in common with the DLP model. The model is discussed below.

Let L denote the set of legs, and for each $\ell \in L$, c_ℓ denote the capacity associated with leg ℓ. Let S denote the set of customer classes and, for $(s, \ell) \in S \times L$, $y_{s,\ell}$ denote the capacity on leg ℓ allocated to class s. The vector of allocations is denoted as y. The primal decision variables, y, will not play a role in executing the control itself; however, constraints on these variables will help us evaluate the marginal value of capacity, which in turn will be used for bid-price control.

To model uncertainty in customer demand for itinerary i, we use the random variable $\tilde{d}_i$, which represents the total number of requests for capacity. The vector of demands is represented as $\tilde{d}$. Finally, let $h(y, d)$ represent the revenue generated given an initial allocation y when demand is d. The problem is

$$\text{Max } E[h(y, \tilde{d})] \tag{SLP}$$

$$\text{s.t.} \quad \sum_s y_{s,\ell} \le c_\ell \quad \forall \ell \in L, \tag{16.1}$$

$$y \ge 0,$$

If we are able to approximate $h(y, d)$ well, then the Lagrange multipliers associated with the capacity constraints (16.1) may be used to provide bid-prices. The function h may be modeled using a second LP, in which revenues are maximized subject to the constraints imposed by the initial allocations, y, and the demand, d. Let

- $f_i \equiv$ the number of customers served on itinerary i.

The parameters used in the model are as follows:

- $I \equiv$ the set of all itineraries;

- $L \equiv$ the set of all legs in the network;
- $I_\ell \equiv$ the set of itineraries that use leg $\ell \in L$;
- $s(i) \equiv$ the class associated with itinerary i;
- $v_i \equiv$ the revenue generated per customer served on itinerary i.

An LP model for revenue maximization is

$$h(y, d) = \text{Max} \sum_{i \in I} v_i f_i \tag{16.2}$$

$$\text{s.t} \quad \sum_{\{i \in I_\ell | s(i) = s\}} f_i \leq y_{sl} \quad \forall s \in S, \quad \ell \in L, \tag{16.3}$$

$$0 \leq f_i \leq d_i \quad \forall i \in I. \tag{16.4}$$

At this point, comments on the differences between our SLP formulation and the DLP formulation are in order. First, DLP and PNLP model allocations based on itineraries (route-class combinations), while SLP models allocations based on leg-class combinations. This provides, to some extent, the ability to reallocate spare capacity when it is available. That is, allocated leg capacity is available to all routes using that leg. However, this reallocation remains restricted by customer class so that unused capacity allocated to one customer class is not available to other customer classes. Second, each leg has one capacity contraint in DLP and PNLP, whereas in problem (16.2)–(16.4), each leg has a separate constraint for each class.

16.4 The simulation experiment and computational results

Although our SP model addresses some of the shortcomings in the DLP and PNLP models, it retains several of their simplifications. For example, all of the models permit noninteger allocations. Similarly, none of them account for the fact that demands for different classes arrive at different rates during the booking period. It is important to study the collective impact of these simplifications on the resulting pricing scheme and the manner in which it influences the revenues that are ultimately generated.

16.4.1 Simulation description

The simulation experiment was designed to imitate the process of revising the bid-prices over a period of time as revised estimates of the projected demands become available. That is, using the initial demand projections, bid-prices are obtained by solving both the DLP and SLP described in section 16.2 and 16.3. Using these prices, tickets are sold until some time in the future when the demand projections are updated. At that time, new bid-prices are recomputed and the process is continued.

Both DLP and SLP model all itinerary demands as being revealed at the same time. This is not consistent with actual demand patterns. In reality, some fare classes are available

only in certain periods (e.g., 21-day advance tickets cannot be purchased after a particular date). Still other fare classes typically see demand only in certain periods (e.g., full-fare tickets are always available but are purchased most often within a few days of the departure date). Thus, in our simulation model, customer demand patterns are time varying. That is, depending on the fare class involved, customer arrivals are permitted to be more heavily concentrated in some periods than in others. Our initial experiment involves two classes, with class 1 customers paying a higher fare than class 2 customers. All class 1 customers arrive within 30 days of the departure date, with 75% arriving within 14 days of departure. Among the class 2 customers, 5% arrive between 180 and 120 days prior to departure, 20% arrive between 120 and 60 days prior to departure, and 75% arrive between 60 and 30 days prior to departure. No class 2 customers arrive within 30 days of departure. Customer arrival times are randomly generated within these time segments. Ticket sales are decided upon customer arrival, so that lower-fare customers may preempt higher-fare sales when bid prices are low. Our simulation spans the period of 180 days prior to departure of the flights in question. Within the 180-day window being simulated, demand projections were updated on days 0, 60, 90, 120, 135, 150, and 165. For lack of a better update mechanism, we simply update the projected demand based on the demand that has already been observed by that time. That is, if by a particular point in time demand of d_i has already been observed on itinerary i, then bid prices are adjusted using the conditional distribution of the demand, $\{\tilde{d}_i | \tilde{d}_i \geq d_i\}$. Of course, each time that the bid prices are updated, the capacity of each leg is adjusted to account for the fact that some capacity has already been utilized.

The stochastic program SLP was solved using the Benders decomposition of the two-stage model. Using the forecasts as described earlier in this section, the cutting plane coefficients, which formally involve an explicit expected value calculation, were approximated statistically based on 50 randomly generated (independent) observations drawn from the forecasted distributions. To avoid experimental bias, the random number generators used by the simulator to generate demands and by the optimization routine to generate the SLP bid-prices were initialized with different seeds. Moreover, to facilitate comparisons, once the bid-prices have been determined, revenues were derived from the same demand stream for both the SP and DLP bid-price models. By using the same demand stream for both pricing schemes, any differences in the revenues generated by the two models may be attributed to the differences in the bid-prices used.

16.4.2 Computational results

To assess the operational differences between the bid-prices identified by DLP and SLP, we undertook 30 independent replications of the simulation experiment described earlier. Thirty replications were performed to permit an observation of trends if they were to emerge. Not coincidentally, 30 replications can also be construed as operating the system for a month. There are numerous measures on which to compare the two pricing schemes. The actual demand is independent of the model used to determine the bid-price. Thus, on a run-by-run basis, both pricing schemes faced the same demand. As a result, we can compare the revenues generated by the two pricing schemes for each of the 30 demand scenarios generated. We begin with the results of two experiments, which differ in the size and structure of the underlying networks used. Data for the networks appear in the appendix.

Figure 16.1. *Network used in Experiment* 1.

Experiment 1: Small network

This experiment involves a small network with seven cities, seven legs, and eight routes. With two customer classes this yields a total of 16 itineraries. The cities and legs are depicted in Figure 16.1.

Of the 30 independent replications undertaken, SLP yielded higher revenues 60% of the time. It is instructive to review the relative values of the revenues (i.e., the revenues generated via DLP and SLP pricing schemes, relative to the maximum revenues generated). In Table 16.1, we summarize the relative revenue advantages of the two pricing schemes. Values reported in each column do not correspond to all 30 runs. Rather, they correspond only to those runs for which that method generated higher returns.

Table 16.1. *Summary of relative revenue advantage with a small network.*

	SLP Pricing	DLP Pricing
Runs w/ Revenue Advantage	60%	40%
Mean Revenue Advantage*	4.1%	3.2%
Max. Revenue Advantage*	10.9%	6.1%
Runs w/advantage $\geq$ 5%*	33%	25%
Runs w/advantage $\geq$ 9%*	11%	0%

*Conditioned on attaining the advantage.

For example, in the 60% of the runs in which SLP yielded higher revenues than DLP, the SLP revenues were an average of 4.1% higher than the DLP revenues. In the 40% of the runs in which DLP yielded higher revenues than SLP, the average DLP advantage was 3.2%. From Table 16.1, we see that for this small network, the SLP prices yield revenue advantages more often. In addition, the mean advantage is higher for SLP than for DLP and there were a significant number of outcomes for which dramatic improvements (i.e., $\geq$ 5%) were obtained over the DLP model.

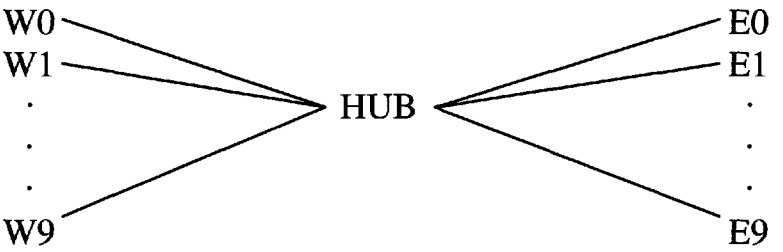


Figure 16.2. *Network used in Experiment* 2.

Experiment 2: Hub network

Since yield management is very widely applied in the airline industry, we also used a hub and spoke–type network in our computational tests. In such a network, there is usually one hub through which passengers on multiple routes are able to connect. It is not unusual for a commercial airline to have about 20 takeoffs and landings at a hub during any one hour. Hence, one may interpret this simulation as one that evaluates the quality of the model for one hour at a hub. Routes that use the hub during the hour are the ones that are competing for capacity on flights. Therefore it is realistic to study a hub-and-spoke network with approximately 20 destinations and one hub. The cities and legs associated with such a network are shown in Figure 16.2. In this experiment, we allow two customers classes on 50 routes, leading to 100 itineraries.

In Table 16.2, we summarize the relative revenue advantages of the SLP and DLP pricing schemes when applied to this network.

Table 16.2. *Summary of relative revenue advantage with a hub network.*

	SLP Pricing	DLP Pricing
Runs w/ Revenue Advantage	83.3%	16.7%
Mean Revenue Advantage*	3.2%	1.1%
Max. Revenue Advantage*	6.2%	2.5%
Runs w/advantage $\geq$ 5%*	17%	0

*Conditioned on attaining the advantage.

Table 16.3. *Fraction of arrivals, by class and time segment.*

Class	Time Segment [0, 60)	[60, 120)	[120, 150)	[150, 166)	[166, 180]
1	-	-	-	-	1.0
2	-	-	-	0.25	0.75
3	0.05	0.20	0.75	-	-
4	0.10	0.80	0.10	-	-

Table 16.4. *Summary of relative revenue advantage with the hub network example.*

	SLP Pricing	DLP Pricing
Runs w/ Revenue Advantage	63.3%	36.7%
Mean Revenue Advantage*	1.2%	0.5%
Max. Revenue Advantage*	3.4%	1.1%

*Conditioned on attaining the advantage.

Again, when the pricing schemes are applied to HUB, SLP yields a revenue advantage more often. With this larger network, the advantage occurs five times more often for SLP than for DLP. As before, we see that the mean advantage is higher for SLP than for DLP, as is the tendency toward dramatic improvements.

To determine the pricing schemes' performances under more extreme conditions, we repeated the experiment with the hub-and-spoke network. However, in this case, we included two additional customer classes on approximately half of the routes. This yields a total of 147 itineraries. As before, the higher class index is associated with lower fares. Table 16.3 indicates, by class, the fraction of customer arrivals that occur within various segments of the 180-day window.

Table 16.4 summarizes the relative revenue advantages of the SLP and DLP pricing schemes for this expanded hub-and-spoke network.

As with the other experiments, SLP prices yield a revenue considerably more often than the DLP prices. However, in this case the extent of the advantage tends to be somewhat dampened.

Table 16.5. *Summary of SLP and SLP′ relative revenue advantage (HUB4).*

	SLP Pricing	SLP′ Pricing
Runs w/ Revenue Advantage	53%	47%
Mean Revenue Advantage*	0.9%	0.6%
Max. Revenue Advantage*	2.3%	1.1%

*Conditioned on attaining the advantage.

16.5 Model extensions

Although SLP yields revenue advantages consistently more often than DLP, it appears that the introduction of additional fare classes may reduce the extent of the advantage. We conjecture that as the number of customer classes increases, the need to incorporate discrepancies between the fare classes within the model increases. There are two primary discrepancies that warrant investigation: nesting and timing. *Nesting* refers to the ability to sell capacity that has been allocated to a particular customer class to customers who are willing to pay a higher price. *Timing* refers to the fact that the demand arrival patterns can vary significantly by class; lower-fare customers will tend to arrive before the higher-fare customers. This SLP model does not account for either of these conditions, and it is possible that a model that is more faithful to these operating conditions will generate prices that yield higher revenues.

To approximately capture the impact of nesting, it is necessary to model the ability of higher-fare customers to access capacity that has been allocated to lower fares. In SLP, constraint (16.3) limits sales by customer class. Assuming that the customer classes are ordered such that $v_s \leq v_{s'}$ whenever $s \geq s'$, we can replace (16.3) with

$$\sum_{i \in I_\ell | s(i) \geq s} f_i \leq \sum_{s' \geq s} y_{s'\ell} \quad \forall s \in S, \ell \in L. \tag{16.5}$$

When solving the resulting stochastic program, this modification will simplify the first stage considerably. That is, all capacity will be allocated to the customer class with the highest index. Capacity can be completely reallocated for each demand scenario via the subproblem solution. In this sense, the resulting model may be overly optimistic.

Table 16.5 summarizes the relative revenue advantage of the two SLP models. Here, SLP refers to the original model (16.1)–(16.4), while SLP′ refers to the model in which (16.5) replaces (16.3). It is clear from Table 16.5 that there is no obvious or significant difference between SLP and SLP′.

The stochastic linear programs were solved using Benders' decomposition (i.e., the L-shaped method [19]) with 50 randomly generated observations. Recognizing that this sample will introduce error in the dual multipliers for (16.1) (i.e., the bid-prices), we also

investigated sample sizes of 100 and 200. As in Table 16.5, we found a slight preference for SLP over SLP'. This suggests that our linear model of the nested demands is not sufficient to adequately capture the complexity of the problem.

It is possible that additional revenue advantages may be obtained with a more faithful representation of the customer arrival processes, or timing. However, capturing the impact of timing requires a more extensive model adaptation. For example, one might consider the use of a multistage stochastic linear program instead of the simplified two-stage program that we have presented. Alternatively, one might consider function approximations of the impact of nesting and timing.

16.6 Conclusions

Several classes of models have been proposed for yield management. The basic DLP approach has remained the model of choice for several reasons: it is easily understood and implemented, and, more important, other models have not yet made the case that greater revenue generation may be possible. The SP model suggested in this paper is different. Our simulation study makes the case that SLP can provide substantial revenue advantages over the DLP model. Our investigation also opens the door to the likelihood that more complex SP models may lead to further revenue advantages. Hence, the SP approach has the potential to replace DLP as the model of choice.

Appendix

Information regarding the routes is detailed in Tables 16.7 and 16.9. For each fare class, we indicate the projected demand and revenue generated per ticket sold. For both the calculation of bid prices and the simulation program, demand for each of the route/class combinations were normally distributed with a mean equal to the projected demand and a coefficient of variation of 0.4. Tables 16.6 and 16.8 provide an indication of the extent to which demand exceeds capacity on each leg. The column labeled "Expected Demand" indicates the expected value of the demand associated with all of the itineraries using each leg.

Data for Experiment 1

Table 16.6. *Projected leg demand for Experiment* 1.

Leg	Capacity	Expected demand
A-B	300	365
B-A	300	475
C-A	150	115
A-D	150	175
G-B	100	175
B-F	300	295
E-B	100	120

Table 16.7. *Itinerary demand and revenue data for Experiment* 1.

Route	*Class* 1		*Class* 2	
	Expected demand	Revenue	Expected demand	Revenue
A-B	50	400	200	200
B-A	100	600	200	200
C-A-B	10	700	50	350
B-A-D	20	1200	100	400
C-A-B-F	5	800	50	400
E-B-F	20	800	100	400
G-B-F	20	700	100	350
G-B-A-D	5	1500	50	500

Data for Experiment 2

Table 16.8. *Projected leg demand for Experiment* 2.

Leg	Capacity	Expected demand
W0-HUB	300	365
W1-HUB	300	365
W2-HUB	300	365
W3-HUB	200	365
W4-HUB	200	365
W5-HUB	150	365
W6-HUB	150	365
W7-HUB	150	365
W8-HUB	100	365
W9-HUB	100	365
HUB-E0	300	475
HUB-E1	300	475
HUB-E2	300	475
HUB-E3	200	475
HUB-E4	200	475
HUB-E5	150	475
HUB-E6	150	475
HUB-E7	150	475
HUB-E8	100	475
HUB-E9	100	475

Table 16.9. *Itinerary demand and revenue for Experiment* 2.

Route	*Class* 1		*Class* 2	
	Expected demand	Revenue	Expected demand	Revenue
W0-HUB	15	700	60	250
W1-HUB	15	800	60	300
W2-HUB	15	1000	60	350
W3-HUB	10	500	40	200
W4-HUB	10	950	40	300
W5-HUB	10	650	30	225
W6-HUB	10	900	30	350
W7-HUB	10	700	30	225
W8-HUB	5	800	20	275
W9-HUB	5	750	20	250
HUB-E0	15	900	60	300
HUB-E1	15	750	60	300
HUB-E2	15	900	60	325
HUB-E3	10	800	40	300
HUB-E4	10	600	40	250
HUB-E5	10	600	30	200
HUB-E6	10	750	30	250
HUB-E7	10	600	30	200
HUB-E8	5	600	20	200
HUB-E9	5	550	20	175
W0-HUB-E5	30	1100	60	370
W0-HUB-E6	30	1250	60	325
W1-HUB-E2	30	1450	60	425
W1-HUB-E3	30	1350	60	400
W1-HUB-E6	30	1320	60	400
W2-HUB-E0	30	1600	60	475
W2-HUB-E3	30	1550	60	450
W2-HUB-E4	30	1350	60	425
W2-HUB-E7	30	1325	60	450
W3-HUB-E0	30	1200	60	375
W3-HUB-E7	20	935	40	300
W4-HUB-E4	20	1350	40	380
W4-HUB-E5	20	1350	40	410
W4-HUB-E8	20	1310	40	420
W4-HUB-E9	20	1275	40	400
W5-HUB-E1	30	1150	60	400
W5-HUB-E2	30	1310	60	400
W6-HUB-E0	30	1525	60	450
W6-HUB-E2	30	1525	60	425
W6-HUB-E3	20	1450	40	425
W6-HUB-E8	15	1275	30	450
W7-HUB-E1	30	1250	60	400
W7-HUB-E6	15	1250	30	375
W7-HUB-E9	15	1050	30	300
W8-HUB-E1	30	1020	60	400
W8-HUB-E3	20	1025	40	375
W8-HUB-E7	15	935	30	350
W9-HUB-E2	30	1000	60	400
W9-HUB-E4	20	935	40	325
W9-HUB-E8	10	875	20	325

Acknowledgment

This work was supported in part by NSF grant DMII-9414680.

Bibliography

[1] M. AVRIEL, *Nonlinear Programming*, Prentice–Hall, Englewood Cliffs, NJ, 1976.

[2] P. P. BELOBABA, *Airline seat management: An overview of seat inventory control*, Transport. Sci., 21 (1987), pp. 63–73.

[3] P. P. BELOBABA, *Application of a probabilistic decision model to airline seat inventory control*, Oper. Res., 37 (1989), pp. 183–197.

[4] J. F. BENDERS, *Partitioning procedures for solving mixed variables programming problems*, Numer. Math., 4 (1962), pp. 238–252.

[5] S. L. BRUMELLE AND J. I. MCGILL, *Airline seat allocation with multiple nested fare classes*, Oper. Res., 41 (1993), pp. 127–137.

[6] R. E. CURRY, *Optimal airline seat allocation with fare classes nested by origins and destinations*, Transport. Sci., 24 (1990), pp. 193–204.

[7] A. FERGUSON AND G. B. DANTZIG, *The allocation of aircraft to routes: An example of linear programming under uncertain demand*, Management Sci., 3 (1956), pp. 45–73.

[8] G. GALLEGO AND G. VAN RYZIN, *A multiproduct dynamic pricing problem and its applications to network yield management*, Oper. Res., 45 (1997), pp. 24–41.

[9] F. GLOVER, R. GLOVER, J. LORENZO, AND C. MCMILLAN, *The passenger-mix problem in the scheduled airlines*, Interfaces, 12 (1982), pp. 73–79.

[10] J. L. HIGLE AND S. SEN, *Stochastic Decomposition: A Statistical Method for Large Scale Stochastic Linear Programming*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[11] J. L. Higle and S. Sen, *Stochastic decomposition: An algorithm for two-stage linear programs with recourse*, Math. Oper. Res., 16 (1991), pp. 650–669.

[12] J. L. HIGLE AND S. SEN, *Statistical verification of optimality conditions*, Ann. Oper. Res., 30 (1991), pp. 215–240.

[13] S. E. KIMES, *Yield management: A tool for capacity-constrained service firms*, J. Oper. Management, 8 (1989), pp. 348–363.

[14] T. C. LEE AND M. HERSH, *A model for dynamic airline seat inventory control with multiple seat bookings*, Transport. Sci., 27 (1993), pp. 252–265.

[15] R. T. ROCKAFELLAR AND R. J.-B. WETS, *Scenarios and policy aggregation in optimization under uncertainty*, Math. Oper. Res., 16 (1991), pp.119–147.

[16] S. SEN, R. D. DOVERSPIKE, AND S. COSARES, *Network planning with random demand*, Telecommun. Syst., 3 (1994), pp. 11–30.

[17] B. C. SMITH, J. F. LEIMKUHLER, AND R. M. DARROW, *Yield management at American Airlines*, Interfaces, 22 (1992), pp. 8–31.

[18] K. TALLURI AND G. VAN RYZIN, *An analysis of bid-price controls for network revenue management*, Management. Sci., 44 (1996), pp. I577–I593.

[19] R. M. VAN SLYKE AND R. J.-B. WETS, *L-shaped linear programs with applications to optimal control and stochastic programming*, SIAM J. Appl. Math., 17 (1969), pp. 638–663.

[20] L. R. WEATHERFORD AND S. E. BODILY, *A taxonomy and research overview of perishable asset revenue management: Yield management, overbooking and pricing*, Oper. Res., 40 (1992), pp. 831–844.

[21] L. R. WEATHERFORD, S. E. BODILY, AND P. E. PFEIFER, *Modeling the customer arrival process and comparing decision rules in perishable asset revenue management situations*, Transport. Sci., 27 (1993), pp. 239–251.

[22] R. J.-B. WETS, *Stochastic programming: Solution techniques and approximation schemes*, in Mathematical Programming: The State of the Art, 1982, A. Bachem, M. Groetschel, and B. Korte, eds., Springer-Verlag, Berlin, 1982, pp. 566–603.

[23] R. J.-B. WETS, *Solving stochastic programs with simple recourse*, Stochastics, 10 (1983), pp. 219–242.

[24] R. J.-B. WETS, *Stochastic programming*, in Handbooks in Operations Research: Optimization, G. L. Nemhauser, A. H. G. Rinnoy Kan, and M. J. Todd, eds., North-Holland, Amsterdam, 1989.

[25] E. L. WILLIAMSON, *Airline Network Seat Control*, Ph.D. thesis, MIT, Cambridge, MA, 1992.

This page intentionally left blank

Chapter 17

Stochastic Optimization and Yacht Racing

*A. B. Philpott**

17.1 Introduction

This paper discusses the application of stochastic optimization methodologies to high-performance yacht racing. Yacht racing at the top level is a professional sport. Like Formula One car racing, yachting campaigns in the major international regattas like the America's Cup, the Admiral's Cup, or the Volvo Ocean Race require substantial financial backing to have any chance of success. A major part of this money is spent on design and performance analysis, since it is expensive to experiment by building more than one or two candidate designs. Indeed, in the America's Cup, the rules of the regatta preclude the competitors building more than two new boats, and so even with large budgets, most of the expenditure is directed towards performance analysis and design work.

Like many sporting contests, yacht racing carries with it a high degree of uncertainty. Since yachts exploit the wind for their speed, a major source of uncertainty lies in the behavior of the wind. The uncertainty of competitors' behavior can also have a dramatic effect on the outcome of a race or a regatta, but we shall not focus on that in this paper. Nearly every yachting campaign that can afford it hires a team of meteorologists and weather experts. Good predictions of prevailing weather conditions at the time of the regatta are important to allow the best design choices to be made. Predictions of wind speed and direction are also important to determine sailing strategies on the day of the race. For example, in the America's Cup, the latest prediction on the size and time of the next wind shift is passed to the onboard navigator minutes before the start. From then on, none of the crew is allowed to communicate with outside observers. In ocean races like the Volvo Ocean Race, wind forecasts are downloaded onto the yachts by satellite at regular intervals

*Department of Engineering Science, University of Auckland, Auckland, New Zealand (a.philpott@auckland.ac.nz).

and used by onboard navigators to make good course decisions.

Although advanced computational tools such as computational fluid dynamics (CFD) have been used in high-performance yachting for some time, there has been some reluctance by yacht designers to adopt the methods of mathematical programming. In this respect the seasoned eye of the designer is often thought to be a better judge of the difficult trade-offs to be made than any optimization software. Furthermore, when compared with full-scale observations, the accuracy of CFD codes is often questionable in yachting applications. Nevertheless, these codes can give valuable insight into the relative performance of different designs, and they have begun to be used in commercial applications (see, e.g., [3]). As observed in [3], "the design evaluation becomes a challenging task in itself since a (probabilistic) measure of merit ought to be considered." This is one area in which the models and techniques of stochastic programming can make a contribution.

The paper proceeds as follows. Since they are a fundamental ingredient in any optimization exercise, we first give a brief account of models for the performance analysis of sailing boats. These use models of the forces on the sails and hull of a yacht to predict its speed and dynamic behavior. In section 17.3 we give an account of race-modeling programs, and in section 17.4 we discuss optimizing routing decisions under uncertainty. Different models are obtained depending on the time scales involved: for short races the wind is best modeled by a stochastic process, and tacking (changing direction through the wind direction) must be accounted for; in long races the weather is typically modeled using scenarios to account for the substantial correlation observed between wind fields at consecutive time intervals. Finally, in section 17.5 we discuss optimal yacht design under uncertainty. Here we focus on designing for America's Cup campaigns, and show how uncertainty has an important effect on the design process.

17.2 Performance modeling

We give a brief overview of the yacht modeling that must be undertaken to carry out an optimization exercise. The key tool in this regard is a computer model of a yacht called a *velocity prediction program* (VPP). VPPs were devised more than 20 years ago by yacht designers and handicappers. The seminal papers in this area are by Letcher [8] and Kerwin [6], who formulate the equilibrium behavior of a yacht as a set of simultaneous nonlinear equations equating the forces and moments on the vessel. Since one of the variables in these equations is the velocity of the yacht, a solution to these equations will give a prediction of the velocity.

A number of authors have proposed improvements (see, e.g., [16, 12, 14]) to the basic model, and much effort has been devoted to improving the force models that provide the appropriate equations in the VPP. In particular, a VPP requires models for both the hydrodynamic forces on the hull and the aerodynamic forces on the sails. The hydrodynamic model uses information from CFD analysis and towing-tank data to fit equations that define the forces and moments on a hull moving through the water at a given velocity and trim. Similarly, the aerodynamic model uses CFD, scale model measurements in wind tunnels, and full-scale measurements to derive models for the lift and drag on the sails and rig as a function of apparent wind speed, wind angle, and sail trim. All of these models depend on some geometrical description of the sails and rig, and the hull and its appendages, namely,

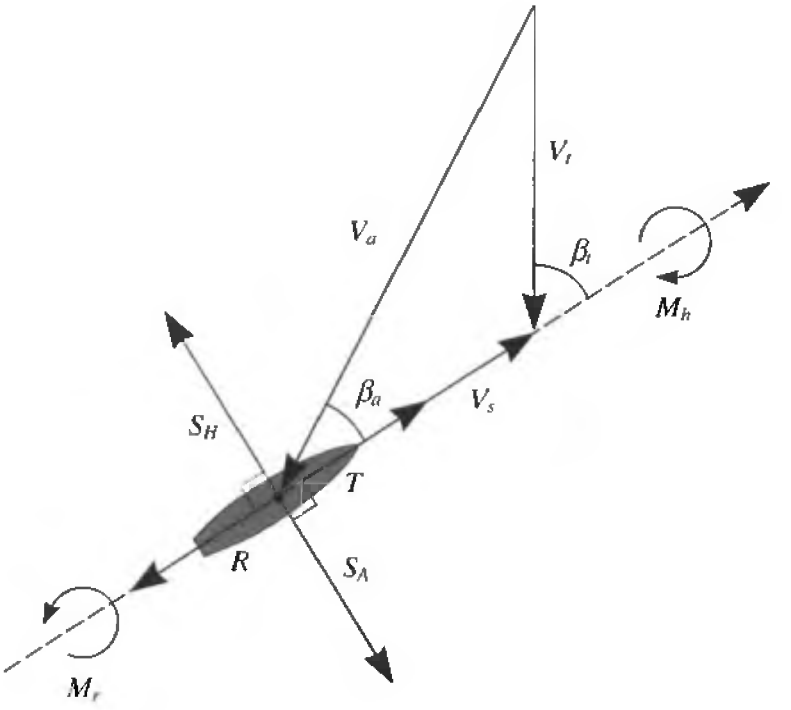


Figure 17.1. *Forces usually modeled in a VPP.*

the rudder, the keel, and (possibly) a bulb and winglets.

The forces modeled in a standard VPP are shown in Figure 17.1. Here S_A is the aerodynamic side force generated by the wind pressure on the sails, and S_H is the hydrodynamic side force generated by the hull and appendages.

The apparent wind velocity V_a that a yacht sees as it sails is not the same as the true wind velocity V_t. As shown in Figure 17.1, the apparent wind speed V_a and the apparent wind angle β_a can be calculated as follows:

$$V_a = \sqrt{(V_s + V_t \cos \beta_t)^2 + (V_t \sin \beta_t)^2},$$

$$\beta_a = \arctan\left(\frac{V_t \sin \beta_t}{V_s + V_t \cos \beta_t}\right).$$

When calculating the aerodynamic forces on the sails, the sail force coefficients in VPP models are modified by a set of standard sail trim variables denoted here by r. These alter the shape of the sail to affect its aerodynamic lift coefficient C_L and its aerodynamic drag coefficient C_D, which are also functions of β_a. The exact form of these functions is found by a combination of experimentation and CFD.

The aerodynamic lift and drag forces are calculated using the density of air ρ, the sail reference area A_s, and the aerodynamic lift and drag coefficients so that

$$L = \frac{1}{2}\rho V_a^2 A_s C_L(\beta_a, r)$$

and

$$D = \frac{1}{2}\rho V_a^2 A_s C_D(\beta_a, r).$$

The aerodynamic lift and drag forces in the plane of the sail are then resolved into the direction of motion of the yacht hull to obtain an aerodynamic thrust

$$T = L \sin \beta_a - D \cos \beta_a$$

and an aerodynamic side force

$$S_A = L \cos \beta_a + D \sin \beta_a.$$

Given values for the yacht velocity, the rudder angle, and the leeway angle (the slight difference between the angle the yacht is heading and the angle of its track though the water), it is possible to calculate hydrodynamic forces on the hull and appendages. The hull and appendages create a resistance force R due to the frictional forces of the water moving over the hull surfaces, as well as resistance from the bow and stern waves made by the hull. These forces are computed either using CFD models or by fitting theoretical formulas to towing tank test data (see, e.g., [12]). At nonzero leeway angles the hull and appendages create a side force S_H (stopping the yacht from slipping downwind) that can also be calculated from hydrodynamic formulas calibrated to experimental data.

This process gives a set of two equilibrium equations,

$$F = R, \qquad S_A = S_H.$$

A further equilibrium equation can be derived by requiring the overturning (or heeling) moment that comes from the sail forces to be balanced by a righting moment produced by the keel and the hydrostatic and hydrodynamic forces on a heeled yacht hull. These three equations (which are solved to yield the yacht velocity) form the most basic form of a VPP.

Figure 17.2 shows a polar representation of the output from a typical VPP. Each curve corresponds to a given true wind speed, where the radius of the curve gives the steady-state velocity of a yacht sailed over a range of angles of attack β_t to the true wind. (The polar plot must be combined with its mirror image to obtain velocities at negative angles of attack.) The speed drops to zero as the boat heads closer to the wind. Figure 17.2 also depicts (by shaded circles) the points of sail giving maximum *velocity made good* (VMG) sailing into the wind (between 40 and 50 degrees for this boat) and sailing downwind (between 130 and 150 degrees). The VMG is found by projecting the velocity vector onto the wind direction to give $V_s \cos \beta_t$ (shown for the upwind case in the figure as giving a maximum VMG of just over 7 knots for the highest wind speed). Maximizing VMG gives the optimum heading to sail when wishing to travel directly upwind (or downwind) in a constant wind.

Velocity prediction programs compute a steady-state velocity of the yacht. By allowing $T - R$ and $S_A - S_H$ to be nonzero, it is possible to derive a system of ordinary differential equations describing the dynamic motion of the yacht, giving

$$\dot{s} = f(s, u, \omega),$$

where s is a vector of state variables defining among other things the velocity, heading, and heel angle of the yacht; u is a vector of yacht controls including rudder angle, and sail trim;

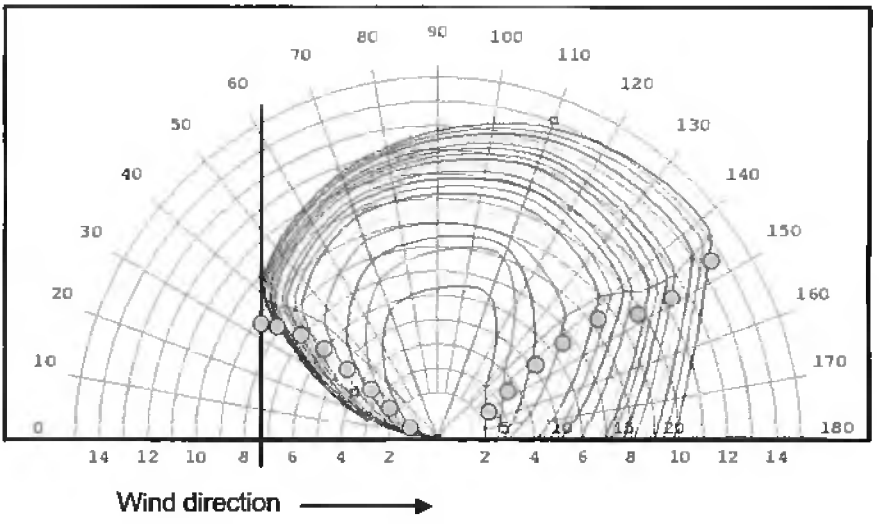


Figure 17.2. *Polar plot obtained from a VPP.*

and ω is a set of exogenous parameters defining the wind and sea state. Dynamical systems based on these principles have been developed for simulating tacking (see, e.g., [11]), as well as modeling the effects of design parameters on dynamic behavior (see [2]), although the calibration of these models to represent real behavior remains a challenging obstacle to their widespread adoption.

17.3 Race modeling programs

A design evaluation tool that is closely related to the VPP is the *race modeling program* (RMP). The output from a VPP along with historical weather data is used in an RMP to assess different yacht designs, by racing candidate designs against each other over a range of weather scenarios. A win/loss probability is then estimated for each pair of yachts.

Although high-performance yacht designers now routinely simulate the performance of candidate designs for ocean races, the use of RMPs in short course races has been confined primarily to the America's Cup. The use of RMPs was first reported by the 1987 Stars & Stripes syndicate in a famous *Scientific American* article [9]. The Stars & Stripes design team realized early in the campaign that

> yacht racing has an essentially random component in that the relative performance of two yachts depends on the wind speed and the sea conditions, which vary randomly from day to day. VPP results by themselves are therefore inconclusive and possibly misleading for determining the order of merit of two candidate yachts over a series of races [9].

For the Stars & Stripes campaign two RMPs were developed. The first was a simple probabilistic model using the predicted time difference between the two yachts as a function of wind speed, and a distribution for the wind speed to determine the winning probability.

The second RMP improved on the first, allowing for wind speed distributions for each leg, also taking into account interaction effects when the yachts are very close. The output from each RMP was used to analyze the win/loss probabilities of two yachts over a specified course.

One of the fundamental realizations during the Stars & Stripes campaign in 1987 was that it is not possible to determine the best possible yacht design without knowing anything about the characteristics of the competing yacht [9]. One of the key factors in the success of Stars & Stripes was identifying that the eventual competitors were much shorter than the initial design for Stars & Stripes. Using the RMPs that had been developed, combined with game theory, it was found that although the chance of success for Stars & Stripes would be very high in the later rounds of the Cup with heavier winds, the probability of success during the preliminary rounds would be small due to the lighter winds. Hence, the original design was shortened so that it would be competitive in the preliminary elimination rounds of the Challenger series but still have a high chance of success in the later rounds. This was undoubtedly one of the most important decisions made during the campaign and was instrumental in Stars & Stripes being chosen to challenge Kookaburra for the Cup.

Since 1987, RMPs have featured in several different America's Cup challenges. For example, one of the major programs for the Partnership for America's Cup Technology (PACT), which was founded in 1990, was to gather site-specific environmental data in San Diego—the location for the next America's Cup in 1992. These data were used in the creation of "a statistical weather model... for use in conjunction with the RMP which was developed for evaluating the probability of success for various designs in conditions likely to be experienced off San Diego during the trials and the America's Cup" [4]. PACT researchers also deployed a wave-measuring buoy to gather sea state spectra, which could then be correlated with local meteorological conditions so that the RMP could be run with rough water effects.

The approach to race modeling pioneered by the Stars & Stripes design team estimates times around a course in different wind and wave conditions, sampled for each leg of an America's Cup course. The leading boat in a match race has an enormous advantage owing to its ability to "cover" its opponent by sailing a course that keeps between the trailing boat and the next mark. When sailing upwind this advantage is enhanced by the ability to spill turbulent "dirty" air onto the trailing boat. Race modeling programs of the type described above can account for this advantage to some extent by conditioning the probability of a yacht being in front at the end of each leg on whether it is in front at the beginning, but these effects are difficult to quantify even with a large amount of experience.

An alternative approach developed by Philpott, Henderson, and Teirney [13] in collaboration with Team New Zealand uses a fixed time increment simulation model of each leg that accounts for wind fluctuations and interactions between the two boats. Their model, called ACROBAT, uses a dynamical system for each yacht that is based on output from a velocity prediction program.

The ACROBAT model, developed specifically for the 2000 America's Cup in Auckland, requires a stochastic process for wind speed and direction that can be used to randomly generate wind realizations. A plot of an hour of wind speed and wind direction data as measured on a buoy located on the Hauraki Gulf race course is shown in Figure 17.3.

It is convenient to assume that the wind speed and its direction are independent; correlation coefficients estimated from observed data are consistent with this assumption.

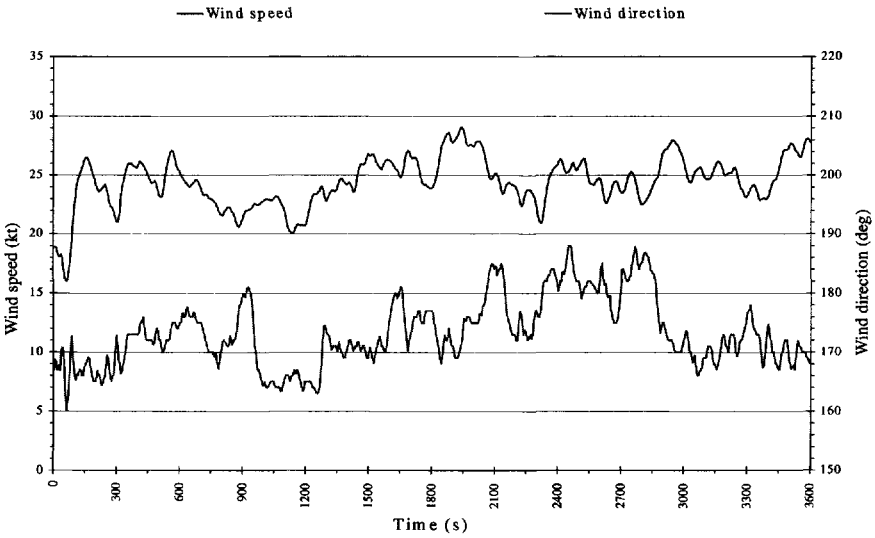


Figure 17.3. *Graph of stationary wind speed and direction.*

In ACROBAT the wind speed at a fixed location is modeled to a reasonable accuracy by an ARMA process that reverts to a mean wind speed for the race. The wind direction observations, however, do not fit an ARMA process well since they show large shifts in the wind direction at random intervals. A simple model that provides a realistic fit to the data as seen by a stationary observer is a hidden Markov chain model, in which the wind direction is an ARMA process that reverts to a mean which is fluctuating according to a Markov chain. The states of this Markov chain are a discrete set of wind angles, being the midpoints of intervals partitioning the range of possible wind angles.

A race-modeling simulation requires samples of the wind speed and wind direction for both boats, which are often at different locations. Let V_1^k and V_2^k be the true wind speeds and β_1^k and β_2^k the true wind directions observed by yacht 1 and yacht 2, respectively, at time step k. The rate at which the wind fluctuates on the boat will depend on the velocity of the boat—when it is sailing upwind, the wind appears to change more quickly than when traveling downwind—so the wind processes must account for this. It is also helpful to assume that the wind over the course obeys the Taylor hypothesis of wind engineering (see, e.g., [7]). The Taylor hypothesis treats the wind turbulence, although random, as fixed in the sense that the eddies in the wind field travel down the race course at a given mean wind speed V, so an anenometer at the top mark a distance d upwind of the bottom mark will give exactly the same reading as an anenometer at the bottom mark (d/V) seconds later. This entails that the wind observation (V_1^k, β_1^k) on the leading yacht should be strongly correlated with $(V_1^{k+l}, \beta_1^{k+l})$, the observation on the trailing yacht l time steps later, given that they are sailing through the same wind eddies. Here l will depend on V and the VMG of the trailing yacht. In other circumstances the yachts sail through different wind fields. When the yachts are level pegging but widely separated across the course, (V_1^k, β_1^k) and (V_2^k, β_2^k) will be very weakly correlated. In ACROBAT we attempt to model this variation in correlation with

separation as follows.

For the case where the yachts are very far apart, the wind speeds can be generated independently. Hence, a separate ARMA process is used for each yacht. In particular,

$$\begin{pmatrix} V_1^k \\ V_2^k \end{pmatrix}_I = \begin{pmatrix} \alpha \\ \alpha \end{pmatrix} + \begin{pmatrix} \phi_1 & 0 \\ 0 & \phi_1 \end{pmatrix} \begin{pmatrix} V_1^{k-1} \\ V_2^{k-1} \end{pmatrix} + \begin{pmatrix} \phi_2 & 0 \\ 0 & \phi_2 \end{pmatrix} \begin{pmatrix} V_1^{k-2} \\ V_2^{k-2} \end{pmatrix} + \begin{pmatrix} \theta_1 & 0 \\ 0 & \theta_1 \end{pmatrix} \begin{pmatrix} \varepsilon_1^{k-1} \\ \varepsilon_2^{k-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_1^k \\ \varepsilon_2^k \end{pmatrix},$$

where

$$\begin{pmatrix} \varepsilon_1^k \\ \varepsilon_2^k \end{pmatrix} \sim N(0, \Lambda_I) \quad \text{and} \quad \Lambda_I = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}.$$

For the case where the yachts are very close, we use a vector process:

$$\begin{pmatrix} V_1^k \\ V_2^k \end{pmatrix}_D = \begin{pmatrix} \alpha \\ \alpha \end{pmatrix} + \begin{pmatrix} \frac{\phi_1}{2} & \frac{\phi_1}{2} \\ \frac{\phi_1}{2} & \frac{\phi_1}{2} \end{pmatrix} \begin{pmatrix} V_1^{k-1} \\ V_2^{k-1} \end{pmatrix} + \begin{pmatrix} \frac{\phi_2}{2} & \frac{\phi_2}{2} \\ \frac{\phi_2}{2} & \frac{\phi_2}{2} \end{pmatrix} \begin{pmatrix} V_1^{k-2} \\ V_2^{k-2} \end{pmatrix} + \begin{pmatrix} \frac{\theta_1}{2} & \frac{\theta_1}{2} \\ \frac{\theta_1}{2} & \frac{\theta_1}{2} \end{pmatrix} \begin{pmatrix} \varepsilon_1^{k-1} \\ \varepsilon_2^{k-1} \end{pmatrix} + \begin{pmatrix} \varepsilon_1^k \\ \varepsilon_2^k \end{pmatrix},$$

where $\begin{pmatrix} \varepsilon_1^k \\ \varepsilon_2^k \end{pmatrix} \sim N(0, \Lambda_D)$ and $\Lambda_D = \begin{pmatrix} \sigma^2 & \sigma^2 \\ \sigma^2 & \sigma^2 \end{pmatrix}$. The wind speed observed by each yacht is then modeled as a convex combination of these processes:

$$\begin{pmatrix} V_1^k \\ V_2^k \end{pmatrix} = (1-\lambda) \begin{pmatrix} V_1^k \\ V_2^k \end{pmatrix}_I + \lambda \begin{pmatrix} V_1^k \\ V_2^k \end{pmatrix}_D, \tag{17.1}$$

where $\lambda \in (0, 1)$ depends on the separation of the yachts.

The hidden Markov chain model defining the wind direction for a single boat requires a similar modification for two boats. When two boats are modeled, the states of the Markov chain consist of ordered pairs (i, j), where i is the state of the first boat's wind direction and j is the state of the second boat's wind direction. Therefore, the one-step transition matrix for the two-boat Markov chain is of the form $P((i, j), (k, l))$, where i, j, k, and l vary over the possible wind direction states.

The transition matrix varies depending on the separation of the boats. When the boats are far apart, the wind direction for each boat evolves independently according to the transition matrix P_S for the single-boat model. The transition matrix for this case is

$$P_I((i, j), (k, l)) = P_S(i, k) P_S(j, l).$$

When the boats are close together, the wind directions for the two boats coincide and evolve according to P_S. The transition matrix (P_D) in this case is

$$P_D((i, i), (k, l)) = \begin{cases} P_S(i, k) & \text{if } k = l, \\ 0 & \text{if } k \neq l. \end{cases} \tag{17.2}$$

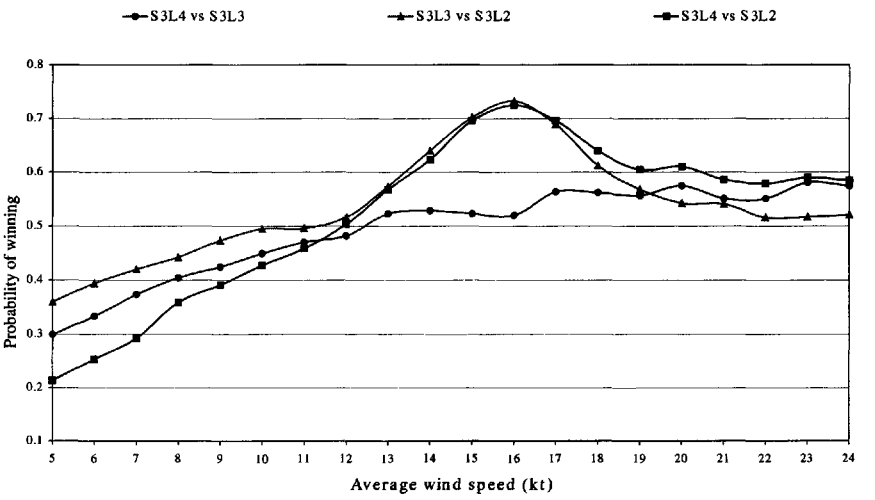


Figure 17.4. *Probability of winning with length variation.*

When $i \neq j$, the transition probabilities $P_D((i, j), (k, l))$ are chosen to force the wind directions together by requiring that the state transition to

$$\begin{pmatrix} k \\ l \end{pmatrix} = \begin{cases} \begin{pmatrix} i+1 \\ j-1 \end{pmatrix} & \text{if } |i-j| > 1, \\ \begin{pmatrix} i \\ j-1 \end{pmatrix} & \text{if } |i-j| = 1, \end{cases}$$

assuming that $i < j$ (the case $i > j$ is similar). Once the wind directions coincide, we then use (17.2) for the dependent transition probabilities. Philpott, Henderson, and Teirney [13] discuss an alternative approach based on Vasershtein coupling (see [10]), but this is shown to produce unrealistic weather realizations when compared with observed data. In fact, ACROBAT uses not P_I or P_D but a convex combination

$$P = (1 - \lambda)P_I + \lambda P_D,$$

where the choice of $\lambda \in (0, 1)$ depends on the separation of the yachts. The resulting Markov chain gives a mean wind direction at each time step for each boat. These are then supplied to a bivariate ARMA process for β_1 and β_2 of the same form as (17.1).

The aim of ACROBAT was to provide an estimate of the probability that a candidate design would beat a given competitor in a given average wind speed. In the 1995 America's Cup in San Diego the wind speed had a low mean and low variance. The Hauraki Gulf, which was the venue for the 2000 America's Cup, experiences higher average wind speeds that are also more variable. Designers at Team New Zealand were interested in the trade-offs to be made in designing a heavy-weather boat and a light-air boat. These are illustrated in Figure 17.4, which shows estimates of win/loss probabilities for three candidate designs with different lengths, obtained by simulating 10,000 races between each pair of yachts in each average wind speed. (This gives an asymptotic 95% confidence interval on the

probability estimate of ± 0.01.). Here S3L2 is shorter than S3L3, which is shorter than S3L4.

Figure 17.4 shows that the longest boat performs well in stronger winds but is beaten by a short boat in lighter airs. One reason for this is that America's Cup design rules penalize long boats by requiring them to have a smaller sail area. Moreover, hydrodynamic drag is a decreasing function of waterline length for high boat speeds but is less sensitive to waterline length when speeds are small (and frictional drag dominates). We discuss the optimization of these trade-offs in section 17.5. In the next section, we discuss the optimal choice of route to be made in uncertain weather.

17.4 Optimal routing under uncertainty

Monte Carlo simulation programs like ACROBAT have many advantages over traditional race modeling programs, since the result of a single run depends not only on the sampled weather but also on how the yachts are sailed and how they interact. In a Monte Carlo simulation model, it is possible to alter the strategy adopted by the helmsman and tactictian on each boat to investigate the performance of different strategies and to enable recommendations to be made to the tacticians about how to sail yachts of a given design. The details of how this is modeled in ACROBAT can be found in [15]. In this section we focus on yacht routing strategies. From this perspective, yacht races are effectively dynamic stochastic games, but our treatment here will ignore the game theory aspects and seek solutions that are optimal responses to a fixed strategy adopted by a competitor or fleet of competitors. We then seek solutions to stochastic dynamic programs. We shall divide the discussion into models for route optimization over short courses and route optimization for ocean races.

17.4.1 Short courses

Short course yacht races involve sailing around a course marked with buoys. Here the wind usually comes from the same quadrant over the duration of the race, with small fluctuations in speed and direction. In this section we consider the problem of sailing between two marks of a short course race in the minimum time in a varying wind and tidal current. It is relatively easy to get very accurate forecasts for currents, and so we assume that deterministic current information is available. By combining wind and current information, we can determine using a VPP the velocity of the yacht over the sea floor at any point and time for any chosen heading.

We restrict attention without loss of generality to a single upwind leg. The principles for downwind and reaching legs are the same albeit with different data. The first step in describing a dynamic programming recursion involves a discretization of the upwind leg into a rectangular grid of possible yacht locations with increments across the course and increments in the direction of the course. The rectangular discretization defines a finite set of locations indexed by (i, n), $i \in I$, $n = 0, 1, \ldots, N$, with coordinates with the property that any route will visit the locations in order of decreasing n, so we can treat n as a stage of the dynamic program, while i, the index of the coordinate measured across the course, becomes a state variable. This approach depends on the assumption that the route we follow will travel approximately in the direction of the course in every time step. We assume that

(0, 0) represents the location of the mark at the end of the leg.

In short course routing, we assume that there is little systematic spatial variation in the true wind speed V and true wind direction β, but we model the wind using the hidden Markov chain process described in section 17.2. For computational purposes we discretize the state space, which results in two discrete Markov chains for V and β, respectively. The model then gives a transition in wind speed and direction after each movement between stages. One can think of this process as defining a set of parallel lines of wind vectors across the course. In each stage the wind vectors across the course are identical, but they make a random transition before we move from locations on one line to locations on the successor line.

In short course racing, time is lost when a boat is tacked, going either upwind or downwind. This loss in time due to tacking is denoted by τ. We assume that the vessel on arriving at location (i, n) under wind realization (V_n, β_n) observes a random transition to (V_{n-1}, β_{n-1}) and then (possibly after tacking) sails an optimal course to the next location $(j, n-1)$. We can choose the optimum tacking decision at each stage by adding a state k that is set to 1 if we are on starboard tack (i.e., wind coming from the right-hand side) and 0 if we are on port tack. Now for any stage n let $c_n(i, j, k, V, \beta)$ be the time taken to sail from location (i, n) to location $(j, n-1)$ on tack k, given a true wind speed of V and a true wind direction of β. This gives the following recursion to minimize the expected time to reach the next mark:

$$f_0(i, k, V_n, \beta_n) = \begin{cases} 0, & i = 0, \\ \infty, & i \neq 0, \end{cases}$$

$$f_n(i, k, V_n, \beta_n) = \min\{S_n(i, k, V_n, \beta_n), \tau + S_n(i, 1-k, V_n, \beta_n)\},$$

where

$$S_n(i, k, V_n, \beta_n) = \min_{j \in \Gamma(i,n,k)} \{c(i, j, k, V_n, \beta_n) + E_{V,\theta}[f_{n-1}(j, k, V, \beta) \mid V_n, \beta_n]\}.$$

Here $\Gamma(i, n, k)$ is the set of j such that location $(j, n-1)$ can be reached from (i, n) by sailing on tack k and $f_n(i, k, V_n, \beta_n)$ is the expected time to sail from location (i, n) to the finish, assuming that the boat arrives at this location under wind realization (V_n, β_n) and on tack k. The conditional expectation is taken over all possible transitions of wind state. Thus the route taken by the boat optimizes with perfect knowledge of the time taken to reach the next stage but adds to this an expectation of the time to go from there to the destination.

There are some obvious weaknesses in the above model. The model assumes that the transitions in wind will occur at regular intervals in time, but these will not necessarily coincide with the time that the yacht reaches the locations corresponding to stage $n-1$. For example, if $c(i, j, k, V_n, \beta_n)$ is longer than than the time interval between transitions, then (V_n, β_n) will represent the wind state only over the first part of the yacht's track to $(j, n-1)$. A similar problem arises if τ is longer than than the time interval between transitions. A possible remedy is to replace the rectangular grid by a different discretization to ensure that transitions do not occur en route between locations, but as the geometry of the discretization will depend on the wind realization, it is not clear how to do this effectively. An alternative approach allows wind transitions to occur en route to $(j, n-1)$. The evolution of the wind

over the time taken to reach $(j, n-1)$ then takes the form of a branching scenario tree. The current step of the dynamic programming recursion then minimizes the expectation of $c(i, j, k, V_n, \beta_n) + f_{n-1}(j, k, V, \beta)$ over these scenarios.

The choice of objective function in these models, being risk-neutral, is also open to some debate. In yacht racing, the crew's attitude to risk alters throughout the regatta, and indeed throughout each leg of a race. Crews leading a race tend to be risk-averse by moderating their optimal route to the destination so as to cover the opposition. If they happen to be coming at the back of the fleet, the crew will tend to seek risk, in the hope that events might turn in their favor. In all circumstances the efforts are directed at maximizing the probability of winning or placing highly in a race as compared with minimizing the arrival time.

Risk-averse and risk-seeking attitudes are typically modeled using utility functions. In the context of minimizing the expectation of the arrival time T at the destination, a risk-averse decision maker would minimize $E[u(T)]$, where $u(.)$ is a convex (disutility) function, and a risk-seeking decision maker would minimize $E[v(T)]$, where $v(.)$ is a concave function. The former function $u(.)$ accentuates large T values and so discourages them disproportionately, whereas a concave function $v(.)$ discounts large values of T and so weights them by less than their proportion of an expected time.

Nonlinear utility functions can be modeled in a dynamic program by adding another state variable (see, e.g., [17]). In our context this is t, the time of arrival at a location. Let $U_n(i, k, V, \beta, t)$ be the expected disutility of sailing an optimal course from location (i, n) if arriving at time t on tack k under weather state (V, β). The alteration to the recursion is as follows:

$$U_0(i, k, V, \beta, t) = \begin{cases} u(t), & i = 0, \\ \infty, & i \neq 0, \end{cases}$$

$$U_n(i, k, V_n, \beta_n, t) = E_{V,\theta}[\min\{S_n(i, k, V, \beta, t), S_n(i, 1-k, V, \beta, t+\tau)\} \mid V_n, \beta_n],$$

where

$$S_n(i, k, V_n, \beta_n, t) = \min_{j \in \Gamma(i,n,k)} \{U_{n-1}(j, k, V_n, \beta_n, t) + c(i, j, k, V_n, \beta_n)\}.$$

With appropriate definitions of u the same recursion can be used to maximize the probability of arriving before a given time or to maximize the probability of arriving before another yacht that has a known arrival time distribution.

As an example of the solutions delivered by these routing methods, Figure 17.5 shows the optimal tacking policy for an America's Cup yacht when sailing in a constant wind speed of 16 knots but a wind direction stochastically varing around 0 degrees. In this experiment a tack costs 4 seconds. Figure 17.5 shows, for each location, the direction the yacht should head if it is currently on starboard tack (wind from the right) and observing a wind direction of 0 degrees. If the wind direction were constant at 0 degrees, then the optimal policy would be to sail two close reaches each at the angle that maximizes VMG upwind. A single tack then occurs on the left layline, which is an imaginary line through the mark at the optimum VMG angle to the direction of the course (shown in bold in the figure). With uncertainty in the fluctuating wind direction the optimal strategy tacks onto port, back toward the center of the course before it meets the laylines, unless it is close to the mark, in which case the boat sails beyond the layline to avoid the possible need to perform extra tacks.

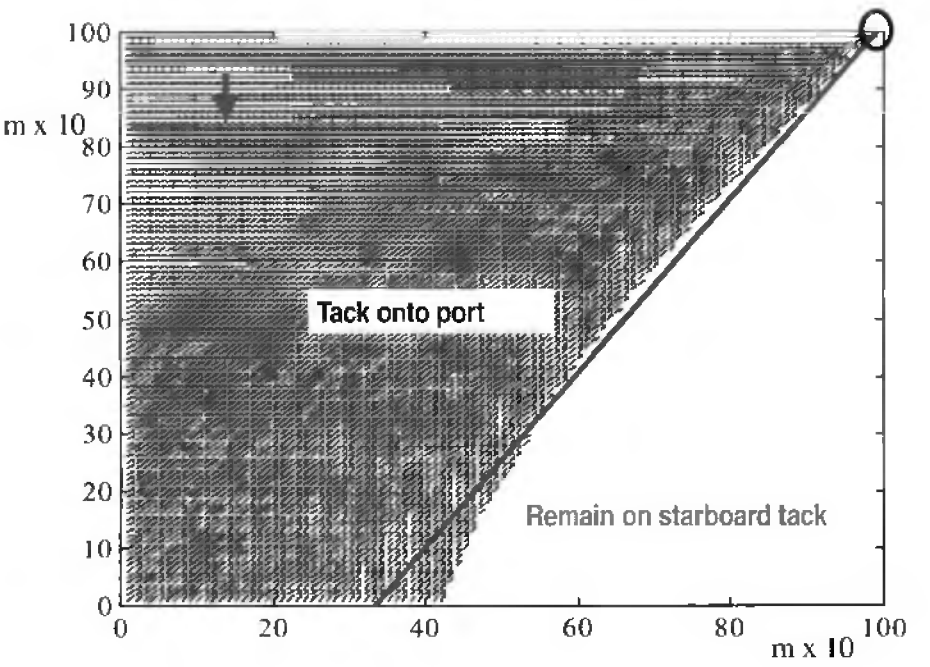


Figure 17.5. *Optimal strategy approaching the top mark.*

17.4.2 Ocean races

In this section we describe a stochastic routing model for ocean races. Here time is measured in days rather than minutes, and routes are planned over many miles of ocean. For this reason the loss in time from tacking is negligible compared with the duration of a leg. If tacking incurs no penalty, then the vessel can travel into the wind (and straight downwind) at its optimum VMG for the wind speed. This is easily modeled by replacing the polar diagram (including both positive and negative angles) by its convex hull.

In long course routing some changes are necessary in the modeling of wind. In section 17.2, we assumed that the wind speed and direction follow a stochastic process with known parameters. In ocean racing the short time scale of such a process is less important than the major weather effects that influence strategic decisions. In these circumstances navigators rely on weather forecasts that are transmitted to the yachts, usually by satellite. A typical weather forecast gives a set of discrete wind fields at evenly spaced time intervals. Each wind field defines a set of wind vectors over a regular grid of points. Even if the forecast is correct, it will usually be provided only for some specified locations and for some specified times. To optimize the route, the wind speed and direction must be known over the whole course and for every time that we might arrive at a given location. Therefore some degree of interpolation in space and time is necessary to produce a workable model. While such interpolation models for wind fields should respect principles of meteorology, they must be able to be computed efficiently to enable a solution to be obtained in a reasonable amount of time, and so most methods use linear interpolation.

For a single forecast, the interpolated wind fields define a dynamically evolving vector field that can be used to determine an optimal deterministic route. To accommodate uncertainty in the weather, we need a model that describes how this uncertainty behaves. The mean reverting equations used in the short course model do not fit particularly well to large-scale weather patterns, in which the wind direction follows the meteorology of depressions and anticyclones rather than reverting to some mean direction. An even simpler approach is to treat the wind speed and direction at a point as a random variable with a known distribution. This climatology model is useful for determining routes in the absence of any weather information, but it will overestimate the expected sailing times by assuming that the wind is not serially correlated, because future weather patterns are less able to be anticipated in route planning.

An alternative approach to modeling the uncertainty is to use weather scenarios, which are self-contained forecasts that unfold according to an underlying scenario tree $\mathcal{N}$. Each node n in the scenario tree corresponds to a unique history of evolving wind fields. When a wind history takes alternative paths into the future, the node n has several successors, which we denote by $\{n^+\}$. Each node n in the scenario tree (except for the root node) has a unique predecessor n^-. The transition probability $p(n)$ is the conditional probability of moving from node n^- to node n, and $t(n)$ is the time at which this transition occurs.

Suppose the boat is at location x at time t in scenario node n, where x is a point in some discretized grid. Being in scenario node n means that $t(n^-) < t \leq t(n)$. We seek a recursion for $f(x, t, n)$ the optimal expected time to sail from x to the destination. Let $c(x, y, t, n)$ be the time it takes to sail from location x to location y under the wind field dynamics corresponding to node n. Define

$$g(x, y, t, n) = \begin{cases} c(x, y, t, n) + f(y, t + c(x, y, t, n)), & c(x, y, t, n) \leq t(n) - t, \\ \sum_{n^+ \in \{n^+\}} p(n^+)[c(x, y, t, n^+) + f(y, t + c(x, y, t, n^+))] \text{ otherwise.} \end{cases}$$

Since scenario changes occur much less frequently than routing decisions, we assume that $c(x, y, t, n) \leq t(n^+) - t$ for any $n^+ \in \{n^+\}$. Then

$$f(x, t, n) = \begin{cases} 0, & x = \text{ the destination,} \\ \min_{y \in \Gamma(x)} g(x, y, t, n) & \text{otherwise.} \end{cases}$$

Here $\Gamma(x)$ is the set of locations that can be reached from location x in one step. Since t is recorded as a state variable, utility functions or other nonseparable functions can be incorporated along the same lines as discussed in the previous section.

The recursion above has been coded into a stochastic yacht routing application developed by Allsopp [1]. Figure 17.6 shows a screen shot from this program applied to a small example. Here there are three weather scenarios that become known after a day's sailing in light winds, and we seek to minimize the expected arrival time at the destination. In scenario 1 the wind strengthens in the north, in scenario 2 it strengthens in the south, and in the third scenario the wind remains light throughout the region. The scenarios occur with probabilities 0.4, 0.3, and 0.3. The optimal solution as shown is to head slightly north and then to change course at the end of the day depending on which outcome is observed. The optimal deterministic solutions under the first two scenarios take extreme routes north and south, and the optimal solution in the third scenario is to sail the direct route.

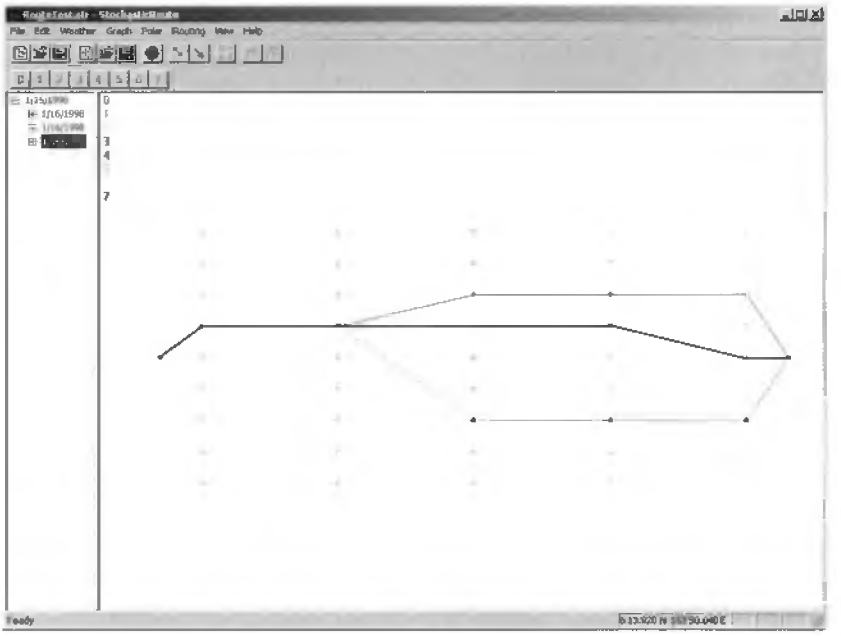


Figure 17.6. *Screen shot from routing application.*

17.5 Optimal design under uncertainty

We conclude this paper with a brief discussion of optimal yacht design under uncertainty. There are a number of levels at which such an optimal design might be attempted. The most ambitious possibility is to model the shape of the hull and appendages of the boat as well as the sails and compute the shape parameters to maximize the speed of the yacht. Although there have been some major advances in numerical shape optimization using CFD in the aerospace industry (see, e.g., [5]), these have not become widely used in yacht design. One reason for this is that CFD software for sail analysis is still imperfect, as compared with the development of codes for the aerospace industry. Despite their apparent similarities with airplane wings, air flow around yacht sails is surprisingly difficult to model. Similarly, CFD codes for hull analysis are not yet accurate or reliable enough to replace towing-tank data. A second reason for the slow adoption of numerical shape optimization in yachting is that the speed of the boat depends on the interplay between hull forces and sail forces, so a shape optimization needs to compute sail shapes and hull shapes at the same time. Although it is not uncommon for CFD to be used to find a keel shape that maximizes its lift-to-drag ratio at a given speed and angle of attack, there have been few attempts (although see [3]) to integrate the keel design into an optimum shape for the yacht as a whole. A third reason for the slow adoption of numerical shape optimization in yachting is that the exact form of the objective function for the design is not easy to model. The objective function for a racing yacht design optimization is not its speed at a specified point of sail in a given wind condition, but rather a complicated function of its speed in various sailing conditions. It is in modeling this aspect of design optimization that stochastic programming comes into play.

Since carrying out a numerical shape optimization for a complete yacht using CFD is an ambitious undertaking, our efforts have been aimed at optimizing the general dimensions of a design using the equations of a VPP. Since these equations depend on the design dimensions x of the yacht (e.g., its sail area, length, hull wetted surface area) we can choose these parameters as well as the control variables u to maximize the performance of the yacht in any point of sail and wind condition (denoted ω). To optimize the performance over a range of conditions,we treat these as scenarios, $\omega \in \Omega$, weighted by probabilities $p(\omega)$ and maximize the expected performance subject to the equilibrium constraints for each scenario to give

$$\begin{array}{lll} \text{P:} \quad \text{maximize} & \sum_{\omega\in\Omega} p(\omega) f(x, u(\omega), w(\omega)) & \\ \text{subject to} & g(x, u(\omega), w(\omega)) = 0, & \omega \in \Omega \text{ (equilibrium conditions)}, \\ & h(x) \le H, & \text{(design constraints)}. \end{array}$$

Here f is a performance criterion, and the last set of constraints limits the design choices.

Apart from technical constraints on the choice of x, the design constraints can represent class rules. For example, the dimensions of IACC yachts are severely restricted by a set of design rules that are specified in a 50-page document (available for downloading from www.rnzys.org.nz/americas/official/official.html). The main rule states that the rated length (L), rated displacement (W), and rated sail area (S) must satisfy the constraint

$$\frac{L + 1.25\sqrt{S} - 9.8\sqrt[3]{W}}{0.679} \le 24.$$

Observe that L, W, and S are not the same as the measured length, displacement, and sail area. They are computed from the measured quantities to include penalty terms if the measured quantities exceed certain bounds. For example, since the IACC technical committee deemed that the measured sail area M should not deviate too much from 285.61 m^2, the following constraint is added:

$$S = M + 0.001M(\sqrt{M} - 16.9)^8.$$

Constraints such as this come from the decisions of a committee rather than accurately modeling any real performance advantage from a large sail area. Stochastic programming models such as P provide a powerful methodology to expose any "corners in the Rule" that might be missed by a designer.

The objective function requires some careful thought, as the appropriate choice of f depends on the circumstances in which the boat is racing. One can illustrate some of the difficulties here by considering a very simple model. Suppose we wish to minimize the expected time to finish the first leg of an America's Cup race in uncertain weather, assuming a constant wind direction but uncertain wind speed. On one hand, one might design a boat assuming that the VMG ($V_s \cos\beta_t$) is an independently and identically distributed random variable in each small time interval. Then the optimal strategy is to maximize $\sum_{\omega\in\Omega} p(\omega)\ V_s(\omega)\cos\beta_t(\omega)$. On the other hand, one might assume a fixed (random) wind speed for the entire leg and then optimize the expected performance over the different possible outcomes for this speed. In the second situation the optimal strategy is to minimize $\sum_{\omega\in\Omega} \frac{p(\omega)}{V_s(\omega)\cos\beta_t(\omega)}$. Since the wind speed has some serial correlation, the real situation will be somewhere between these extremes.

The objective functions above assume that the yacht is racing against the clock, when in fact the yacht will be competing against others. Here it is the relative performance that is required. Suppose there is one other yacht with known performance, so in each weather outcome ω we can estimate its VMG to be $V_c(\omega)$, say. In the first case of no serial correlation, and assuming no interaction between the boats, the number of meters $d(t)$ that the yacht is ahead of the competition after t seconds can be approximated by a random walk with drift. Under this approximation $d(t)$ will be approximately normally distributed with mean μt and variance $\sigma^2 t$, where

$$\mu = \sum_{\omega \in \Omega} p(\omega)[V_s(\omega) \cos \beta_t(\omega) - V_c(\omega)]$$

and

$$\sigma^2 = \sum_{\omega \in \Omega} p(\omega)[V_s(\omega) \cos \beta_t(\omega) - V_c(\omega)]^2 - \mu^2.$$

This variance estimate assumes that the boats see identical winds at all times. One might seek for some fixed time interval t to maximize $\Pr[d(t) > 0]$.

In the second case, where the wind does not vary over the race course but is randomly selected at the start of the race, there are $|\Omega|$ possible outcomes for the race (again assuming that the boats see identical wind) each occuring with probability $p(\omega)$ and yielding a time difference at the next mark of

$$\delta(\omega) = \frac{d}{V_c(\omega)} - \frac{d}{V_s(\omega) \cos \beta_t(\omega)}.$$

Ideally, one seeks to maximize $\Pr[\delta > 0]$, which, however, is not easy to compute. Alternatively one might seek to maximize $E[U(\delta)]$ for some concave utility function U.

The approaches above seek a functional representation of the objective function under a number of assumptions on the wind behavior. In fact the assumptions are likely to be invalid for most practical situations. First, there will always be some degree of serial correlation in the wind speed, so none of the objective functions above will represent the wind variation properly. Second, the yachts will see identical winds only when they are close. Even then, the interaction between the boats is likely to disturb the wind of the trailing boat (sailing upwind). The true nature of the trade-offs can really be determined only by a race modeling simulation program like the ACROBAT model presented in section 17.3. This can be used to estimate a response surface for the probability of beating a given competitor in a given wind speed, as a function of the design parameters, to then be used in the design problem P. Alternatively, the objective function of P and its gradients can be estimated at each iteration of an optimization algorithm by carrying out simulations. Doing this effectively is a subject of ongoing research.

All of the above discussion has focused on variations in the wind speed during the course of a single race. In many locations (such as Auckland) these variations will be smaller than the variations in mean wind speed seen from day to day. Since the finals of the America's Cup regatta span a period of up to two weeks, the design choice must account for these variations. In fact the yachts constructed for the America's Cup are designed and built up to a year before the regatta, and so the probability distribution of average wind speeds (denoted *climatology* by meteorologists) during the race period must be accounted

for. We now give a brief discussion of this problem with specific reference to the recent 2000 America's Cup Defence.

The America's Cup was defended by Team New Zealand on behalf of the the Royal New Zealand Yacht Squadron (RNZYS) in Auckland in a series starting on Saturday, February 26, 2000. The races were scheduled to be run on Saturday, Sunday, Tuesday, Thursday, Saturday, Sunday, Tuesday, Thursday, ..., until one boat won five races. Article 9 of the America's Cup XXX Protocol of April 1996 laid down the precise conditions under which the America's Cup regatta was to be run in 2000. Since this is pertinent to this paper we state the regulations in full:

> If there is no Defender Selection Series then RNZYS may nominate one or two yachts which will be involved in a public unveiling ceremony on the agreed date at least three clear days prior to the first race of the finals of the Challenger Selection Series on which the yachts participating in the Challenger Selection Series are publicly unveiled.
>
> In addition, the challenging and defending yachts for the Match will be part of a public unveiling ceremony which will be held three clear days prior to the first day of the Match. If RNZYS only nominated one yacht to be involved in the public unveiling ceremony which takes place prior to the first race of the finals of the Challenger Selection Series, that yacht shall be the defending yacht. If RNZYS nominated two yachts to be involved in the public unveiling ceremony which took place prior to the first race of the finals of the Challenger Selection Series, then RNZYS may select in its absolute discretion one of those two yachts to be the defending yacht. Both the challenging and defending yachts must have been through the official prematch measurement provided for in Article 10 and have been accepted by RNZYS as Challenger and Defender for America's Cup XXX prior to the date of the public unveiling ceremony for the Match.

Article 17 of the America's Cup XXX Protocol states that "Each Challenger and candidate for the defence may build, acquire or otherwise obtain two New IACC yachts eligible for America's Cup competition." These new boats provided Team New Zealand, along with NZL32 and NZL38 (the two black boats from the 1995 challenge), a collection of boats from which the defending yacht would eventually be chosen. As specified in Article 9 above, Team New Zealand had to make choices which narrowed down the choice of the ultimate defending boat. These choices were made with incomplete, but accumulating, information regarding the challenging boat and the weather conditions for the match. This is a multistage decision process, which takes the following form:

1. Design and build two boats.
2. Discover the Challenger Series finalists, and then choose a portfolio of two boats to defend with.
3. Discover the Challenger Series winner and what the weather will be during the regatta, and then choose the defending yacht from the portfolio.
4. Discover the average wind speed on a given race day and make small modifications to the defending yacht to make it fast for this race.

At each stage of this process, the defender obtains some new information and uses this to make an informed decision, which narrows down the later options. These later options are dependent on the designs of the two new boats, so the specification of these is critical to maximizing the probability of defending successfully.

We proceed to describe a three-stage stochastic programming model for determining the design parameters for the two new boats (denoted by x_1 and x_2), so as to maximize the probability of defending successfully. The first-stage decisions of this model are x_1 and x_2. The second-stage decisions give the portfolio of two boats we settle on before the Challenger Final (denoted by $\{y_1, y_2\}$), which is the most appropriate pair of designs chosen from x_1, x_2, x_3, and x_4, where x_3 and x_4 are the given design parameters of the existing boats (NZL32 and NZL38). The choice of y_1 and y_2 is made with knowledge of the Challenger Series finalists. The third-stage decision is the choice of defender from $\{y_1, y_2\}$. We assume that this decision is made with perfect information of the challenging yacht and the weather to be faced during the America's Cup match. Prior to the completion of the Challenger Final, our knowledge of the weather outcomes faced during the America's Cup match is modeled by a finite number of possible weather scenarios $\omega \in \Omega$, each with probability $p(\omega)$.

As observed by Letcher et al. [9] the choice of defending yacht will depend on the array of challengers. To determine a distribution of challengers, a list of potential challenging yacht designs is constructed. Denote them by $i = 1, 2, \ldots, n$. For every pair of designs (i, j) we can estimate (using a race modeling program like ACROBAT) the probability $\pi(i, j, \omega)$ that design i beats design j in weather outcome ω. Similarly we can estimate the probability $\rho(i, j)$ that i and j will contest the Challenger Final. To do this we might simulate the entire sequence of races in the Challenger Series over a distribution of weather outcomes to be expected over this series (which takes several months). For simplicity we assume here that these weather outcomes are independent of ω.

By an abuse of notation let $\pi(y, j, \omega)$ be the probability that a yacht design with parameters y beats challenger j in weather outcome ω. The evaluation of this function of the design variables and its incorporation into the design optimization presents a major computational challenge. The function can be either evaluated online using simulation or simulated offline to produce a response surface that can be used in an optimization code.

To derive a multistage formulation of this problem, we begin by modeling the final stage, which must choose the better of two candidate yachts y_1 and y_2 to defend the America's Cup against a known challenger in known weather. The optimal objective function of this problem is the probability $Q_i(y_1, y_2, \omega)$ of defending the America's Cup given the second-stage decisions $\{y_1, y_2\}$ and the known state of the world defined by the weather realization ω and the challenging yacht i. The optimal choice of yacht is determined by $\lambda(i, y_1, y_2, \omega)$ solving

$$Q_i(y_1, y_2, \omega) = \max_{\lambda \in [0,1]} \{\lambda \pi(y_1, i, \omega) + (1 - \lambda)\pi(y_2, i, \omega)\}.$$

For each pair (i, j) of potential Challenger Series finalists, the second-stage problem is to select from x_1, x_2, x_3, and x_4 a pair of yachts $\{y_1(i, j), y_2(i, j)\}$, to maximize the probability $P_{ij}(x_1, x_2, x_3, x_4)$ of defending the America's Cup against the winner (out of i and j) of the Challenger Series final. For each (i, j), $\{y_1(i, j), y_2(i, j)\}$ is the solution to the following

mixed integer nonlinear program:

$$\begin{aligned}
\text{maximize} \quad & \sum_{\omega} p(\omega)\pi(i, j, \omega)Q_i(y_1, y_2, \omega) + \sum_{\omega} p(\omega)\pi(j, i, \omega)Q_j(y_1, y_2, \omega) \\
\text{subject to} \quad & y_1 = \delta_1 x_1 + \delta_2 x_2 + \delta_3 x_3 + \delta_4 x_4, \\
& y_2 = \epsilon_1 x_1 + \epsilon_2 x_2 + \epsilon_3 x_3 + \epsilon_4 x_4, \\
& \delta_1 + \delta_2 + \delta_3 + \delta_4 = 1, \\
& \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4 = 1, \\
& \delta_k, \epsilon_k \in \{0, 1\}.
\end{aligned}$$

The first-stage problem is now to select feasible designs x_1 and x_2 to maximize the probability of defending the America's Cup:

$$\begin{aligned}
\text{maximize} \quad & \sum_{i<j} \rho(i, j)P_{ij}(x_1, x_2, x_3, x_4) \\
\text{subject to} \quad & h(x_1) \leq H, \\
& h(x_2) \leq H.
\end{aligned}$$

17.6 Conclusions

This chapter gives the reader an overview of the breadth and diversity of stochastic optimization problems arising in high-performance yacht racing. Yachting as a sporting contest is a fascinating combination of athleticism, skill, teamwork, and strategy. The optimization problems faced by yacht designers and tacticians are essentially stochastic, and so stochastic programming has an enormous contribution to make to this area.

It is unfortunate that there is still some reluctance by designers and tacticians to model problems in this way. Since this paper is included in the Applications section of this volume, it is pertinent to discuss why this might be the case. One fundamental problem is the difficulty and degree of accuracy of the computations that must be carried out. The performance of a yacht as a function of its design variables is a very flat but noisy function. However tiny improvements in performance can be significant over a two-hour race and thus accurate modeling is needed. As mentioned above, the VPP models that form the basis of the techniques we have described here are relatively crude approximations to the true situations, being valid essentially for equilibrium sailing in still water. Since yachts are rarely in equilibrium in still water, VPPs can give a false reading, and America's Cup teams continue to pursue full-scale testing on the water to moderate the outputs from their VPPs. When yacht designers are reluctant to put unlimited faith in the physical models underlying a VPP, it makes it difficult to sell them on a stochastic programming approach based on these models.

Weather routing in a deterministic framework has been used for over ten years in ocean races. In the early days, the navigators would receive faxes of forecast weather maps. These were digitized on board, and then wind fields were forecast using on-board meteorological models. All these tasks were carried out by the navigator in the open ocean, so optimal routes were not computed often. In the current Volvo Ocean Race, forecast wind fields are downloaded to an on-board computer automatically by satellite, and the navigator runs standard software to compute optimal routes. A barrier to the adoption of stochastic

routing as described in this paper is the requirement for navigators to construct scenarios from these wind fields. The development of stochastic routing technologies on yachts over the next decade will lead to scenario trees being computed by on-shore meteorologists and downloaded by satellite to all competitors. Possibly the only input of the navigator then will be to provide scenario probabilities and a utility function.

Acknowledgments

This research has benefited from the support of the University of Auckland Foundation, the New Zealand Foundation for Research Science and Technology, and Team New Zealand. The author would like to acknowledge the contributions of Toby Allsopp, John Birge, Peter Jackson, Shane Henderson, Andrew Mason, Clay Oliver, and David Teirney to this work.

Bibliography

[1] T. ALLSOPP, *Stochastic Weather Routing for Sailing Vessels*, Master of Engineering thesis, School of Engineering, University of Auckland, Auckland, New Zealand, 1998.

[2] N. DAVIES, *A Real-Time Yacht Simulator*, Master of Engineering thesis, School of Engineering, University of Auckland, Auckland, New Zealand, 1990.

[3] S. HARRIES, C. ABT, AND K. HOCHKIRCH, *Hydrodynamic modeling of sailing yachts*, in Proceedings of the 15th Chesapeake Sailing Yacht Symposium, Annapolis, MD, 2001.

[4] J. A. GRETZKY AND J. K. MARSHALL, *The partnership for America's Cup technology: An overview*, in Proceedings of the 11th Chesapeake Sailing Yacht Symposium, Annapolis, MD, 1993.

[5] A. JAMESON AND J. C. VASSBERG, *Computational fluid dynamics for aerodynamic design: Its current and future impact*, in Proceedings of the 39th AIAA Aerospace Sciences Meeting, Reno, NV, AIAA, Reston, VA, 2001.

[6] J. E. KERWIN, *A Velocity Prediction Program for Ocean Racing Yachts Revised to February*, 1978, MIT Ocean Engineering Report 78-11, MIT, Cambridge, MA, 1978.

[7] T. V. LAWSON, *Wind Effects on Buildings: Statistics and Meteorology*, Applied Science Publishers, London, 1980.

[8] J. S. LETCHER, JR., *Handicapping rules and performance of sailing yachts*, in Proceedings of the 1st Chesapeake Sailing Yacht Symposium, Annapolis, MD, 1974.

[9] J. S. LETCHER, JR., J. K. MARSHALL, J. C. OLIVER III, AND N. SALVESEN, *Stars and Stripes*, Scientific American, 257 (1987), pp. 34–40.

[10] T. LINDVALL, *Lectures on the Coupling Method*, John Wiley, New York, 1992.

[11] Y. MASAYUMA, T. FUKASAWA, AND H. SASAGAWA, *Tacking simulation of sailing yachts—Numerical integration of equations of motion and application of neural network technique*, in Proceedings of the 12th Chesapeake Sailing Yacht Symposium, Annapolis, MD, 1995.

[12] P. VAN OOSSANEN, *Predicting the speed of sailing yachts*, in Proceedings of the Centennial Meeting of SNAME, New York, 1993.

[13] A. B. PHILPOTT, S. G. HENDERSON, AND D. P. TEIRNEY, *A Simulation Model for Predicting Yacht Match Race Outcomes*, Technical Report 608, School of Engineering, University of Auckland, Auckland, New Zealand, 2001.

[14] A. B. PHILPOTT, P. S. JACKSON, AND R. M. SULLIVAN, *Yacht velocity prediction using mathematical programming*, Eur. J. Oper. Res., 67 (1993), pp. 13–24.

[15] A. B. PHILPOTT AND A. J. MASON, *Optimizing yacht routes under uncertainty*, in Proceedings of the 15th Chesapeake Sailing Yacht Symposium, Annapolis, MD, 2001.

[16] E. C. SCHLAGETER AND J. R. TEETERS, *Performance prediction software for IACC yachts*, in Proceedings of the 11th Chesapeake Sailing Yacht Symposium, Annapolis, MD, 1993.

[17] D. J. WHITE, *Utility, probabilistic constraints, mean and variance of discounted rewards in Markov decision processes*, OR Spektrum, 9 (1987), pp. 13–22.

Chapter 18

Stochastic Approximation, Momentum, and Nash Play

H. Berglann and S. D. Flåm**

18.1 Introduction

This note briefly explores a novel use of stochastic programming. Specifically, it applies Gupal's [19] stochastic version of Polyak's [25, 26] heavy ball method to model repeated interaction among noncooperative agents. The purpose is to find a Nash equilibrium. Our enterprise has three different motivations.

First, so-called gradient projection algorithms [9], while prominent in optimization theory, offer—maybe as an unintended by-product—valuable input to cognitive sciences. There one asks: How do real agents view their decision problems? How is information processed? What sort of behavior reflects and facilitates optimization? In the end, only empirical data can decide these issues. Casual observation indicates, however, that typical agents approximate data, proceed stepwise, and adapt to circumstances. But these are precisely the features that characterize gradient methods. And a fortiori the same features—whence the attending methods—should fit situations where several agents interact. Such situations are the main objects of game theory—a theory that now provides some unity and coherence to many, diverse inquiries; see books by Hofbauer and Sigmund [22], Gintis [17], or Watson [28]. Indeed, various social sciences, while steadily growing more game-theoretic, increasingly reckon Nash equilibrium as a focal point and key solution concept.[1]

Our second motivation is that such equilibrium—while formalizing stable interaction—often demands more knowledge and rationality from the players than can easily be had in one shot. Usually, strategists need much learning or experience to become clever. Therefore, the Nash solution, if any, begs justification in dynamic terms.[2] Quite simply, we cannot expect

*Department of Economics, University of Bergen, Fosswinckelsgate 6, 5007 Bergen, Norway (helge.berglann @econ.uib.no, sjur.flaam@econ.uib.no).

[1]Additional bonus and impetus comes with making those sciences more experimental; see [18].

[2]Such studies include [16, 23, 27].

a specific Nash equilibrium to approximate real behavior unless some plausible process makes precisely that outcome a stable steady state. In our opinion, any candidate process had better reflect that typical agents

- often are plagued by uncertainty, be it endogenous or exogenous;
- always try to improve their own welfare (or payoff) whenever possible;
- never quite know all strategic possibilities, intentions, or consequences;
- persistantly form local perspectives and approximations; and
- invariably hesitate in making quick or large adjustments.

These features, each part of gradient methods, are also embodied in stochastic approximation theory [6, 24]. That theory leans heavily on differential equations and subscribes to a tradition that goes back to classical mechanics, notably to Newton's claim that the initial state of a mechanical system determines its future development.

Our third motivation stems from the fact that many noncooperative games resemble nonconvex programming in that "gradients" fail to be monotone. Then, adding some momentum, viscosity, or inertia to the gradient may help. Again, on this account as well, Newtonian mechanics offer good guidance.

This chapter views the play of noncooperative games, featuring uncertain payoffs, from a Newtonian perspective; see also [13]. Alternatively, one may read the paper as dealing with single-agent stochastic programming concerned with global optimization—or as many-agent parallel computation of stochastic equilibrium. Subsequent arguments are organized, however, around a noncooperative stage game repeated time and again. Technicalities and proofs are found in the references.

18.2 The game

There is a fixed, finite set I of players. Agent $i \in I$ is constrained to choose his strategy x_i from a nonempty compact convex subset X_i of a Euclidean space. He always seeks to improve his own expected payoff

$$\pi_i(x_i, x_{-i}) := E(\Pi_i(x_i, x_{-i}, \omega)).$$

Here $x_{-i} =: (x_j)_{j \neq i}$ denotes the part of the overall strategy profile $x = (x_i)$ that is controlled by the rivals of is. The elementary event ω belongs to a complete probability space (Ω, σ, μ), with respect to which one takes the mathematical expectation $E(\cdot) = \int_\Omega \cdot \mu(d\omega)$. Each bivariate function $(x, \omega) \mapsto \Pi_i(x, \omega) \in \mathbb{R}$ is concave, continuously differentiable in $x_i \in X_i$, and integrable in $\omega \in \Omega$.

Of prime interest are points $x \in X := \Pi_{i \in I} X_i$, where each expected marginal payoff $m_i(x) := \frac{\partial}{\partial x_i} \pi_i(x)$ is nil or normal to X_i. That is, letting P_i denote the orthogonal projection onto X_i, we seek a fixed point $x = (x_i)$ of the dynamic system

$$x_i \leftarrow P_i[x_i + s m_i(x)] \quad \text{for all } i \text{ and arbitrary } s > 0. \tag{18.1}$$

Any such point is a Nash equilibrium. Process (18.1) amounts to a decentralized projected gradient method. It portrays fairly myopic parties, each trying to improve a linear approximation of their own expected payoff. Modern, stochastic versions of such methods incorporate two quite common aspects of human behavior. First, mean values (i.e., mathematical expectations E) are costly—and sometimes plainly impossible—to compute. Second, information concerning levels and gradients is readily available only at the current point. So, in our optic, letting $M_i(x, \omega) := \frac{\partial}{\partial x_i} \Pi_i(x, \omega)$ denote agent i's realized marginal payoff, one might hope to have almost sure convergence of the following stochastic process: for each i recursively posit

$$x_i^{k+1} := P_i[x_i^k + s_k M_i(x^k, \omega^k)]. \tag{18.2}$$

Here, at stage $k = 0, 1, \ldots$, occurs a new event $\omega^k \in \Omega$, independently sampled according to the prescribed measure μ. As input at that stage k we also use a positive *stepsize* s_k, selected a priori subject to

$$\sum s_k = +\infty \text{ and } \sum s_k^2 < +\infty. \tag{18.3}$$

The hope that process (18.2) converges is well founded when $x \mapsto m(x) := [m_i(x)]_{i \in I}$ is globally monotone; see [10, 11, 12]. Otherwise, there are good reasons to be worried about the long-run behavior. Therefore, our object here is to expand on gradient methods while preserving their many appealing properties.

Like (18.2) the procedure considered below does not presume much foresight, experience, competence, or optimization on the part of players. In fact, it merely reflects iterated, noncoordinated, time-consuming pursuit of better payoffs. It does, however, modify the first-order gradient dynamics (18.2) by adding a second-order heavy-ball momentum—just like the harmonic, damped oscillator of classical mechanics. Essentially, instead of assuming that player i pursues the gradient method $0 = m_i(x) - \dot{x}_i$ we posit that he rather drives the second-order process $\ddot{x}_i = m_i(x) - \dot{x}_i$ (see [1, 2, 3, 4]). The latter must be suitably adapted though, to account for discrete time, uncertainty, and constraints. This is done next.

18.3 Repeated play

Let $\{\omega^k\}$ be a sequence of independent realizations of ω, each having distribution μ. As a model of repeated play we advocate that iteratively at stages $k = 0, 1, \ldots$ each individual i updates his current strategy x_i^k and velocity v_i^k by the rule

$$\left.\begin{aligned} x_i^{k+1} &:= P_i[x_i^k + s_k v_i^k], \\ v_i^{k+1} &:= v_i^k + s_k\{M_i(x^k, \omega^k) - v_i^k\}. \end{aligned}\right\} \tag{18.4}$$

As above, P_i denotes orthogonal projection onto X_i. Also as above, the parameter $s_k > 0$ is the stepsize used at stage k, selected a priori subject to (18.3). The initial points (x_i^0, v_i^0), $i \in I$, are determined by accident or historical factors better discussed in each particular setting.

When I is a singleton, method (18.4) has already been studied by Gupal [19] who added to the heavy ball method of Polyak [25, 26]. To appreciate (18.4) it helps to endow

that process with a clock which shows accumulated "time" $t_k := s_0 + \cdots + s_{k-1}$ ($t_0 := 0$) at the onset of stage k. Then, on writing $x_i(t_k) := x_i^k$ and $v_i(t_k) := v_i^k$, system (18.4) assumes the form

$$\{x_i(t_{k+1}) - x_i(t_k)\}/s_k := \{P_i[x_i(t_k) + s_k v_i(t_k)] - x_i(t_k)\}/s_k,$$
$$\{v_i(t_{k+1}) - v_i(t_k)\}/s_k := M_i(x(t_k), \omega^k) - v_i(t_k).$$

Since $s_k = t_{k+1} - t_k \to 0^+$, behind (18.4) there lurks, in expectation and long-run limit, the determinstic differential system

$$\left.\begin{aligned} \dot{x}_i &= P_{T_i x_i}[v_i], \\ \dot{v}_i &= m_i(x) - v_i. \end{aligned}\right\} \tag{18.5}$$

Orthogonal projection $P_{T_i x_i}$ is done here onto the tangent cone $T_i x_i := cl\mathbb{R}_+(X_i - x_i)$ of X_i at $x_i \in X_i$; see [21, Proposition III 5.3.5]. By a solution to (18.5) we understand an absolutely continuous profile $0 \le t \mapsto [x(t), v(t)] = [x_i(t), v_i(t)]_{i \in I}$ which satisfies (18.5) almost everywhere. For stability suppose the potential energy

$$0 \le t \mapsto \int_0^t \sum_{i \in I} m_i(x(\tau)) \cdot v_i(\tau) d\tau \tag{18.6}$$

remains bounded above along solution trajectories of (18.5).

Theorem 18.1 (convergence of repeated play [15]). *Suppose system* (18.5) *has unique solution trajectories. Then, under the hypotheses above and the assumption that Nash equilibria are isolated, any discrete-time trajectory* (x^k, v^k) *generated by* (18.4) *must be such that* x^k *converges almost surely to a Nash equilibrium.*

We have also considered replacing the second equation in (18.5) with

$$\dot{v}_i = P_{T_i x_i}[m_i(x)] - P_{T_i x_i}[v_i]. \tag{18.7}$$

Then substitute

$$0 \le t \mapsto \int_0^t \sum_{i \in I} P_{T_i x_i} m_i(x(\tau)) \cdot \dot{x}_i(\tau) d\tau \tag{18.8}$$

for the potential energy (18.6). The prospects of maintaining that energy bounded above are then often better. When applying the alternative format (18.7), we replaced the last equation in (18.4) with

$$v_i^{k+1} := v_i^k + P_i[x_i^k + s_k M_i(x^k, \omega^k)] - P_i[x_i^k + s_k v_i^k].$$

In either case, at any stage k, player i might, quite reasonably, first update his velocity v_i^{k+1} as prescribed and thereafter set $x_i^{k+1} := P_i[x_i^k + s_k v_i^{k+1}]$. In the subsequent simulations we observe that this practice speeds up convergence.

18.4 Time-homogeneous play

When telling our tale about repeated play, we find it difficult sometimes to argue in favor of the time-inhomogeneous system (18.4). So, what happens if s_k is constant? Clearly,

fixing this parameter is risky, notably with stiff systems. To address that issue set $d_i = d_i(x, v_i, \omega) := M_i(x, \omega) - v_i$. Iteration (18.4) then comes in autonomous, more tractable form

$$\left.\begin{aligned} x_i^{k+1} &:= P_i[x_i^k + sv_i^k], \\ v_i^{k+1} &:= v_i^k + sd_i^k. \end{aligned}\right\} \tag{18.9}$$

In simulations of (18.9), to hedge against stiffness and facilitate convergence, one may replace d_i^k with a weighted sum $a_i d_i^k + b_i d_i^{k-1}$, using thus

$$x_i^{k+1} := P_i[x_i^k + sv_i^k], \qquad v_i^{k+1} := v_i^k + s(a_i d_i^k + b_i d_i^{k-1}). \tag{18.10}$$

The parameters a_i, b_i could account for accumulated learning on how to adapt in a complex dynamic environment.[3] The second equation in (18.10) is strikingly similar to a control algorithm commonly used in process industries, namely, the so-called proportional-integral-controller; see [5, 20, 29]. We take advantage of this to find appropriate a_i, b_i in the application that follows next.

18.5 An application

Let each $i \in I$ be an oligopolist [8] who supplies the quantity $x_i \geq 0$ of a homogeneous, perfectly divisible good to a common market. Thereby he receives sales revenues px_i and incurs convex, continuously differentiable production costs $c_i(x_i)$. The price is determined by a smooth inverse demand curve subject to stochastic fluctuations. More precisely, the realized price equals $p = \omega P(Q)$ with $\omega > 0$, $E\omega = 1$, and $Q := \sum_{i \in I} x_i$. Thus

$$\Pi_i = \omega P(Q)x_i - c_i(x_i) \quad \text{and} \quad M_i = \omega P(Q) + \omega P'(Q)x_i - c_i'(x_i).$$

Figures 18.1–18.3 depict individual supply x_i over stages k with constant stepsize $s = 1$, as generated by (18.10). There are 10 players; ω has a lognormal distribution with standard deviation 0.2, $P(Q) = 10 - Q$, $c_i(x_i) = x_i$, $x_i^0 = 1$, $v_i^0 = 0$, and finally, $d_i^0 = 0$ for all i. The resulting Nash equilibrium x is unique with all $x_i = 0.818$.

The resemblance with the above-mentioned controller algorithms made us look in the engineering literature for methods to determine efficient values of a_i and b_i in (18.10). The widespread use of such algorithms notwithstanding, none is generally accepted for tuning these parameters. The empirical method developed by Ziegler and Nichols [29] still holds good ground, and it has the great advantage of requiring very little information. It gave us the values of a_i and b_i that label Figure 18.1. These are used by all agents. The dashed line in the figure shows behavior in the deterministic case (when $\omega \triangleq 1$), while in the dotted curve, ω is sampled anew for each k.

Figure 18.2 brings out responses when all agents halve the values a_i and b_i employed in Figure 18.1. Absent uncertainty, the time needed to reach a steady level now becomes longer. Present uncertainty (using the same series ω^k as in Figure 18.1), it causes less fluctuations than before.

[3] In fact, if agent i were new to the kind of dynamic process in question, his optimal behavior might entail experimentation to determine the said parameters. Most likely agents would learn from each other and, if possible, imitate those who do well. Henceforth assume that each i has previous experience, perhaps from similar processes, and has appropriately tuned his a_i and b_i.

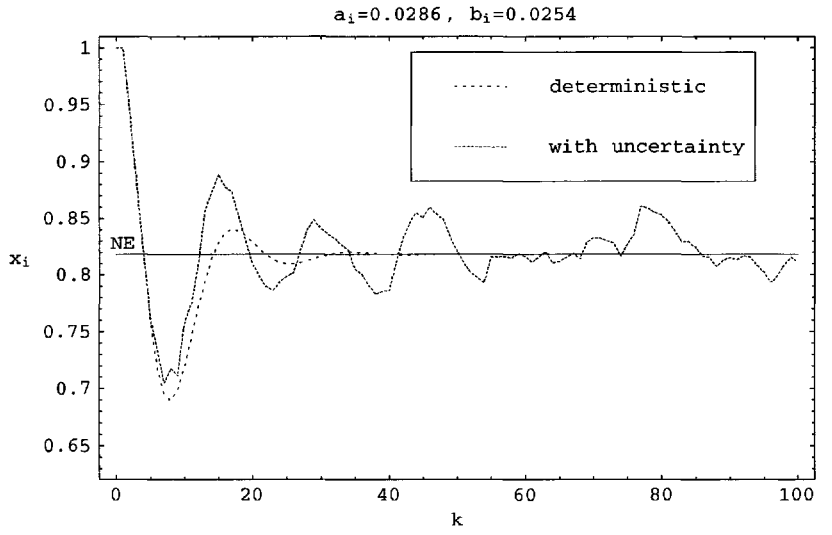


Figure 18.1. *Supply x_i over stages k when all players employ parameters a_i and b_i determined by the Ziegler–Nichols method.*

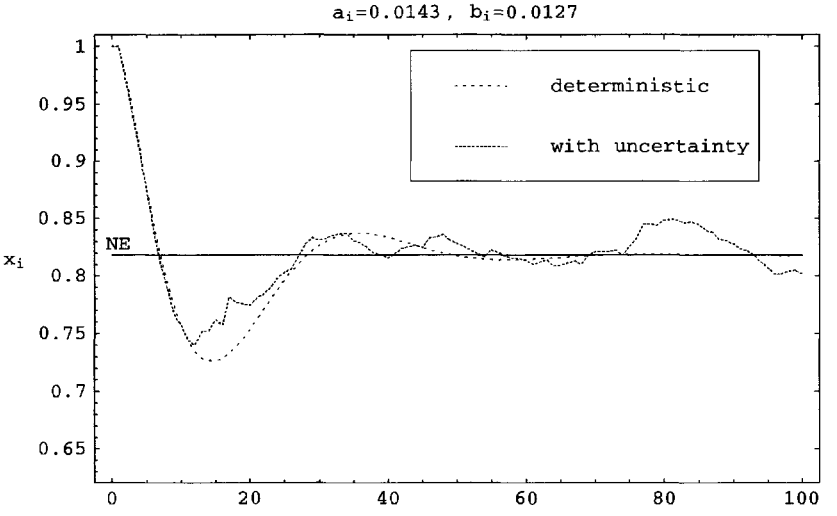


Figure 18.2. *Supply x_i over stages k when all players employ parameters a_i and b_i with values half of the size determined by the Ziegler–Nichols method.*

Figure 18.3 illustrates what happens when the parameters a_i, b_i differ across agents. The five first players use values listed in Figure 18.1; the others use those mentioned in Figure 18.2. Members of the first group adapt fastest initially. Differences across agents make for slower convergence in the deterministic case.

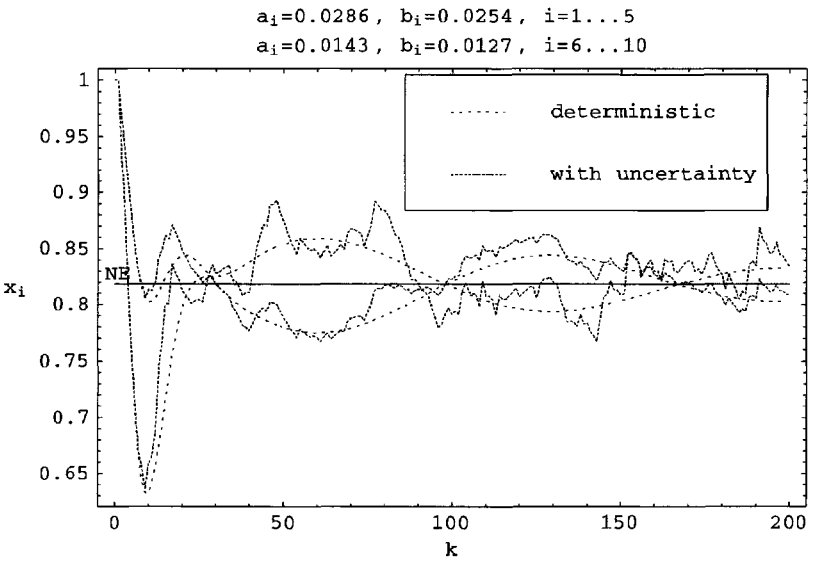


Figure 18.3. *Supply x_i over stages k with different a_i, b_i. Group* 1 *uses parameters a_i and b_i determined by the Ziegler–Nichols method, while group* 2 *halves these values.*

18.6 Concluding remarks

We presented (18.4) as a tale about repeated play of noncooperative, constrained games. A main motivation behind the heavy-ball philosophy is to handle instances where the solution set is disconnected. A supplementary aim, referred to as equilibrium selection [27], amounts to exploiting uncertainty or blurred data to arrive at particularly stable solutions. In fact, randomness, if not already a key ingredient, could artificially be introduced to escape from unstable equilibria.

When I is a singleton, this paper fits the frames of single-agent optimization under uncertainty. In that regard (18.4) has something to offer in two respects. First, process (18.9) is amenable to parallel computing [7]. Second, following [3], the same process is applicable for global optimization—or for the selection of "good," "robust" stationary points; see also [1, 2, 4]. Admittedly, method (18.10) may require, for efficient operation, some auto-tuning of the parameters a_i, b_i. Appropriate routines to that effect are found in the engineering literature on control; see, for instance, [5].

Acknowledgments

The first author gratefully acknowledges the support of NFR. The second author thanks S. Wallace and W. Ziemba for their useful comments and Ruhrgas, Røwdes Stiftelse, and Meltzers høyskolefond for their support.

Bibliography

[1] F. ALVAREZ, *On the minimizing property of a second order dissipative system in Hilbert spaces*, SIAM J. Control Optim., 38 (2000), pp. 1102–1119.

[2] F. ALVAREZ AND J. M. PÉREZ, *A dynamical system associated with Newton's method for parametric approximations of convex minimization problem*, Appl. Math. Optim., 38 (1998), pp. 193–217.

[3] H. ATTOUCH, X. GOUDOU, AND P. REDONT, *The heavy ball with friction method* I, *The continuous dynamical system: Global exploration of the local minima of a real-valued function by asymptotic analysis of a dissipative dynamical system*, Comm. Contemp. Math., 2 (2000), pp. 1–34.

[4] H. ATTOUCH AND P. REDONT, *The second-order in time continuous Newton method*, in Approximation, Optimization and Mathematical Economic, M. Lassonde, ed., Physica-Verlag, Heidelberg, 2001, pp. 4–36.

[5] R. BANDYOPADHYAY AND D. PATRANABIS, *A fuzzy logic based PI auto-tuner*, ISA Trans., 37 (1998), pp. 227–235.

[6] M. BENAIM, *A dynamical system approach to stochastic approximations*, SIAM J. Control Optim., 34 (1996), pp. 437–472.

[7] D. P. BERTSEKAS AND J. N. TSITSIKLIS, *Parallel and Distributed Computation: Numerical Methods*, Prentice–Hall, New York, 1989.

[8] A. COURNOT, *Recherches sur les principes mathématiques de la théorie des richesses*, Riviere and Cie, Paris, 1838.

[9] Y. M. ERMOLIEV AND R. J.-B. WETS, EDS., *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin, 1988.

[10] S. D. FLÅM, *Approaches to economic equilibrium*, J. Econ. Dynam. Control, 20 (1996), pp. 1505–1522.

[11] S. D. FLÅM, *Restricted attention, myopic play, and the learning of equilibrium*, Ann. Oper. Res., 82 (1998), pp. 473–482.

[12] S. D. FLÅM, *Learning equilibrium play: A myopic approach*, Comput. Optim. Appl., 14 (1999), pp. 87–102.

[13] S. D. FLÅM, *Repeated play and Newton's method*, Internat. Game Theory Rev., 2 (2001), pp. 141–154.

[14] S. D. FLÅM, *Approaching equilibrium in parallel*, in Inherently Parallel Algorithms in Feasibility and Optimization and Their Applications, D. Butnariu, Y. Censor, and S. Reich, eds., North-Holland, Amsterdam, 2001, pp. 267–278.

[15] S. D. FLÅM, *Optimization under uncertainty using momentum*, in Dynamic Stochastic Optimization, K. Marti, Y. Ermoliev, and G. Pflug, eds., Lecture Notes in Econ. and Math. Syst. 532, Springer-Verlag, Berlin, 2004, pp. 249–256.

[16] D. Fudenberg and D. K. Levine, *The Theory of Learning in Games*, MIT Press, Cambridge, MA, 1998.

[17] H. Gintis, *Game Theory Evolving*, Princeton University Press, Princeton, NJ, 2000.

[18] J. K. Goeree and C. A. Holt, *Ten little treasures of game theory and ten intuitive contradictions*, Amer. Econ. Rev., 91 (2001), pp. 1402–1422.

[19] A. M. Gupal, *Stokhasticheskie Methody Resheniya Negladkikh Extremal'nykh Zadach* (Stochastic Methods for Solving Nonsmooth Extremal Problems), Naukova Dumka, Kiev, 1979.

[20] C. C. Hang, K. J. Astrøm, and Q. G. Wang, *Relay feedback auto-tuning of process controllers: A tutorial review*, J. Process Control, 12 (2002), pp. 143–162.

[21] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex Analysis and Minimization Algorithms*, Springer-Verlag, Berlin, 1993.

[22] J. Hofbauer and K. Sigmund, *Evolutionary Games and Population Dynamics*, Cambridge University Press, Cambridge, UK, 1998.

[23] J. Hofbauer, *From Nash and Brown to Maynard Smith: Equilibria, dynamics and ESS*, Selection, 1 (2000), pp. 81–88.

[24] J. Hofbauer and W. H. Sandholm, On the global convergence of fictitious play, Econometrica, 70 (2002), pp. 2265–2294.

[25] B. T. Polyak, *Some methods of speeding up the convergence of iteration methods*, Zh. Vychisl. Mat. Mat. Fiz., 4 (1964), pp. 1–17.

[26] B.T. Polyak, *Introduction to Optimization*, Optimization Software, New York, 1987.

[27] L. Samuelson, *Evolutionary Games and Equilibrium Selection*, MIT Press, Cambridge, MA, 1997.

[28] J. Watson, *Strategy*, Norton, New York, 2002.

[29] J. G. Ziegler and N. B. Nichols, *Optimum settings for automatic controllers*, Trans. ASME 64 (1942), pp. 759–768.

This page intentionally left blank

Chapter 19

Stochastic Optimization for Lake Eutrophication Management

Alan J. King, László Somlyódy,† and Roger J.-B. Wets‡*

19.1 Introduction

Man-made (or artificial) eutrophication has been considered as one of the most serious water quality problems of lakes during the last 20-plus years. Increasing discharges of domestic and industrial waste water and the intensive use of crop fertilizers—all leading to growing nutrient loads of the recipients—can be mentioned among the major causes of this undesirable phenomenon. The typical symptoms of eutrophication are, among others, sudden algal blooms, water coloration, floating water plants and debris, excreation of toxic substances causing taste and odor problems of drinking water, and fish kills. These symptoms can easily result in limitations of water use for domestic, agricultural, industrial, or recreational purposes.

One of the major features of artificial eutrophication is that although the consequences appear within the lake, the cause—the gradual increase of nutrients (various phosphorous and nitrogen compounds) reaching the lake—and most of the possible control measures lie in the region. Consequently, eutrophication management requires analysis of complex interactions between the water body and its surrounding region. In the lake, different biological, chemical, and hydrophysical processes—all being time and space dependent, furthermore nonlinear—are important, while in the region one must take into account human activities generating nutrient, residuals, and control measures determining that portion of the emission which reaches the water body.

Eutrophication management requires a sound understanding of all these processes and activities which, in fact, belong to quite diverse disciplines. Additionally, various uncer-

*IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 (kingaj@us.ibm.com).
†Budapest University of Technology and Economics, H-1111 Budapest, Hungary (somlyody@vsct.bme.hu).
‡Department of Mathematics, University of Califomia at Davis, Davis, CA 95616 (rjbw@math.ucdavis.edu).

tainties and stochastic features of the problem have to be taken into account—for example, the estimation of loads from infrequent observations and the dependence of water quality on hydrologic and meteorologic factors, respectively. The fact that we are dealing with a stochastic environment is especially important for shallow lakes, primarily due to the absence of thermal stratification that predicates a much more definite response to randomness, as would be the case for deep lakes.

This chapter discusses an approach that allows the combined use of descriptive, simulation, and management optimization models. This is discussed in section 19.2. The derivation of the aggregated lake and planning type nutrient load models to be used in the management model is the subject of section 19.3. Alternative management models are formulated in section 19.4. Two of them were implemented: a "true'" stochastic model (which uses as the starting point of the iterative solution procedure the corresponding deterministic model) and a linear programming approach capturing stochastic features of the problem through a linearized expectation-variance model. A brief survey of the solution approach to the stochastic problem is presented in section 19.5. In section 19.6 the methods are applied to Lake Balaton and the results compared.

This chapter is based on the original technical report [11], parts of which appeared in [12].

19.2 The approach

The approach to eutrophication and eutrophication management is based on the idea of decomposition and aggregation [7, 9]. The first step is to *decompose* the problem into smaller, tractable units forming a hierarchy of issues (and models), such as biological and chemical processes in the lake, sediment-water interaction, water circulation and mass exchange, nutrient loads, watershed processes, and possible control measures, as well as the influence of uncontrollable meteorological factors, etc. This step is followed by *aggregation*, the aim of which is to preserve and integrate only the issues that are essential for the higher level of the analysis, ruling out unnecessary details. The procedures followed for the derivation of the eutrophication management optimization model (EMOM) presented in this paper may be found in [8] and [10].

The major assumptions we make for the application of EMOM to Lake Balaton are as follows:

1. The lake is shallow with vertically uniform water quality.

2. The lake can be subdivided into sequentially connected basins.

3. The lake is phosphorus (P) limited, like most water bodies, and thus nutrients other than P are not involved in the analysis.

4. A single water quality indicator, the maximum annual chlorophyll-a concentration $(\text{Chl} - \text{a})_{\max}$, is used for defining the trophic state and the goals of management.

5. A linear relationship holds between $(\text{Chl} - \text{a})_{\max}$ and the annual average P load.

6. The management horizon is short-term (a few years). Longer-term renewal processes between the lake and its sediment layer and the staging of investments are out of the scope of the present effort.

7. Only certain types of P sources and associated control alternatives are taken into account.

19.3 Formulation of the stochastic model

Based on the assumptions made in the approach and the insights gained from the study on Lake Balaton, the short-term response of water quality to load reduction—taking into account macroscopic effects of biological and biochemical processes, interbasin mass exchanges, and the influence of stochastic factors—can be written as follows [8]:

$$\mathbf{Y} = E\{\mathbf{Y}_0\} + \mathbf{w} - (D + d\mathbf{w}) \Delta \mathbf{L}, \tag{19.1}$$

where the elements of the m-vector $\mathbf{Y}$ represent the water quality in the m basins as measured by $(\text{Chl} - \text{a})_{\max}$, the m-vector $\mathbf{Y}_0$ is the uncontrolled nominal state, and the symbol E denotes expectation. The m-vector $\Delta\mathbf{L}$ expresses the reduction in P load due to controls

$$\Delta\mathbf{L} = E\{\mathbf{L}_0\} - \mathbf{L}, \tag{19.2}$$

where the elements of $\mathbf{L}$ are the annual mean volumetric biologically available P load in each basin $i = 1, \ldots, m$. Biologically available P refers to the fraction of P that can be taken up by algae and thus contribute to short-term trophic status of the water body. The random vector $\mathbf{w}$ represents the impact of noncontrollable meteorological factors in each basin. (Stochastic variables and parameters are represented in boldface.)

The elements of the matrix D are the reciprocals of lumped reaction rates. The main diagonal comprises primarily the effect of biological and biochemical processes in the basins, and the off-diagonal elements refer to interbasin exchange due to hydrological flow and mixing. The meaning of the slopes d_i is similar to that of the diagonal elements of D. The term $d_i\mathbf{w}_i \Delta \mathbf{L}_i$ expresses a change in the random component of the water quality indicator in basin i due to the impact of weather. Of course, the effect of the random fluctuations $\mathbf{w}_i$ caused by meteorology decreases if the loads diminish.

Next we model the P loads. The units of $\mathbf{L}_{0i}$ and $\mathbf{L}_i$ are measured in [mg/m^{3}d]. The absolute annual mean load $\mathbf{L}^a$ is the daily flow [mg/d] averaged over an entire year, thus

$$\mathbf{L}_i = \mathbf{L}_i^a / V_i, \tag{19.3}$$

where V_i is the volume of basin i. To model the term (19.2) we must first analyze the contributions to the absolute annual mean load $\mathbf{L}_i^a$ and then divide through by the volume. We consider three P sources, as indicated in Figure 19.1:

1. direct point-source sewage L_S,

2. indirect point-source sewage flowing into a tributary of the lake L_{SN},

3. miscellaneous load from various point and nonpoint sources.

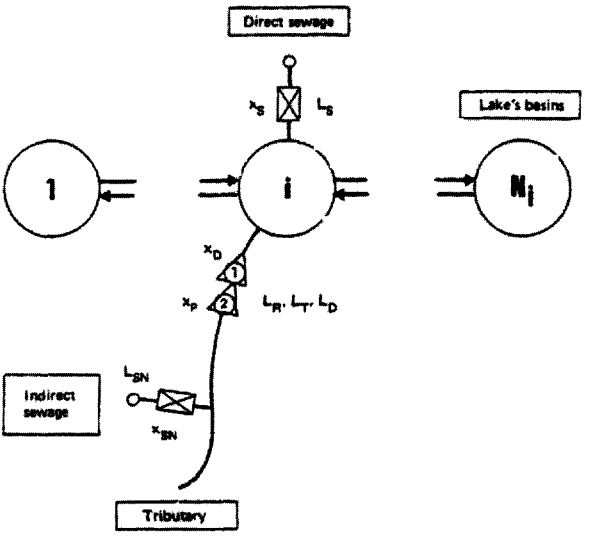


Figure 19.1. *Simple illustration of lake eutrophication and nutrient load modeling.*

The sewage loads L_S and L_{SN} are known, deterministic, and biologically available. The uncertain biologically available (reactive) portion of the tributary load is

$$\mathbf{L}_R = \mathbf{L}_D + \delta(\mathbf{L}_T - \mathbf{L}_D), \tag{19.4}$$

where $\mathbf{L}_D$ is the dissolved reactive P load and $\delta(\mathbf{L}_T - \mathbf{L}_D)$ is the proportion of the undissolved P load (the difference between the total P load and the dissolved P load) that becomes available through biological or biochemical processes. The availability ratio δ is about 0.2.

The basic control options for the P loads are as follows:

1. precipitation by sewage treatment plants for both the direct and tributary point-source sewage loads, with corresponding control variables x_S and x_{SN}, respectively, considered as continuous variables;

2. prereservoir systems, established on tributaries before the water enters the lake, consisting of two segments: one that removes particulate P through sedimentation and one that removes dissolved P through benthic eutrophication in reed basins, sorption, etc., with corresponding control variables x_P and x_D, respectively. The prereservoir control variables should normally be considered as binary {0, 1} variables but for numerical tractability are modeled as continuous.

The control variables are modeled as removal coefficients with

$$0 \le r^- \le x \le r^+ \le 1. \tag{19.5}$$

Consider a simple situation for a single basin, as in Figure 19.1, with one tributary with a single point-source sewage inflow and one point-source direct sewage inflow. The original uncontrolled load $\mathbf{L}$ can be expressed as

$$\mathbf{L}_0^a = \mathbf{L}_D + \delta(\mathbf{L}_T - \mathbf{L}_D) + L_S + L_{SN} + \mathbf{L}_{NC}, \tag{19.6}$$

where $\mathbf{L}_{NC}$ is the portion of the load that is beyond the controls considered. The controlled load of the basin is

$$\mathbf{L}^a = (1-x_D)\,[\mathbf{L}_D - (1-r_t)x_{SN}L_{SN}] + \delta(1-x_P)(\mathbf{L}_T - \mathbf{L}_D) + (1-x_S)L_S + \mathbf{L}_{NC}, \quad (19.7)$$

where r_t is the retention coefficient defining that portion of P from the tributary sewage discharge that is retained in the tributary and does not reach the lake. There is an obvious interaction between the impact of sewage treatment plants on a given tributary x_{SN} and the impact of the prereservoir segment x_D on the same tributary, which is discussed below. With (19.6) and (19.7) we obtain an expression for the amount of dissolved P that is removed from the lake and its tributaries due to controls

$$\begin{aligned}\Delta\mathbf{L}^a := E\{\mathbf{L}_0^a\} - \mathbf{L}^a = {} & x_D\,[E\{\mathbf{L}_D\} - (1-r_t)x_{SN}L_{SN}] \\ & + (x_D - 1)\,[\mathbf{L}_D - E\{\mathbf{L}_D\}] \\ & + \delta\,[(x_P - 1)(\mathbf{L}_T - \mathbf{L}_D) + E\{\mathbf{L}_T - \mathbf{L}_D\}] \\ & + (1-r_t)x_{SN}L_{SN} + x_S L_S.\end{aligned} \quad (19.8)$$

The terms in (19.8) have been rearranged for interpretive purposes. With this rearrangement, one can see that

- the first and fourth terms express the reduction in the *expectation* of the tributary's dissolved P load,
- the second term represents the effects of the reed basin on the *fluctuations* of the tributary's dissolved P load,
- the third term gives the modification in the particulate P load of the tributary, and
- finally, the fifth term expresses the control of the direct sewage load on the lake.

If we set all the control variables to zero in (19.8), we obtain the fluctuations in the original uncontrolled load, the expectation of which is zero.

It is apparent from (19.8) that the tributary load can be controlled by P precipitation at sewage treatment plans and/or by sedimentation and benthic removal in prereservoirs. The former influence only the expectation, whereas the latter influence linearly both the expectation and the variance.

Equation (19.8) is nonlinear in the control variables because of the product term $x_D \cdot x_{SN}$, which may cause difficulties in the optimization scheme. There are many ways to treat this issue. In the present case, the surface-dependent character of benthic P removal in the second segment of the prereservoir system offers a possibility. Generally, for a reed reservoir one cannot estimate more than the P removal per unit of surface area, independent of the inflow concentration. This suggests that the effect of x_D can be approximated in terms of the original uncontrolled load, and the term involving $x_D \cdot x_{SN}$ can be dropped from (19.8). The price of this elimination of nonlinearity is twofold:

1. An upper limit should be specified for the impact of x_D stating that no more nutrient can be removed than that which reaches the lake via the tributary. In expectation, this constraint reads

$$x_D E\{\mathbf{L}_D\} \leq E\{\mathbf{L}_D\} + (1-r_t)x_{SN}L_{SN}. \quad (19.9)$$

This relation should in fact be applied to all realizations of $\mathbf{L}_D$, but this would introduce a stochastic constraint difficult to manage in the optimization.

2. A new variable $x_U \geq x_D$ should replace x_D in the second term of (19.8) to account for the fact that the impact of the reservoir system on the *fluctuations* $\mathbf{L}_D - E\{\mathbf{L}_D\}$ is not restricted by the condition (19.9).

The general situation, when the ith basin is fed by N_1 direct sewage discharges and N_2 tributaries each with M_m indirect sewage discharges, becomes

$$\begin{aligned}\Delta \mathbf{L}_i^a = \sum_{m=1}^{N_2} \{ & x_D^m E\{\mathbf{L}_D^m\} \\ & + (x_U^m - 1)\left[\mathbf{L}_D^m - E\{\mathbf{L}_D^m\}\right] \\ & + \delta\left[(x_P^m - 1)(\mathbf{L}_T^m - \mathbf{L}_D^m) + E\{\mathbf{L}_T^m - \mathbf{L}_D^m\}\right] \\ & + \sum_{l=1}^{M_m}(1 - r_t^{ml}) x_{SN}^{ml} L_{SN}^{ml} \} \\ & + \sum_{n=1}^{N_1} x_S^n L_S^n,\end{aligned} \tag{19.10}$$

and the equivalent version of (19.9) becomes

$$x_D^m E\{\mathbf{L}_D^m\} \leq E\{\mathbf{L}_D^m\} + \sum_{l=1}^{M_m}(1 - r_t^{ml}) x_{SN}^{ml} L_{SN}^{ml} \tag{19.11}$$

for $m = 1, \ldots, N_2$.

Observations and careful analysis of the load (point versus nonpoint source contributions) and watershed are required for the derivation of the uncertain load components $\mathbf{L}_D$ and $\mathbf{L}_T$. Unfortunately, insufficient observations, short historical data, and our lack of understanding make the problem quite difficult; see [1, 2]. In general they have positive lower bounds and can be characterized by strongly skewed distributions. Very often they can be expressed as simple functions of annual mean streamflow rates $\mathbf{Q}$, which generally have much longer historical records than those for P loads. We return to this issue of the derivation of the load distributions in section 19.6.

The final element of the model concerns the budget constraint. The cost of implementing the control options will be a combination of fixed costs for construction and cost functions capturing the exponential growth in capital outlay and operating expense for increased P removal rates. For tractability in the optimization, the costs are modeled by piecewise linear functions increasing in the size of the control variable. To select among management alternatives of different investment costs (IC) and operational, maintenance, and repair costs (OC), the total annual cost (TAC) term is used,

$$\text{TAC} = \sum_j (\text{OC}_j + \alpha_j \text{IC}_j), \tag{19.12}$$

where α_j is the capital recovery factor for project j. In the planning model the TAC is limited by annual budgetary constraints

$$\text{TAC} \leq \beta, \tag{19.13}$$

where β is the annual allocation of budget to the lake treatment plan, or expressing this in terms of the control variables,

$$\sum_j c_j(x_j) \leq \beta. \tag{19.14}$$

A standard technique represents (19.14) as a linear constraint involving variables corresponding to each linear piece of $c_j(\cdot)$.

19.4 Formulation of the eutrophication management optimization model

There are a number of variants available in the building of the management optimization model that allow us to capture the stochastic features of the water quality management problem. For convenience of presentation, we substitute, regroup terms, and reindex the control variables in the model equations (19.1) through (19.10) to obtain an affine relation for the water quality indicators $(\mathbf{y}_i)_{i=1}^m$ of the type

$$\mathbf{y} = \mathbf{T}x - \mathbf{h}, \tag{19.15}$$

where $\mathbf{h}_i$ incorporates all the noncontrollable factors that affect the water quality $\mathbf{y}_i$ in basin i, and the random coefficients associated to the x-variables in (19.10) determine the entries of the random matrix $\mathbf{T}$ through the transformation $(D + d\mathbf{w}) \Delta \mathbf{L}$. The decision variables have been reindexed as an n-vector $(x_1, \ldots, x_n)$, each x_j corresponding to a specific control project affecting the load in some basin i. $\mathbf{T}$ is thus an $m \times n$-matrix and $\mathbf{h}$ is an m-vector. We also write

$$y(x, \omega) = T(\omega)x - h(\omega) \tag{19.16}$$

for the preceding equation. The notation $y(x, \omega)$ is used to stress the dependence of the water quality indicators $y_i(x, \omega)_{i=1}^m$ on the decision variables x and on the existing (stochastic) environmental conditions ω that determine the entries of T and h.

The distribution function $G_y(x, \cdot)$ of the random vector $y(x, \cdot)$ also depends on the choice of the control measures. We could view our objective as finding x' that satisfies the constraints and such that for every other feasible x

$$G_y(x', \cdot) \geq G_y(x, \cdot), \tag{19.17}$$

i.e., such that for all $z \in R^m$

$$\text{prob}[y(x', \cdot) < z] \geq \text{prob}[y(x, \cdot) < z].$$

If such an x' existed, it would, of course, be the "absolute" optimal solution, since it guarantees the best water quality whatever the actual realization of the random environment. There always exists such a solution if there are no budgetary limitations: simply build all possible projects to their physical upper bounds! It is precisely because there are budgetary limitations that we are led to choose a restricted number of treatment plants and/or prereservoirs. Unless the problem is very unusual there will be no choice of investment program that will dominate all other feasible programs in terms of the preference ordering suggested by (19.17).

We are thus forced to examine more carefully the objectives we want to achieve. We could, somewhat unreasonably, see the goal as bringing the water quality indicator to a near-zero level in all basins. This would ignore the individual characteristics of each basin as well as the user-oriented criteria—such as, for example, recreational versus agricultural. A more sensible approach is to choose the control measures to achieve certain desirable trophic states basin by basin. Let

$$\gamma_i, \quad i = 1, \ldots, m,$$

be water quality goals expressed in terms of the indicator, $(\text{Chl} - \text{a})_{\max}$, each γ_i corresponding to the particular use of basin i. The sensitivity of the solution to these fixed levels γ_i would have to be a part of the overall analysis of the system. We are interested in the quantities

$$[y_i(x, w) - \gamma_i]_+ \qquad \text{for} \quad i = 1, \ldots, m$$

that measure the deviations between realized water quality and the fixed goals γ_i, where $[z]_+$ denotes the nonnegative part of z:

$$[z]_+ = \begin{cases} 0 & \text{if } z < 0, \\ z & \text{if } z \geq 0. \end{cases}$$

The vector

$$[y_i(x, \cdot) - \gamma_i]_+, \qquad i = 1, \ldots, m,$$

is random with distribution function $G(x, \cdot)$ defined on R^m. The problem is again to choose among all feasible control measures a program x' that generates the "best" distribution $G(x', \cdot)$ by which one could again mean

$$G(x', z) \geq G(x, z)$$

for all $z \in R^m$.

Such an x' exists only in very unusual circumstances. We must find a way to compare the distribution functions that takes into account their particular characteristics but leads to a measure that can be expressed in terms of a scalar functional.

19.4.1 Reliability criteria

A first possibility would be to introduce a pure *reliability criterion*, i.e., to fix in consultation with the decision maker certain reliability coefficients to guide the choice of an investment program. More specifically, we would fix $0 < \alpha \leq 1$, so that among all feasible x we should restrict ourselves to those satisfying

$$\text{prob}[y(x, \cdot) < \gamma] \geq \alpha. \tag{19.18}$$

Or preferably, if we take into account the fact that each basin should be dealt with separately, we would fix the reliability coefficients $(\alpha_i)_{i=1}^m$ and impose the constraints

$$\text{prob}[y_i(x, \cdot) < \gamma_i] \geq \alpha_i, \quad i = 1, \ldots, m, \tag{19.19}$$

where the scalars α or $(\alpha_i)_{i=1}^m$ would be chosen sufficiently large so that we would observe the unacceptable concentration level only on rare occasions. In terms of the distribution function G these constraints become

$$G(x, 0) \geq \alpha \tag{19.20}$$

for (19.18) and

$$G_i(x, 0) \geq \alpha_i \qquad \text{for } i = 1, \ldots, m \tag{19.21}$$

for (19.19), where $G_i(x, \cdot)$ are the marginal distributions of the random variables $[y_i(x, \cdot) - \gamma_i]_+$. These are *probabilistic* (or *chance*) *constraints*. One refers to (19.18) as a *joint* probabilistic constraint. These model simple accept/reject criteria: namely, if $G(x, \cdot)$ is either larger than or equal to α, or for each $i = 1, \ldots, m$, $G_i(x, \cdot)$ is larger than or equal to α_i, then the investment program x is acceptable. This means that we "compare" the possible distributions $\{G(x, \cdot), x \text{ feasible}\}$ at the single point α.

Assuming we opt for the more natural separable version of the probabilistic constraints (19.19), we would rely on the following model for the policy analysis:

$$\begin{aligned}
&\text{find } x \in R^n \text{ such that} \\
&r_j^- \leq x_j \leq r_j^+, && j = 1, \ldots, n, \\
&\sum_{j=1}^n \alpha_{ij} x_j \leq b_i, && i = 1, \ldots, m, \\
&\text{prob}\left[\sum_{j=1}^n t_{ij}(\omega) x_j - h_i(\omega) < \gamma_i\right] \geq \alpha_i, && i = 1, \ldots, m, \\
&\text{and } z = \sum_{j=1}^n c_j(x_j) \text{ is minimized,}
\end{aligned} \tag{19.22}$$

where as before the vectors r^- and r^+ are upper and lower bounds on x, the inequalities $\sum_{j=1}^n \alpha_{ij} x_j \leq b_i$ describe the technological constraints, and for every j

$$c_j : R \to R_+$$

is the cost function associated to project j; see (19.14). The overall objective would thus be to find the smallest possible budget that would guarantee meeting the present goals γ_i at least a portion α_i of the time.

We do not pursue this approach because it does not allow us to distinguish between situations where we almost meet the goals γ_i and those that generate "catastrophic" situations, i.e., when some of the values of the $(y_i(x, \omega))_{i=1}^m$ would exceed by far $(\gamma_i)_{i=1}^m$. For a eutrophication model this is a serious shortcoming.

Let us also point out that probabilistic contraints involving affine functions with random coefficients are difficult to manage. We have only very limited knowledge about such constraints, and then only if the random coefficients $((t_{ij}(\cdot))_{j=1}^n, h_i(\cdot))$ are jointly normally distributed. (See [14, section 1] for a survey of the available results and the relevant references.) Since in environmental problems the coefficients are generally not normally distributed random variables, we could not even use the few results that are available, except

possibly by replacing the probabilistic constraints by approximations using Chebyshev's inequality, as suggested by Sinha; cf. [14, Proposition 1.26].

19.4.2 Recourse formulation

A second possibility is to recognize the fact that one should distinguish between situations that barely violate the desired water quality or levels $(\gamma_i)_{i=1}^m$ and those that deviate substantially from these norms. This suggests a formulation of our objective in terms of a penalization that would take into account the observed values of $[y_i(x,\omega) - \gamma_i]_+$ for $i = 1, \ldots, m$. We expect such a function

$$\Psi : \mathrm{R}^m \to \mathrm{R}$$

to have the following properties:

1. Ψ is nonnegative.
2. $\Psi(z) = 0$ if $z_i \leq 0$ for $i = 1, \ldots, m$.
3. Ψ is separable, i.e., $\Psi(z) = \sum_{i=1}^m \Psi_i(z_i)$.

This last property comes from the fact that the objectives for each basin are or may be different and there are essentially no joint rewards to be accrued from having given concentration levels in neighboring basins, the interconnections between the basins being already modeled through (19.1). A more sophisticated model would still work with separate penalty functions $(\Psi_i(z_i))_{i=1}^m$ but instead of simply summing these penalties would treat them as multiple objectives. A solution to such a problem would eventually assign specific weights to each basis, making it equivalent to an optimization problem with a single objective function. We assume that these weighting factors have been made available to or have been discovered by the model builder and have been incorporated in the functions Ψ_i themselves; however, the methodology developed here applies equally well to a multiple objective version of the model. In addition to 1–3, we would expect the following properties: for $i = 1, \ldots, m$,

4. Ψ_i is differentiable, with derivative Ψ_i'.
5. Ψ_i' is monotone increasing, i.e., Ψ_i is convex.
6. $\Psi_i'(z_i) > 0$ whenever $z_i > 0$,
   - relatively small if z_i is close to 0,
   - leveling off when z_i is much larger than 0.

A couple of possibilities, both with $\Psi_i(z_i) = 0$ if $z_i \leq 0$, are

$$\Psi_i(z_i) = \beta_i z_i^2 \quad \text{if } z_i \geq 0,$$

with $\beta_i > 0$, and

$$\Psi_i(z_i) = \beta_i(e^{z_i} - z_i - 1) \quad \text{if } z_i \geq 0,$$

also with $\beta_i > 0$.

There is a wide variety of functions that have the desired properties. What is at stake here is the creation of a (negative) utility function that measures the socioeconomic consequences of the deterioration of the environment. We found that the following class of functions provided a flexible tool for the analysis of these factors. Let $\Theta : \mathrm{R} \to \mathrm{R}_+$ be defined by

$$\Theta(\tau) := \begin{cases} 0 & \text{if } \tau \leq 0, \\ 1/2\tau^2 & \text{if } 0 \leq \tau \leq 1, \\ \tau - 1/2 & \text{if } \tau \geq 1. \end{cases} \tag{19.23}$$

This is a piecewise linear-quadratic-linear function. The functions $(\Psi_i)_{i=1}^m$ are defined through

$$\Psi_i(z_i) = q_i e_i \Theta(e_i^{-1} z_i) \qquad \text{for} = 1, \ldots, m, \tag{19.24}$$

where q_i and e_i are positive quantities that allow us to scale each function Ψ_i in terms of slopes and the range of its quadratic component. By varying the parameters e_i and q_i we are able to model a wide range of preference relationships and study the stability of the solution under perturbation of these scaling parameters.

The objective is to find a program that on the average minimizes the penalties, or negative utilities, associated with exceeding the desired concentration levels. This leads us to the following formulation of the water quality management problem:

$$\begin{aligned} &\text{find } x \in R^n \text{ such that} \\ &r_j^- \leq x_j \leq r_j^+, && j = 1, \ldots, n, \\ &\sum_{j=1}^n \alpha_{ij} x_j \leq b_i, && i = 1, \ldots, m, \\ &\sum_{j=1}^n c_j(x_j) \leq \beta, \\ &\sum_{j=1}^n t_{ij}(\omega) x_j - y_i(\omega) = h_i(\omega), && i = 1, \ldots, m, \\ &\text{and } z = E\left\{\sum_{i=1}^m q_i e_i \Theta(e_i^{-1}[y_i(\omega) - \gamma_i])\right\} \text{ is minimized,} \end{aligned} \tag{19.25}$$

where β is the available budget. This type of stochastic optimization problem is called a stochastic program with recourse: a decision x (the investment program) must be chosen before we can observe the outcome of the random events (the environment modeled here by the random quantities $t_{ij}(\omega)$, $h_i(\omega)$), at which time a recourse decision is selected so as to make up whatever discrepancies there may be; the variables y_i are just measuring the difference between $\sum_j t_{ij} x_j$ and h_i. One refers to (19.25) as a program with *simple recourse* in that the recourse decision is uniquely determined by the first-stage decision x and the values taken on by the random variables.

Observe that no attempt has been made to combine budgetary considerations and the penalty functions that measure the deviations from the desired concentration levels in a

single objective function, although there are financial considerations that may affect the choice of the coefficients q_i and e_i of the penalty terms. In our approach we handle these two criteria separately. We rely on a (discrete) parametric analysis of the solution of (19.25) as a function of β, the available budget. An essentially equivalent approach would have been to formulate (19.25) as a multiobjective program with one objective corresponding to the penalization terms and the other to the cost function.

In terms of the distribution functions $\{G(x, \cdot), x \text{ feasible}\}$, the entire "tail" of the distributions enters into the comparison, not just the value of $G(x, \cdot)$ at 0, as was the case in model (19.22) with probabilistic contraints. Indeed, the objective function can now be expressed as

$$z = \sum_{i=1}^{m} q_i e_i \int_0^{\infty} \Theta(e_i^{-1} s) dG_i(x, s).$$

19.4.3 Expected value model

A third possibility is to essentially ignore the stochastic aspects of the eutrophication model and replace the random variables that appear in the formulation of the water quality management problem by fixed quantities. This would lead us to the following *deterministic optimization problem*:

$$\begin{aligned}
&\text{find } x \in R^n \text{ such that} \\
&r_j^- \le x_j \le r_j^+, && j = 1, \ldots, n, \\
&\sum_{j=1}^{n} \alpha_{ij} x_j \le b_i, && i = 1, \ldots, m, \\
&\sum_{j=1}^{n} c_j(x_j) \le \beta, && \qquad (19.26) \\
&\sum_{j=1}^{n} \hat{t}_{ij} x_j - y_i = \hat{h}_i, && i = 1, \ldots, m, \\
&\text{and } z = \sum_{i=1}^{m} q_i e_i \Theta \left(e_i^{-1} [y_i - \gamma_i] \right) \text{ is minimized.}
\end{aligned}$$

The choice of the parameters $\hat{t}_{ij}$ and $\hat{h}_i$ is left to the model builder. One possibility is to choose

$$\begin{aligned}
\hat{t}_{ij} &= \bar{t}_{ij} = E\left[t_{ij}(\omega)\right], \\
\hat{h}_i &= \bar{h}_i = E\left[h_i(\omega)\right],
\end{aligned}$$

i.e., replace the random quantities by their expectations. Without accepting the solution of (19.26), we could always use it as part of an initialization scheme for solving the stochastic optimization problem (19.25), and this is actually how the algorithm proceeds; see section 19.5.

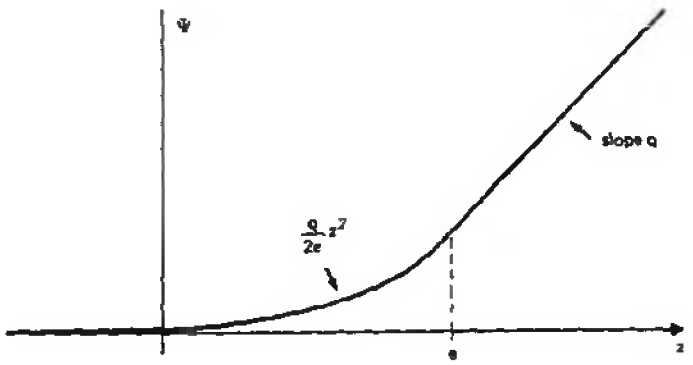


Figure 19.2. *Criteria functions.*

19.4.4 Optimal reliability levels

A fourth model in which reliability considerations again occupy a central role, but in which the shapes of the distribution functions $\{G_i(x,\cdot)\}_{i=1}^m$ play a much more important role than just their values at one point, allows for variable concentration levels. Again let $(\alpha_i)_{i=1}^m$ be scalars that correspond to desired reliability level. The objective is to find an investment program x such that

$$\text{prob}\,[y_i(x,\cdot) < v_i] \geq \alpha_i \quad \text{for } i = 1,\ldots,m, \tag{19.27}$$

but now the $(v_i)_{i=1}^m$ are also decision variables that we would like to choose as low as possible. There are a variety of ways of measuring "as low as possible," for example, by minimizing

$$\sum_{i=1}^m q_i[v_i - \gamma_i]_+, \tag{19.28}$$

where q_i are nonnegative scalars that assign different importance to meeting the desired water quality goals $(\gamma_i)_{i=1}^m$ in the various basins, or by minimizing

$$\max_i v_i, \tag{19.29}$$

i.e., by bringing the overall concentration level as far down as possible (at least a certain portion of the time determined by the α_i'), or by minimizing as in model (19.25) the function

$$\sum_{i=1}^m q_i e_i \Theta\left[e_i^{-1}(v_i - \gamma_i)\right], \tag{19.30}$$

which penalizes the deviations from γ_i in a nonlinear manner (cf. Figure 19.2), or still to handle the minimization of the $(v_i)_{i=1}^m$ as a multiple objective optimization problem, each coordinate of v corresponding to an objective that we seek to minimize.

We shall here formulate our optimization problem in terms of the objective (19.28) but, of course, any of the other variants could or should be considered. The optimization

problem again involves probabilistic constraints but its structure now resembles the stochastic program with recourse (19.25) much more than it does the first model (19.22) involving probabilistic constraints. We obtain

$$
\begin{aligned}
&\text{find } x \in R^n \text{ such that} \\
&r_j^- \le x_j \le r_j^+, && j = 1, \dots, n, \\
&\sum_{j=1}^{n} \alpha_{ij} x_j \le b_i, && i = 1, \dots, m, \\
&\sum_{j=1}^{n} c_j (x_j \le \beta), && (19.31) \\
&\text{prob}\left[\sum_{j=1}^{n} t_{ij}(\omega) x_j - h_i(\omega) - v_i < 0 \right] \ge \alpha_i, && i = 1, \dots, m, \\
&\text{and } z = \sum_{i=1}^{m} q_i [v_i - \gamma_i]_+ \text{ is minimized.}
\end{aligned}
$$

At this point it may be worthwhile to observe that (19.28) is just a limit case of (19.30). Recall that the range over which $q_i e_i \Theta(e_i^{-1}(\cdot - \gamma_i))$ is quadratic is $[0, e_i]$; cf. Figure 19.2. If we shrink this interval to one point, we are left with the piecewise linear function $q_i[\cdot - \gamma_i]_+$.

As for our earlier models, we should study the solution as a parametric function of β, the available budget. However, solving (19.31) presents all the technical challenges mentioned in connection with the first model (19.22) involving probabilistic constraints. The presence of the $(v_i)_{i=1}^m$ has in no way simplified the problem, and in fact we do not know of any direct method for solving (19.31). One possibility is to find an approximation of (19.31) that could be handled by available linear or nonlinear programming techniques. We return to this in the next section.

19.4.5 Expectation-variance formulation

A fifth possibility is to deploy a model as in [8] that is based on expectation-variance considerations for the water quality indicators. The justification of the model relies on the validity of certain approximations and thus in some situations one should accept the solution with some circumspection. However, as shall be argued, its solution always points in the right direction and is usually far superior to that obtained by solving a "deterministic" problem such as (19.26). In the Lake Balaton case study the results for both this expectation-variance model and the stochastic programming model (19.25) lead to remarkably similar investment decisions, as shown by the analysis of the results in section 19.6.

As a starting point for the construction of this model, consider the recourse objective function

$$
\sum_{i=1}^{m} q_i E\left\{ (y_i(x, \cdot) - \gamma_i)_+^2 \right\}. \tag{19.32}
$$

The objective being quadratic in the area of interest, and the distribution functions $G_i(x, \cdot)$ of the $y_i(x, \cdot)$ not being too far from normal, one should be able to recapture the essence

of its effect on the decision process by considering just expectations and variances. This observation and the "soft" character of the management problem (which in any case means that there is a large degree of flexibility in the choice of the objective) suggest that we could substitute

$$\sum_{i=1}^{m} q_i \left[E\{y_i(x,\cdot) - \bar{y}_{0i}\} + \Theta\sigma(y_i(x,\cdot) - \bar{y}_{0i}) \right] \tag{19.33}$$

for (19.32), where Θ is a positive scalar (usually between 1 and 2.5), $\bar{y}_{0i} = E\{y_{0i}\}$ is the expected nominal state of basin i, and σ denotes standard deviation,

$$\sigma\left[y_i(x,\cdot) - \bar{y}_{0i} \right] = E\left\{ \left[y_i(x,\cdot) - E\{y_i(x,\cdot)\} \right]^2 \right\}^{1/2}. \tag{19.34}$$

Since for each $i = 1, \ldots, m$ the y_i are affine (linear plus a constant term) with respect to x, the expression for

$$E\{y_i(x,\cdot) - \bar{y}_{0i}\} = \sum_{j=1}^{n} \mu_{ij} x_j + \mu_{i0}$$

as a function of x is easy to obtain from (19.1) and (19.10). The μ_{ij} are the expectations of the coefficients of x, and μ_{i0} are the expectation of the constant term. Unfortunately the same does not hold for the standard deviation $\sigma(y_i(x,\cdot) - \bar{y}_{0i})$. Equations (19.1) and (19.10) suggest that

$$\sigma(y_i(x,\cdot) - \bar{y}_{0i}) \sim \left(\sum_j \sigma_{ij}^2 x_j^2 \right)^{1/2}, \tag{19.35}$$

where σ_{ij} is the part of the standard deviation that can be influenced by the decision variable x_j, for example, the standard deviation of the tributary load $L(\omega)_D$. Cross terms are for all practical purposes irrelevant in this situation since the total load in basin i is essentially the result of a sum of the loads generated by various sources that are independently controlled. This justifies using

$$\sum_{i=1}^{m} q_i \left[\left(\sum_{j=1}^{n} \mu_{ij} x_j \right) + \Theta \left(\sum_{j=1}^{n} \sigma_{ij}^2 x_j^2 \right)^{1/2} \right] \tag{19.36}$$

instead of (19.33) as an objective for the optimization problem. This function is convex and differentiable on R_+^n except at $x = 0$, and conceivably one could use a nonlinear

programming package to solve the optimization problem

$$\begin{aligned}
&\text{find } x \in R^n \text{ such that} \\
&r_j^- \le x_j \le r_j^+, && j = 1, \ldots, n, \\
&\sum_{j=1}^{n} \alpha_{ij} x_j \le b_i, && i = 1, \ldots, m, \\
&\sum_{j=1}^{n} c_j(x_j \le \beta, \\
&\text{and } z = \sum_{i=1}^{m} q_i \left[\sum_{j=1}^{n} \mu_{ij} x_j + \Theta \left(\sum_{j=1}^{n} \sigma_{ij}^2 x_j^2 \right)^{1/2} \right] \text{ is minimized.}
\end{aligned} \tag{19.37}$$

Assuming that the cost functions have been linearized, i.e., with each c_j piecewise linear, the MINOS package [5] would be an excellent choice since the solution is bounded away from 0.

We can go one step further in simplifying the problem to be solved, namely, by replacing the term

$$\left(\sum_{j=1}^{n} \sigma_{ij}^2 x_j^2 \right)^{1/2}$$

in the objective by the linear (inner) approximation

$$\sum_{j=1}^{n} \sigma_{ij} x_j .$$

On each axis of R_+^n, no error is introduced by relying on this linear approximation; otherwise we are overestimating the effect a certain combination of the x_j will have on the variance of the concentration levels. Thus, at a given budget level we shall have a tendency to start projects that affect the variance more strongly if we use the linear approximation, and this is actually what we observed in practice (see section 19.6). Assuming the cost functions c_j are piecewise linear, we have to solve the *linear* program

$$\begin{aligned}
&\text{find } x \in R^n \text{ such that} \\
&r_j^- \le x_j \le r_j^+, && j = 1, \ldots, n, \\
&\sum_{j=1}^{n} \alpha_{ij} x_j \le b_i, && i = 1, \ldots, m, \\
&\sum_{j=1}^{n} c_j(x_j) \le \beta, \\
&\text{and } z = \sum_{i=1}^{m} q_i \sum_{j=1}^{n} (\mu_{ij} + \Theta \sigma_{ij}) x_j \text{ is minimized.}
\end{aligned} \tag{19.38}$$

We refer to this problem as the *linearized expectation-variance model* (see also [8] and [10]).

For the sake of illustration, let us consider the ith basin of Figure 19.1 and suppose that there is no mass exchange with neighboring basins. To obtain a linear form in the x_j, we proceed as indicated in (19.9). To derive the remaining term in the objective of (19.38) we only need to consider the controllable portion of the variance of $\mathbf{y}_i(x) - \bar{y}_{0i}$. We rewrite (19.1) as

$$\mathbf{y}_i(x) - \bar{y}_{0i} = \mathbf{w}_i - (d_{ii} + d_i \mathbf{w}_i)\Delta \mathbf{L}_i.$$

Let us write

$$\Delta \mathbf{y}_i = (d_{ii} + d_i \mathbf{w}_i)\Delta \mathbf{L}_i.$$

We may assume that $\mathbf{w}_i$ and $\Delta \mathbf{L}_i$ are independent, from which we obtain

$$\sigma^2(\Delta \mathbf{y}_i) = d_{ii}^2 \sigma^2(\Delta \mathbf{L}_i) + d_i^2 \sigma^2(\mathbf{w}_i)\left[\sigma^2(\Delta \mathbf{L}_i) + E^2\{\Delta \mathbf{L}_i\}\right]. \tag{19.39}$$

From (19.3), (19.6), (19.7), and (19.8) we have

$$\sigma^2(\Delta \mathbf{L}_i) = \frac{1}{V_i^2}\left[\delta^2(x_p - 1)^2\sigma^2(\mathbf{L}_T - \mathbf{L}_D) + (x_D - 1)^2\sigma^2(\mathbf{L}_D)\right], \tag{19.40}$$

where we have made the plausible assumption that the measurement uncertainties in the tributary dissolved load $\mathbf{L}_D$ and undissolved particulate load $\mathbf{L}_T - \mathbf{L}_D$ are independent. This would lead to an expression for $\sigma(\Delta \mathbf{y}_i)$ that would be nonlinear in the x variables. To avoid the nonlinearities we specify $\sigma_a(\Delta \mathbf{y}_i)$ and $\sigma_a(\Delta \mathbf{L}_i)$ as the linear combination of the additive terms in (19.39) and (19.40),

$$\sigma_a(\Delta \mathbf{y}_i) := d_{ii}\sigma_a(\Delta \mathbf{L}_i) + d_i\sigma(\mathbf{w}_i)\left[\sigma_a(\Delta \mathbf{L}_i) + E\{\Delta \mathbf{L}_i\}\right] \tag{19.41}$$

and

$$\sigma_a(\Delta \mathbf{L}_i) := \frac{1}{V_i}\left[\delta(x_P - 1)\sigma(\mathbf{L}_T - \mathbf{L}_D) + (x_D - 1)\sigma(\mathbf{L}_D)\right]. \tag{19.42}$$

In (19.39) and (19.40) all the coefficients of x are positive and the behavior of the "new" σ_a is similar to the standard deviations as defined through (19.39) and (19.40). Substituting the terms involving x in (19.42) into (19.41) yields

$$\begin{aligned}\sigma_{ai} = V_i^{-1}\Big\{ & x_P\left[d_{ii} + d_i\sigma(\mathbf{w}_i)\delta\left[\sigma(\mathbf{L}_T) - \sigma(\mathbf{L}_D)\right] + d_i\sigma(\mathbf{w}_i)\delta\left[E\{\mathbf{L}_T\} - E\{\mathbf{L}_D\}\right]\right] \\ & + x_D\left[\left[d_{ii} + d_i\sigma(\mathbf{w}_i)\sigma(\mathbf{L}_D)\right] + d_i\sigma(\mathbf{w}_i)\left[E\{\mathbf{L}_D\} - (1 - r_t)x_{SN}L_{SN}\right]\right] \\ & + x_{SN}d_i\sigma(\mathbf{w}_i)L_{SN} + x_s d_i\sigma(\mathbf{w}_i)L_S\Big\}.\end{aligned}$$

Collecting terms, we obtain the coefficients σ_{ij} that appear in the objective of the linear program (19.38). (A more detailed, but similar, derivation also yields the expression for the standard derivation when there is mass exchange between neighboring basins.)

The arguments that we have used to justify the expectation-variance model are mostly of a heuristic nature, in that they rely on a good understanding of the problem at hand and engineering intuition. In the formulation of the models of this section, the objective has usually been formulated in terms of finding control measures such that the observed

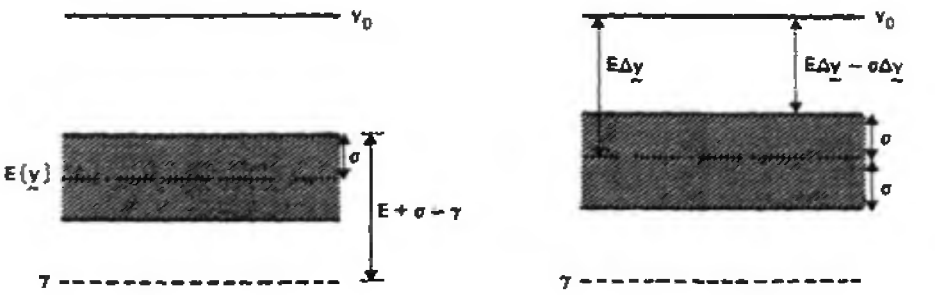


Figure 19.3. *Objective of expectation-variance model.*

concentration levels (water quality indicators) are not too far from preset goals (given trophic states). If by "not too far" we mean that

$$\left\{E\left[\mathbf{y}_i(x)\right]+\sigma\left[\mathbf{y}_i(x)\right]\right\}-\gamma_i \tag{19.43}$$

should be as small as possible, we could also reformulate the problem in terms of the nominal concentration levels. Instead of minimizing (19.43), we could maximize

$$E\left[\Delta\mathbf{y}_i(x)\right]-\sigma\left[\Delta\mathbf{y}_i(x)\right], \tag{19.44}$$

and this should give about the same results. This is actually the motivation behind the formulation of (19.38); see Figure 19.3.

There is, however, another approach that does not rely so extensively on heuristic considerations, which leads us to the model (19.37), i.e., the nonlinear version of the expectation-variance model. The fourth model, described in section 19.4.4, which integrates both reliability considerations and penalties for fixing the reliability levels, led us to the nonlinear program

find $x \in R^n$ such that

$$r_j^- \le x_j \le r_j^+, \qquad j=1,\ldots,n,$$

$$\sum_{j=1}^{n} \alpha_{ij}x_j \le b_i, \qquad i=1,\ldots,m,$$

$$\sum_{j=1}^{n} c_j(x_j) \le \beta,$$

$$\text{prob}\left[\sum_{j=1}^{n} t_{ij}(\omega)x_j - h_i(\omega) - y_i < 0\right] \ge \alpha_i, \quad i=1,\ldots,m,$$

and $z=\sum_{i=1}^{m} q_i[y_i-\gamma_i]_+$ is minimized.

Because these probabilistic constraints are very difficult to handle, we may consider finding an approximate solution by replacing the probabilistic constraints by

$$(1-\alpha_1)^{-2}\left[\sum_{j=0}^{n}\sum_{k=0}^{n}\sigma_{ijk}x_jx_k\right]^{1/2}+\sum_{j=0}^{n}\mu_{ij}x_j\leq y_i, \tag{19.45}$$

where

$$t_{0i}(\cdot)=-h_i(\cdot),$$

and for $j=0,\ldots,n$ and $k=0,\ldots,n$

$$\mu_{ij}:=E\left\{t_{ij}(\omega)\right\},$$

$$\sigma_{ijk}:=\operatorname{cov}\left[t_{ij}(\cdot),t_{ik}(\cdot)\right].$$

If the random variables $\{t_{ij}(\cdot)_{j=0}^{j}\}$ are jointly normal, then the restrictions generated by the deterministic constraints (19.45) are exactly the same as those imposed by the probabilistic constraints, but, in general, they are more restrictive (cf. [14, Propositions 1.25 and 1.26]). Without going into details, we can see that (19.45) is obtained by applying Chebyshev's inequality and this, in general, determines an upper bound for the probabilistic event

$$\left\{\omega|\sum_{j=0}^{n}t_{ij}(\omega)x_j\geq 0\right\}.$$

Thus if we can justify a near-normal behavior for the random variable (for fixed x)

$$\sum_{j=1}^{n}t_{ij}(\omega)x_j-h_i(\omega)=:y_i(x,\omega),$$

we can use the constraints (19.45) instead of the probabilistic constraints to obtain an approximate solution of (19.31). In this setting, "near normality" of the $\mathbf{y}_i(x)$ is a much more natural, and weaker, assumption than normality of the $t_{ij}(\omega)$. Assuming that we proceed in this fashion, we obtain the nonlinear program

find $x\in R^n$ such that

$$\begin{aligned}
&r_j^-\leq x_j\leq r_j^+, && j=0,\ldots,n,\\
&\sum_{j=1}^{r}\alpha_{ij}x_j\leq b_i, && i=1,\ldots,m,\\
&\sum_{j=1}^{n}c_j(x_j)\leq\beta, && (19.46)\\
&\text{and}
\end{aligned}$$

$$z = \sum_{i=1}^{m} q_i \left[\sum_{j=0}^{n} \mu_{ij} x_j + (1-\alpha_i)^{-2} \left(\sum_{j=0}^{n} \sum_{k=0}^{n} \sigma_{ijk} x_j x_k \right)^{1/2} - \gamma_i \right]_+$$

is minimized.

We have eliminated the variables $(y_i,\ i = 1, \ldots, m)$ from the formulation of the problem by using the fact that the optimal y_i^* can always be chosen so that (19.45) is satisfied with equality. Moreover, if the desired concentration levels γ_i are low enough, then we know that the optimal solution will always have $y_i^* > \gamma_i$ and thus we can rewrite (19.46) as follows:

find $x \in R^n$ such that

$$r_j^- \leq x_j \leq r_j^+, \qquad j = 0, \ldots, n,$$

$$\sum_{j=1}^{r} \alpha_{ij} x_j \leq b_i, \qquad i = 1, \ldots, m,$$

$$\sum_{j=1}^{n} c_j(x_j) \leq \beta, \tag{19.47}$$

and

$$z = \sum_{i=1}^{m} q_i \left[\sum_{j=0}^{n} \mu_{ij} x_j + (1-\alpha_i)^{-2} \left(\sum_{j=0}^{n} \sum_{k=0}^{n} \sigma_{ijk} x_j x_k \right)^{1/2} - \gamma_i \right]$$

is minimized. (19.48)

The objective of this optimization problem is sublinear, i.e., convex and positively homogeneous. Assuming that the cost functions c_j are linear, or more realistically have been linearized (19.14), we are thus confronted with a nearly linear program that we could solve by specially designed subroutines (nondifferentiability at 0) or by a linearization scheme that would allow us to use linear programming packages. Observe that the nonlinear program (19.47) is exactly of the same type as (19.37) if we make the following adjustments:

1. In the objective of (19.47) replace the covariance term $\sum_{j=0}^{n} \sum_{k=0}^{n} \sigma_{ijk} x_k x_j$ by the sum of the variances $\sum_{j=1}^{n} \sigma_{ijk} x_j^2$.

2. If for all $i = 1, \ldots, m$, the α_i are the same set, $\Theta = (1-\alpha_i)^{-1}$; otherwise we replace Θ by $\Theta_i = (1-\alpha_i)^{-2}$ in (19.37).

To justify 1, we appeal to (19.35).

In the derivation that led us from (19.31) to (19.47), we stressed the fact that the solution of (19.47) and thus equivalently of (19.37) would be feasible for the original program (19.31) and that, in fact, it would more than meet the probabilistic constraints specified in (19.31). The further linearization of the objective bringing us from (19.37) to (19.38) overstresses (possibly only slightly) the role that the variance will play in meeting the prescribed reliability levels. In terms of model (19.31), we can thus view the solution

of (19.38) as a conservative solution that overestimates the importance to be given to the stochastic aspects of the problem. In that sense, the solution of (19.38), especially in comparison to that of the deterministic problem (19.26), always indicates how we should adjust the decisions so as to take into account the stochastic features of the problem.

In our analysis (see section 19.6), we have used the linear programming version (19.38) of this expectation-variance model; the wide availability of reliable linear programming packages makes it easy to implement, and thus an attractive approach, provided one keeps in mind the reservations expressed earlier.

19.5 Solving the stochastic model

We briefly outline here the method used to solve the full stochastic version of the eutrophication management model (19.25). We recognize it as a stochastic program with quadratic simple recourse, with stochastic technology matrix T and stochastic right-hand-side h. When only h is stochastic and the objective function is piecewise linear, efficient procedures are available [13]. But to deal with this class of problems new techniques were required.

A new procedure called the Lagrange finite generation method was developed [6] that exploits the properties of the dual associated to problem (19.25). Each iteration ν of the algorithm solves the dual over a convex hull spanned by a given basis of dual solutions. This turns out to be a certain finite-dimensional quadratic program, which produces a feasible primal solution x^ν. The basis of dual solutions is then extended by adding a new member calculated from the newly obtained primal solution. The sequence of primal solutions converges at a known rate to an optimal solution of the original stochastic program.

An experimental version of this algorithm was implemented at IIASA by King; see [4]. The procedure is started by solving the deterministic problem (19.26) with expected values for the stochastic parameters. To solve the dual quadratic program we used MINOS [5]. The distribution used to calculate the dual solutions is derived by sampling from the stochastic model (19.10).

For a number of reasons (including numerical stability considerations) it is recommended to start with a relatively small sample, increasing its size only for verification purposes. We have observed that a relatively small sample (about 50) will give surprisingly good results. Asymptotic error bounds for solutions under this type of sampling approximation have been derived in [3].

19.6 Application to Lake Balaton

Lake Balaton (Figure 19.4), one of the largest shallow lakes of the world, which is also the center of the most important recreational areas in Hungary, has recently exhibited the unfavorable signs of artificial eutrophication. An impression of the major features of the lake-region system, the main processes and activities, the underlying research, data availability, and control alternatives can be gained from Figure 19.4 and Table 19.1. (For details, see Somlyódy [8, 9] and Somlyódy and van Straten [10].) Four basins of different water quality can be distinguished in the lake, determined by the increasing volumetric nutrient load from east to west. The absolute loads are roughly equal for the four basins, but the

biologically available load (BAP) is about 10 times higher in Basin I than in Basin IV (see Table 19.1, line 6). This is due to the asymmetric geometry of the system: the smallest western basin drains half the total watershed, while only 5% of the catchment area belongs to the larger basin (Table 19.1, lines 2 and 3).

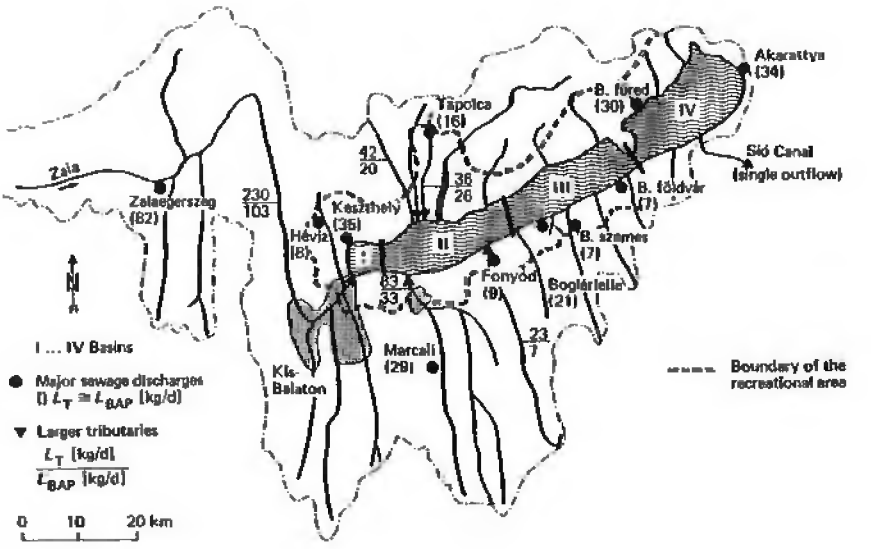


Figure 19.4. *Lake Balaton: major nutrient sources and control options.*

Based on observations for the period 1971–1982, the average deterioration of water quality of the entire lake is about 20% (in terms of Chl-a). According to the OECD classification, the western part of the lake is in a hypertrophic stage, while the eastern portion of it is in a eutrophic stage (Table 19.1, line 7).

The lakes' total P load $\mathbf{L}_T$ is on average 315 t/year (the BAP load is 170 t/year) but depending on the hydrologic regime, it can reach 550 t/year. Fifty-three percent of total load is carried by tributaries, 30% of which is of sewage origin (see, e.g., the largest city of the region, Zalaegerszeg; in Figure 19.4); 17% of total load is associated with direct sewage discharges. Atmospheric pollution is responsible for 8% of the lake's P load, and the rest is from direct runoff (urban and agricultural). Tributary load increases from east to west, while the change in the direct sewage load goes in the opoosite direction. The sewage contribution (direct and indirect loads) to $\mathbf{L}_T$ is 30%, but it contributes about 52% of the BAP $\mathbf{L}_D + \delta(\mathbf{L}_T - \mathbf{L}_D)$. The load of agricultural origin can be estimated as 47% and 33%, respectively. This suggests the importance of sewage load from the viewpoint of the short-term eutrophication control. Figure 19.4 also indicates the loads of sewage discharges and tributaries which were involved in the management optimization model. These cover about 85% of the nutrient load,[1] which we consider controllable in the short term.

Control alternatives are sewage treatment (upgraded biological treatment and P precipitation) and the establishment of prereservoirs as indicated in Figure 19.4. The Kis–Balaton reservoir system is planned for a surface area of about 75 km^2. Besides Hungarian research

[1]The rest represented by several small creeks and sewage outlets were neglected for the sake of simplicity.

Table 19.1. *Major features of Lake Balaton and its watershed.*

1.	**Basin**	**I**	**II**	**III**	**IV**	**Lake**
2.	**Watershed area** [km^2]	2750	1647	534	249	5180
3.	**Lake surface area** [km^2]	38	144	186	228	536
4.	**Volume** [$10^6 m^3$]	82	413	600	802	1907
5.	**Depth** [m]	2.3	2.9	3.2	3.7	3.2
6.	**BAP load** [$mg/m^3 d$]	1.70	0.30	0.15	0.14	0.25
7a.	**(Chl − a)$_{max}$** [mg/m^3]	75	38	28	20	(late 70s)
7b.		150	90	60	35	(1982)

8. Use of the watershed. Agriculture and intensive tourism (main season: July and August)

9. Climatic influences. No stratification; large fluctuation in temperature (up to 25–28°C); 2-month ice cover; strong wind action

10. Eutrophic status. Hypereutrophic state: P limitation until the end of the 1970s; large year-to-year fluctuation in Chl-a depending on meteorology and hydrology; 20% per-year increase in Chl-a during 1971–1982; marked longitudinal gradient

11. Sediment. Internal load is roughly equal to the external BAP load

12. Data. Long hydrological and weather records; regular water quality and load survey since 1971 and 1975, respectively

13. Research. Increasing activity in Hungary in various institutes during the last 30 years; joint study of IIASA, the Hungarian Academy of Sciences, and the Hungarian National Water Authority, 1978–1982; see Somlyódy [8, 9]

14. Models developed. Various alternative models; see Somlyódy [8, 9]

15. Methodologies. ODE and PDE models, regression analysis, Kalman filtering, time series analysis, Monte Carlo simulations, uncertainty analyses, optimization techniques

16. Measures of short-term control. P precipitation on existing treatment plants; pre-reservoirs

17. Policy making. Government decision in 1983: P control is under realization (as of 1985)

activities, the problem of Lake Balaton was studied in the framework of a four-year cooperative research project on Lake Balaton involving the International Institute for Applied Systems Analysis, IIASA (Laxenburg, Austria), the Hungarian Academy of Sciences, and the Hungarian National Water Authority [7, 8, 9]. The development of the management model to be discussed here formed a part of the case study. The results achieved were then

utilized in 1982[2] in the policy-making procedure associated with the Lake Balaton water quality problem, which was completed by a governmental decision in 1983 (Láng, 1985).

19.6.1 Specification of elements of EMOM for Lake Balaton

Nutrient load model. The nutrient load model for Lake Balaton can be derived on the basis of Figure 19.4 from relation (19.10). The tributary loads $\mathbf{L}_T$ and $\mathbf{L}_D$ are computed from regression models [10],

$$\mathbf{L} = (L_0 + \alpha_1 \mathbf{Q} + \mathbf{L}_\rho)(\xi^- + \hat{\xi}), \tag{19.49}$$

where $\mathbf{Q}$ is the stream flow rate, $\mathbf{L}_\rho$ is the residual, and the variable $\hat{\xi}$ accounts for the influence of infrequent sampling (ξ^- is the lower bound). The most detailed data set, consisting of 25 years of continuous records for $\mathbf{Q}$ and 5 years of daily observations for the loads, was available for the Zala River[3] (see Figure 19.4) draining half of the watershed and representing practically the total load of Basin I. For the Zala River, $\mathbf{L}_\rho$ was found to have a normal distribution, while $\mathbf{Q}$ was approached by a lognormal distribution. The loads of other tributaries were established on the basis of much more scarce observations. For modeling the uncertainty component of $\hat{\xi}$, first a Monte Carlo analysis was performed on the Zala River data by assuming various sampling strategies. Subsequently, the conclusions were extended to the other rivers, and the parameters of the (assumed) gamma distributions of $\hat{\xi}$ were estimated.

Control variables and cost functions. All the optimization models implemented use real-valued control variables. Integer {0, 1} variables for the two-reservoir systems (see Figure 19.4) were also used by simply fixing the variable values of 0 and 1 as part of the input. The elaboration of cost functions was based on analyzing a variety of technological process combinations (leading to different removal efficiencies) for treatment plants included in the analysis (Figure 19.4). As an example, the cost function for the largest treatment plant, Zalaegerszeg (see Figure 19.4), the capacity of which is $Q_c = 15{,}000$ m^3/d, is given in Figure 19.5.[4] Three groups of expenses are illustrated in the figure:

1. investment cost required for upgrading biological treatment;
2. investment cost of P precipitation which increases rapidly with increasing requirements; the use of piecewise linear cost functions required the introduction of three dummy variables for each treatment plant;
3. running cost.

19.6.2 Results of the expectation-variance model

To gain an impression of the character of the problem and the behavior of the solution, first we specify a basic situation (which is close to the real case) having the following features and with the following assumptions [9]:

[2]At that time only the results of the expectation-variance model were available.

[3]Its annual load estimated from daily data can be considered accurate.

[4]Roughly, US$1 is equivalent to 50 Forints (Ft).

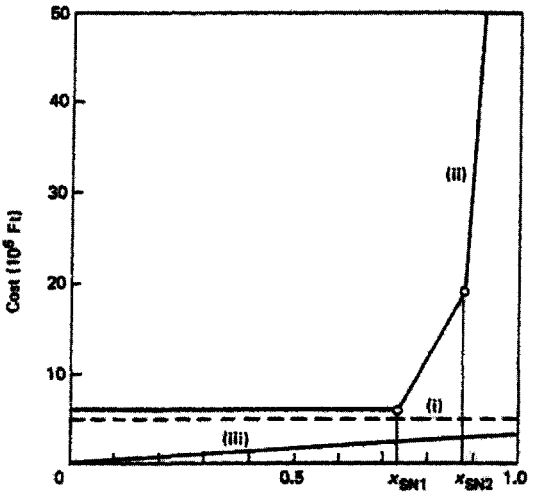


Figure 19.5. *Costs of sewage treatment (Zalaegerszeg).*

- control variables are continuous;
- no effluent standard prescription is given;
- no P retention takes place in rivers ($r_t = 0$ in (19.10));
- the capital recovery factor is equal for all the projects, $\alpha_j = \alpha = 0.1$; and
- equal weighting is adopted (see q_i and Θ in section 19.4.5).

With these assumptions optimization was performed under different budgetary conditions ($TAC \leq \beta = 0.5 - 25 \times 10^7$ ft/year). Statistical parameters (expectation, standard deviation, and extremes) of the water quality indicators gained from the Monte Carlo procedure[5] are illustrated in Figure 19.6 for the Keszthely basin as a function of the TAC.[6]

In Figure 19.7, we record the changes in the two major control variables (x_{SN1} and x_{D1}) associated with the treatment plant of Zalaegerszeg and the reed lake segment of the Kis–Balaton system (see Figure 19.4). There is a significant trade-off between these two variables. For decision-making purposes, it is important to observe that there are four ranges of possible values of β (the budget), in which the solution has different characteristics.

1. In the range of $\beta = 0.5 - 5 \times 10^7$ ft/year, it appears that sewage treatment can be intensified and tertiary treatment introduced. Expectation of the concentration levels will decrease considerably, but not the fluctuations. Under very small costs ($\sim 0.3 \times 10^7$ ft investment costs) it turns out that only the sewage of Zalaegerszeg (Figure 19.4) should be treated. Under an increasing budget, potential treatment plants are built, going from west to east.

[5]One thousand simulations were performed in each case.

[6]Running cost is about 10 times larger than TAC.

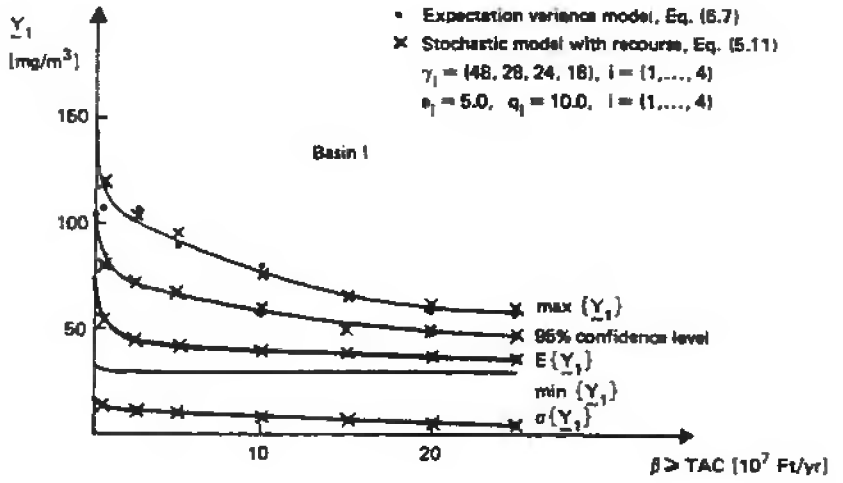


Figure 19.6. *Water quality indicator* $(\text{Chl} - \text{a})_{\max}$ *as a function of the total annual cost.*

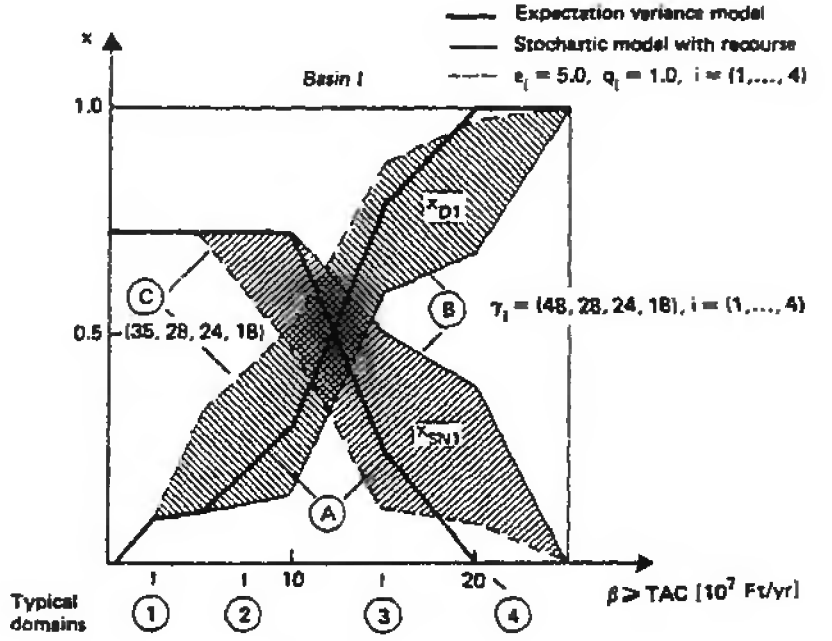


Figure 19.7. *Change of major decision variables.*

2. If β is between 5×10^7 and 10×10^7 ft/year, the effectiveness of sewage treatment cannot be increased further but reservoir systems are still too expensive.

3. At about $\beta = 15 \times 10^7$ ft/year the solution is a combination of tertiary treatment and reservoirs. Fluctuations in water quality are reduced by the latter control alternatives.

4. Finally, around $\beta = 20 \times 10^7$ ft/year, tertiary treatment is dropped in regions where reservoirs can be built. After constructing all the reservoirs, no further water quality improvement can be achieved.

Concerning the model sensitivity on major parameters, the following conclusions can be drawn (for details see [9, 10]):

1. Fixed water quality standards not reflecting the properties of the system (spatial nonuniformities) can result in a strategy far from the optimal one, since the distribution of a portion of the budget is a priori determined by the preset standard.

2. Under increasing P retention in rivers the improvement in water quality is less remarkable in the budget range $0-10 \times 10^7$ ft/year than in the basic case. The worst—nevertheless, nearly unrealistic—situation is if all the phosphorus were removed along the river and still treatment had to be performed: the budget should be partially allotted for investments having no influence on the lake's load.

3. If only deterministic effects are considered ($\Theta = 0$), reservoir projects enter the solution under much larger budget values.

4. If the capital recovery factor is smaller for reservoir projects than for sewage treatment plants (19.12), reservoir projects start to be feasible at smaller budgets. Errors in the efficiency or in costs of reservoirs cause similar shifts in the solution.

5. When selecting properly the model parameters, the combination of the absolute load reductions for the four basins is maximized by the model (as is suggested most frequently in the literature; see the Introduction). Since, however, the absolute loads alone do not reflect the spatial changes in water quality, the policy drastically differs from the optimal one.

Subsequently we give the realistic solution for the Lake Balaton management problem by using

- actual retention coefficients (ranging between 0.3 and 0.5),
- upper limits of 0.9 for the P removal rate of reservoirs, and
- fixed variables {0, 0.9} for the Kis–Balaton reservoir system.

Figure 19.8,[7] which refers again to the Keszthely bay, shows remarkable differences from Figure 19.6. First, the drastic effect of reservoirs on expectation but even stronger upon fluctuation of water quality is stressed. Reservoirs enter the solution between 15×10^7 and 17.5×10^7 ft/year total annual cost, resulting in a reduction in the mean $(\text{Chl} - \text{a})_{\max}$ concentration from about 55 to 35 mg/m^3 and in the extreme values from more than 100 to about 60 mg/m^3.

While Figure 19.6 offers several solutions for a decision maker, depending on the budget available, on the basis of Figure 19.8, only two feasible alternatives come to mind:

[7] In the figure ± standard deviation and the upper 95% confidence level are also illustrated. (The distributions are bound toward small $\mathbf{Y}_1$ concentrations, and the lower 95% confidence level values are close to the minimum.)

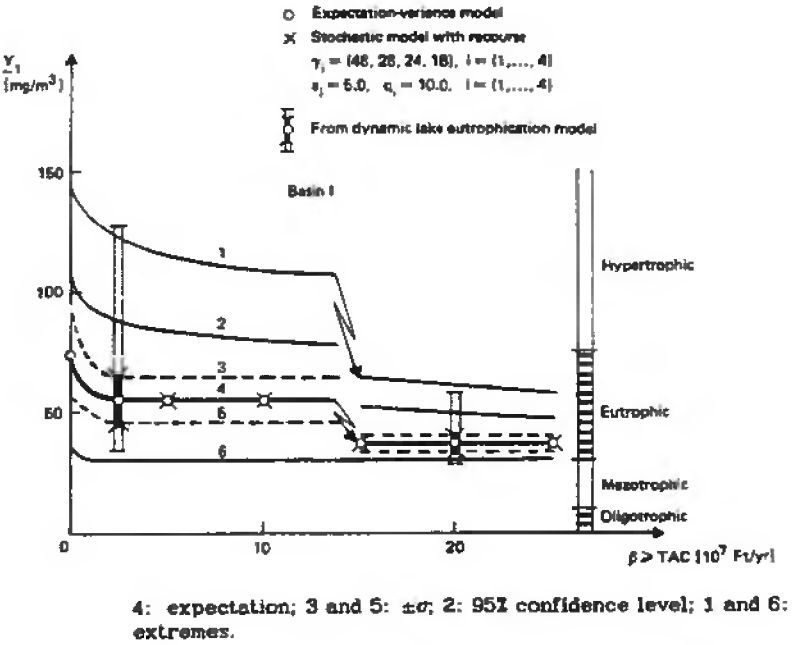


Figure 19.8. *Solutions of EMOM for Lake Balaton, Basin* I.

1. If total annual cost of about 2.5×10^7 ft/year is available, all the sewage projects can and should be realized (going from west to east). Through this alternative the expectation of $\mathbf{Y}_1 = (\text{Chl} - \text{a})_{\max}$ is reduced to about 55 mg/m^3 (tertiary treatment affects the water quality at a slightly smaller extent than in the basic case due to P retention of tributaries) but still extremes larger than 110 mg/m^3 can occur (hypertrophic domain according to the classification of OECD, 1982). Further increases in the budget (up to 10×10^7 ft/year) have no impact on water quality (under the alternatives included in the analysis).

2. For a budget around 20×10^7 ft/year given not only the Kis–Balaton, but all the reservoirs, tertiary treatment can be realized for direct sewage sources. The mean $(\text{Chl} - \text{a})_{\max}$ concentration is about 35 mg/m^3, while the maximum is about 60 mg/m^3 (eutrophic stage).

Figure 19.8 also gives the results of a detailed simulation model for two optimal solutions (TAC $= 2.5 \times 10^7$ ft and 20×10^7 ft). The agreement between the calculated concentration indicators suggests that the aggregated lake eutrophication model is quite appropriate for our present purpose.

Figure 19.9 compares the typically skewed probability density functions of two considerably different solutions ($\beta = 2.5 \times 10^7$ ft and 20×10^7 ft, respectively) for four basins, derived from Monte Carlo simulations. (The noncontrolled state is also given in this figure.) Also from this figure we can conclude that tertiary treatment is more effective than reservoirs (when both alternatives are available) for controlling the mean concentration, but

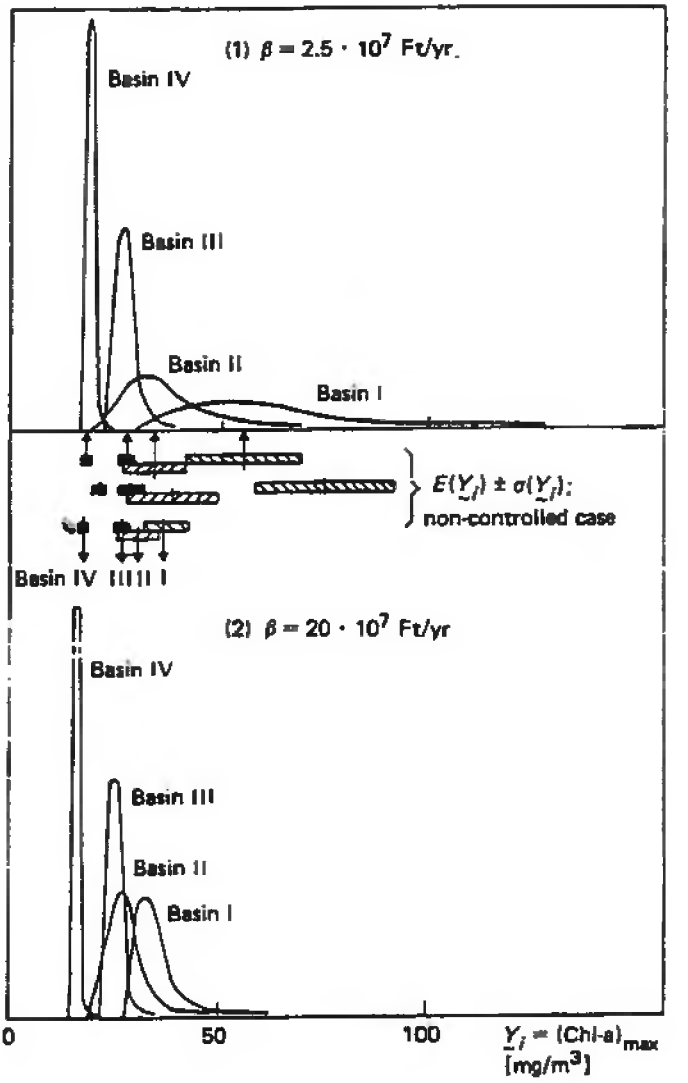


Figure 19.9. *Probability density functions for two different situations (from* 1000 *Monte Carlo simulations).*

fluctuation can be controlled by reservoirs only. In the first case ($\beta = 2.5 \times 10^7$ ft/year) Basin I remains hypertrophic, Basins II and III eutrophic, and Basin IV mesotrophic. In the second situation ($\beta = 20 \times 10^7$ ft/year) the spatial differences and stochastic changes are much smaller: Basins I, II, and III are eutrophic and Basin IV mesotrophic. (The long-term improvement of water quality is certainly larger than the short-term one discussed here.)

From what we learned through the management model, it follows that in order to realize the optimal short-term strategy of eutrophication management,

- tertiary treatment of direct sewage discharges should be introduced (from west to east);
- depending on the budget available, tertiary treatment of indirect sewage discharges of prereservoirs (again from west to east) should be realized.

For further details of the management strategy worked out for Lake Balaton and other management models not discussed in this paper, see [8, 9].

19.6.3 Results of the stochastic recourse model

As seen from Table 19.1 (line 7), the nominal state of water quality is given by the indicator vector $\mathbf{Y}_{01} = (75, 38, 28, 20)$ $(i = (1, \ldots, 4))$. Goals were specified by $\gamma_i = (48, 28, 24, 18)$ expressing the desire that Basin I should be shifted to the eutrophic and other segments to the mesotrophic state (see Figure 19.8), but without forcing a completely homogeneous water quality in the entire lake on the short term, which would be unrealistic.

The definition of these goals, however, means that the improvement intended to be achieved is quite uniform for the four basins in a relative sense: as compared to the maximal possible reduction in the water quality indicator on the short term (see Figure 19.6), we plan 50% to 60% improvement for Basins I, II, and III. Basin IV (with 20%) is the only exception, as its water quality is presently quite good, but this segment plays a secondary role from the viewpoint of the management problem.

Other parameters of the objective function (see (19.25) and Figure 19.3), e_i and q_i, were selected uniformly for the four basins: $e_i = 5$ and $q_i = 10$, $i = 1, \ldots, 4$. This corresponds, in the region $z_i \leq 5$, to a variance formulation of the objective function (being similar to (19.32)) as $q/2e = 1$. With these parameter values, the quadratic portion of the utility function is predominant in Basins II, III, and IV, while for Basin I the upper linear portion of the utility functional is also of importance.

Results of the stochastic optimization model with recourse are also illustrated in Figures 19.6–19.8, in comparison with that of the expectation-variance model. As seen from Figures 19.6–19.8, the two models produce practically the same results in terms of the water quality indicator (including also its distribution). There are minor deviations in detail. According to Figure 19.7, the expectation-variance model gives more emphasis to fluctuations in water quality and consequently to reservoir projects than the stochastic recourse model (with the parameters specified above). This is in accordance with the remarks made in section 19.4.5 that the role of the variance is overstressed in the expectation-variance model.

From this quick comparison of the performance of the two models, we may conclude that the more precise stochastic model validates the use of the expectation-variance model in the case of Lake Balaton.

For a more systematic comparison of the two models, the difference in the objective functions should be kept in mind. The stochastic model has more parameters than the expectation-variance model; in particular, the exclusion of the water quality goals γ_i from the expectation-variance model plays an important role. Figure 19.7 illustrates clearly that the prescription of the goal close to the lowest realizable value for Basin I (see Figure 19.6) leads to a stronger emphasis on reservoirs as compared to the expectation-variance model. The faster increase in x_{D1} as a function of the budget β is associated with a decrease in

x_{SN1}—as expected—in addition to smaller allocations to the other basins. Depending on the value of γ_l, the solutions lie in the shaded regions indicated in Figure 19.7. The solution to the expectation-variance model is located in the center of these regions.

As mentioned, the expectation-variance model gives more weight to variance than the stochastic recourse model. For computational justification we compared curves (A) and (B) in Figure 19.7. The rationale is that lacking water-quality goals, the expectation-variance model follows to some extent the principle of "equal relative" water quality improvements in all basins (other factors—e.g., the distribution of costs for basins—also play a role) and in this sense its solution can best be compared to solution (B) of the stochastic method.

We have the following conclusions:

1. The stochastic optimization model with recourse justified the applicability of the much simpler expectation-variance model for Lake Balaton.

2. Replacing the stochastic objective function with a deterministic version leads to a strikingly different and incorrect management strategy.

3. The most influential parameter in the stochastic model is the prescribed water-quality goals for the different basins. The inclusion of the goal in the objective function is the primary advantage in comparison with the expectation-variance model.

Bibliography

[1] M. B. Beck, *Time-Series Analysis of Zala River Nutrient Loadings*, Technical Report WP-82-116, International Institute of Applied Systems Analysis, Laxenburg, Austria, 1982.

[2] D. Haith, *Models for Analyzing Agricultural Nonpoint-Source Pollution*, Technical Report RR-82-017, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1982.

[3] A. King and R. T. Rockafellar, *Asymptotic theory for solutions in generalized m-estimation and stochastic programming*, Math. Oper. Res., 18 (1993), pp. 148–162.

[4] A. King, R. T. Rockafellar, L. Somlyódy, and R. Wets, *Lake eutrophication management: The Lake Balaton project*, in Numerical Techniques for Stochastic Optimization, Y. Ermoliev and R. J.-B. Wets, eds., Springer-Verlag, New York, 1988, pp. 435–444.

[5] B. Murtagh and M. Saunders, *Minos* 5.0 *User's Guide*, Technical Report 83-20, Systems Optimization Laboratory, Stanford University, Stanford, CA, 1983.

[6] R. T. Rockafellar and R. Wets, *A Lagrangian finite generation technique for solving linear-quadratic problems in stochastic programming*, in Stochastic Programming: 1984, A. Prékopa and R. Wets, eds., Math. Programm. Stud. 28, 1986, pp. 63–93.

[7] L. Somlyódy, *Modeling a complex environmental system: The Lake Balaton case study*, Math. Modeling, 3 (1982).

[8] L. SOMLYÓDY, *Lake eutrophication management models*, in Eutrophication of Shallow Lakes: Modeling and Management. The Lake Balaton Case Study, L. Somlyódy, S. Herodek, and J. Fischer, eds., Report CP-83-53, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1983, pp. 367ff.

[9] L. SOMLYÓDY, *A systems approach to eutrophication management with application to Lake Balaton*, Water Quality Bull., 9 (1983), 25–37.

[10] L. SOMLYÓDY AND G. VAN STRATEN, *Modeling and Managing Shallow Lake Eutrophication with Application to Lake Balaton*, Springer-Verlag, New York, Berlin, 1985.

[11] L. SOMLYÓDY AND R. WETS, *Stochastic Optimization Models for Lake Eutrophication Management*, Technical Report WP-82-116, International Institute of Applied Systems Analysis, Laxenburg, Austria, 1985.

[12] L. SOMLYODY AND R. WETS, *Stochastic optimization models for lake eutrophication management*, Oper. Res., 36 (1988), pp. 660–681.

[13] R. WETS, *Solving stochastic programs with simple recourse*, Stochastics, 10 (1983), pp. 219–242.

[14] R. WETS, 1983, *Stochastic programming: Solution techniques and approximation schemes*, in Mathematical Programming: The State-of-the-Art, A. Bachem, M. Grötschel, and B. Korte, eds., Springer-Verlag, New York, pp. 566–603.

Chapter 20

Mitigating Anthropogenic Climate Change

*Gary W. Yohe**

20.1 Introduction

Schneider [37] presents perhaps the most concise explanation of how the earth's atmosphere works to maintain an inhabitable temperature and how it might be altered by human activity. Clouds and particles in the atmosphere, together with the earth's surface, reflect roughly 30% of the incoming solar energy, but the remaining 70% of the energy is absorbed. This residual heats the surface of the earth and the atmosphere, and it is then reemitted in the infrared spectrum. An energy balance for the planet is achieved by this radiation, but only after energy trapped by clouds and greenhouse gases warms its surface. In fact, preindustrial concentrations of greenhouse gases made the earth about 33°C warmer than it would have been otherwise, and increased concentrations can further warm the planet. Since it is now understood that concentrations are increasing from human activity, the fundamental questions are clear. How much higher will temperatures climb, and how fast? How will this warming be distributed across the globe? Will some regions warm more quickly than others? Will other regions actually grow colder? How will higher temperatures affect sea levels? How might precipitation patterns change? Could warming change the frequencies and geographical distributions of extreme (weather) events? Might there be abrupt changes in climate and climate variability that could exacerbate damages? Assessments of the state of the art in our understanding of the answers to these questions can be found in [8, 11]. The take-home message in the first source was that anthropogenic emissions of greenhouse gases were causing atmospheric concentrations to rise; in the second, that those higher concentrations had already caused the climate to change.

There are, though, two sides to the economics of climate change. The first recognizes the economic costs and potential benefits that can be attributed to these physical and natural

*Department of Economics, Wesleyan University, Middletown, CT 06459 (gyohe@wesleyan.edu).

impacts, including the cost of adapting to change as well as the economic consequences that remain after such adaptation is effected. They also include the benefits that climate change might bring that would otherwise not have been forthcoming. The Intergovernmental Panel on Climate Change (IPCC) [9, 12] assessed the state of the art in our understanding of these interactions and their intersection with issues of equity and sustainable development; [48] provides some insight into how economic research into their economic consequences might proceed in ways that can accommodate enormous diversity across path-dependent and location-specific impacts. The other side recognizes that policies designed to mitigate the pace of climate change can also impose costs. In [10, 13] the IPCC assesses our understanding of these costs for a wide variety of decision tools. In all cases, though, the fundamental insight is that estimates of the economic impacts on both sides of the calculus are highly uncertain and evolving. It is, nonetheless, instructive to think about the dynamic economic trade-off between the costs of mitigation and the benefits that it might provide (in reduced damage) under the assumption that we do know what is coming and then to investigate how we might cope with the uncertainty. This paper will review some of the insights that have been derived from integrated applications of dynamic and stochastic programming techniques to these questions.

20.2 Optimization in a deterministic environment

The simplest context within which to frame an optimization problem for climate change in a deterministic environment has evolved over the past decade from early work of Nordhaus [28, 30]. Nordhaus has contributed significantly to this evolution, and the present discussion will be built on the foundations of his most recent constructions—the DICE-99 and RICE-99 models presented in [31]. This was an easy choice, because these constructions have framed the issues involved in confronting the climate problem so succinctly and because many other researchers have employed modest variants of their earlier incarnations in their work. Prime examples of this extended family include [5, 15, 32, 33, 34, 46, 47], but this list can be extended much further (see [13, Chapter 10]). This discussion will also use Nordhaus's notation in lieu of translating his models into the generic notation of stochastic programming because his notation has become standard in the climate literature.

20.2.1 Global optimization

The construction follows [30, 31]; it begins with the definition of a time-dependent objective function. Let global welfare, denoted by W, be the discounted value of future utility, i.e., let

$$W = \sum_t \frac{U[c(t), L(t)]}{R(t)} \tag{20.1a}$$

with utility function U, where $c(t)$ is per capita consumption of some aggregate product, $L(t)$ is population, $R(t)$ is a discount factor, and t indexes time. Some structure can be attached to (20.1a) by specifying a particular form to utility. The most popular choice lets

$$U[c(t), L(t)] = \{L(t)\}\{u(c(t))\} = \{L(t)\}\ln\{c(t)\}$$

so that the elasticity of the marginal utility of consumption, a measure of relative aversion to risk, is equal to unity and utility in consumption is convex; see, for example, [31, 34, 38]. Let

$$R(t) = \prod_{v=0}^{t} [1 + \rho(v)] \tag{20.1b}$$

with ρ(t) reflecting the pure rate of time preference. Impatience for future consumption may change over time. Indeed, many have argued that it should decline over time, and they reflect their argument by letting

$$\rho(t) = \rho(0)\{\exp(-g^{\rho}t)\} \tag{20.1c}$$

with $g^{\rho} > 0$. The equation of motion for population can be specified to accommodate changes in rate of growth over time by letting

$$L(t) = L(0) \exp \int_0^t g^{\text{pop}}(v)dv$$

with

$$g^{\text{pop}}(t) = g^{\text{pop}}(0) \exp\{(-\delta^{\text{pop}})t\}.$$

Setting the parameter $\delta^{\text{pop}} > 0$ captures a systematic decline in the rate of future growth of population.

Economic activity is the source of consumption goods, but it is also the source of the emissions that drive climate change as well as investment that drives future growth. One approach, adopted by, for example, DICE-99 and its "offspring," represents aggregate global economic activity in a Cobb–Douglas formulation:

$$Q(t) = \{\Omega(t)\}\{1 - b(t)\mu(t)\}\{A(\tau)K(t)^{\gamma}L(t)^{1-\gamma}\}. \tag{20.2}$$

The left-hand side of (20.2) denotes net economic output at time t with $Q(t)$. The first term on the right-hand side of (20.2) depicts economic damages attributed to climate change in $\Omega(t)$; details underlying its specification will be discussed shortly. The second term on the right-hand side reflects the economic cost of reducing emissions that are derived from the burning of fossil fuel by the industrial sector. Details of this term will also be provided in due course. The third term, finally, represents a production function in capital, denoted $K(t)$, and labor (population). The parameter γ reflects the elasticity of gross aggregate output with respect to the capital stock. The mirror elasticity for labor is given by $(1 - \gamma)$ so that production displays constant returns to scale. Neutral technological change (the ability to squeeze more output from the same levels of employment of capital and labor) is captured by the $A(t)$ term whose equations of motion are given by

$$A(t) = A(0) \exp \int_0^t g^A(v)dv$$

and

$$g^A(t) = g^A(0) \exp\{(-\delta^A)t\}.$$

The pace of technological change can, therefore, decay or accelerate over time, just like the rate of growth of population.

Net economic output is, in every year, allocated between consumption $C(t)$ and investment $I(t)$ according to

$$Q(t) = C(t) + I(t). \tag{20.3}$$

This simple accounting identity allows per capita consumption defined by

$$c(t) = \left\{\frac{C(t)}{L(t)}\right\}.$$

The equation of motion for the capital stock is

$$K(t) = (1 - \delta_K)K(\tau - 1) + I(\tau),$$

where $\delta_K > 0$ is the rate of depreciation. Moreover, $K(0)$ fixes a critical initial condition—the stock of capital at time 0.

Emissions of carbon dioxide from industrial sources, denoted $E(t)$, can now be modeled. Let

$$E(t) = \{1 - \mu(t)\}\{\sigma(t)\}\{A(t)K(t)^{\gamma}L(t)^{1-\gamma}\}. \tag{20.4}$$

The last term on the right-hand side is a scale factor based on gross economic activity from (20.2). The second term relates gross output to gross emissions representing the carbon intensity of production by $\sigma(t)$. This intensity can change with time, so

$$g^{\sigma}(t) = g^{\sigma}(0)\exp\{-\delta^{\sigma}t\}$$

with

$$\sigma(t) = \frac{\sigma(t-1)}{[1 + g^{\sigma}(t)]}$$

and $\sigma(0)$ again fixes an initial condition. The first term, finally, provides a control handle on the level of emissions. The $\mu(t)$ factor can cause net emissions to fall below gross emissions, but emissions reductions are not free. The same factor, modified by $b(t)$, worked in (20.2) to reduce net output. Reducing emissions in time t can therefore reduce either consumption or investment in time t, or both.

Equation (20.2) also allowed net economic output to be affected through $\Omega(t)$ by damage that could be attributed to climate change. To relate this damage to changes in global mean temperature, denoted relative to 1990 levels by $T(t)$, define

$$\Omega(t) = \frac{1}{(1 + D(t))},$$

where damages are taken to be quadratic in temperature

$$D(t) = \theta_1 T(t) + \theta_2 T(t)^2. \tag{20.5}$$

The θ_i parameters are both positive so that damages increase with temperature at an increasing rate.

All that remains at this point is to explain how emissions $E(t)$ are translated into changes in global mean temperature. Atmospheric concentrations of carbon dioxide, denoted $M_A(t)$, provide the critical link. The simplest representations of this link found in the early models hold that annual emissions contribute to concentrations according to

$$M_A(t) = M_{AT}^{PI} + \xi E(t-1) + (1-\delta_M)\{M_A(t-1) - M_{AT}^{PI}\}.$$

The parameter ξ is termed the "airborne fraction"—the proportion of emissions released in any year that stay in the atmosphere. Meanwhile, the parameter δ_M reflects the slower rate at which mixed atmospheric concentrations "leak" into the biosphere and the oceans, and M_{AT}^{PI} represents an historically based estimate of the preindustrial level of concentrations. The scientific community, as evidenced in [14, 16, 20, 27, 35], was skeptical about this reduced form representation of the carbon cycle, and Schultz and Kasting [36] suggested that its simplicity could be the source of systematic underestimation of future concentrations. As a result, more recent constructions have adopted a three-equation formulation that reflects the interactions between atmospheric concentrations, concentrations in the biosphere and upper oceans ($M_{UP}(t)$), and concentrations in the lower oceans ($M_{LO}(t)$):

$$\begin{aligned} M_A(t) &= E(t-1) + \phi_{11} M_A(t-1) + \phi_{21} M_{UP}(t-1); \\ M_{UP}(t) &= \phi_{22} M_{UP}(t-1) + \phi_{12} M_A(t-1) + \phi_{32} M_{LO}(t-1); \text{ and} \\ M_{LO}(t) &= \phi_{33} M_{LO}(t-1) + \phi_{23} M_{UP}(t-1). \end{aligned}$$

In either case, atmospheric concentrations are translated into atmospheric forcing according to

$$F(t) = \eta \left\{ \frac{\ln[M_A(t)/M_{AT}^{PI}]}{\ln[2]} \right\} + O(t), \tag{20.6}$$

where η is a forcing constant and $O(t)$ reflects forcing from concentrations of other gases that are taken to be independent of economic activity. Finally, increases in global mean temperature above 1990 levels are related to this forcing by another two-equation system that also tracks $T_{LO}(t)$, the temperature of the lower ocean:

$$T(t) = T(t-1) + \sigma_1\{F(t) - \lambda T(t-1), -\sigma_2\{T(t-1) - T_{LO}(t-1)\}\}, \tag{20.7}$$

where

$$T_{LO}(t) = T_{LO}(t-1) + \sigma_3\{T(t-1) - T_{LO}(t-1)\}.$$

Estimates of η, λ, the ϕ_{ij}, the σ_i, and initial conditions for T, T_{LO}, M_A, M_{UP}, and M_{LO} are drawn from the scientific literature. A concise description of the calibration process can be found in [31, pp. 57–67].

20.2.2 A variation on a theme: Highlighting energy services

Relating carbon emissions directly to the Cobb–Douglas structure recorded in equation (20.2) allows the modeling to capture increased carbon efficiency through the $\sigma(t)$ factor of equation (20.3), but changes in that parameter over time must reflect both energy conservation and price-induced substitution out energy. Since these are two separate economic phenomena, several variants of the model build energy services ES(t) directly into

the aggregate production function. Some incorporate fossil and nonfossil fuels directly into functional forms that are more general than Cobb–Douglas. Others, like RICE-99, continue in the Cobb–Douglas mode so that

$$Q(t) = \{\Omega(t)\}\{A(t)K(t)^{\gamma}L(t)^{1-\beta-\gamma}\,\mathrm{ES}(t)^{\beta} - c^{E}(t)\,\mathrm{ES}(t)\}, \tag{20.8}$$

where $c^{E}(t)$ represents the cost of producing carbon-based energy. Emissions of carbon can then be directly related to energy intensity according to

$$E(t) = \{1 - \varsigma(t)\}\,\mathrm{ES}(t) \tag{20.9}$$

with another set of motion equations designed to track secular trends over time:

$$\varsigma(t) = \varsigma(0)\exp\int_{0}^{t} g^{Z}(v)dv$$

and

$$g^{Z}(t) = g^{Z}(0)\exp\{(-\delta^{Z})t\}.$$

Meanwhile, energy employment decisions minimize cost by setting the (net) marginal product of energy equal to a price that reflects the sum of production costs, a Hotelling [6] rent $h(t)$ that captures the effect of current extraction on future costs, and a carbon tax $\tau(t)$ that effects the desired rate of emission control. Notice that the carbon tax replaces $\mu(t)$ as a control variable in this formulation.

20.2.3 Another variation on a theme: Regional disaggregation

Recent analyses have also focused attention on models that disaggregate economic activity across geographical regions for a variety of reasons. Their structures are designed to capture diversity across regions in production, investment, carbon intensity, vulnerability to climate change, and other important distinguishing characteristics, but accommodating diversity is not necessarily the most important rationale for adding complexity to a programming problem that is already very difficult. The Kyoto Protocol has made it clear that global climate policy will treat different countries differently. As a result, it is important to be able to reflect that potential not only in global optimization problems but also in second-best contexts where, for example, developing countries are held to different standards and are afforded different opportunities than developed countries (see http://www.unfccc.de for details).

RICE-99 from [31] is an example of such a model; it divides the globe into nine regions according to economic criteria and policy relevance. Other models divide the globe into important geographical regions. In either case, however, there is more to the process of disaggregation than producing different calibrations and attaching identifying subscripts to all of the regionally specific parameters. The regions must be integrated so that global portraits of economic activity, emissions, and climate change can be discerned. RICE-99, for example, links countries by allocating emissions permits to different regions and allowing them to be traded. The total number of permits thereby becomes the control variable, but

the market clearing price tracks the tax rate $\tau(t)$ identified in the previous section. As a result, any country J can augment or diminish its economic product in any year by selling some of its allocation of $\Pi_J(t)$ permits if it has a surplus or buying some additional permits if it has a deficit. As a result

$$Q_J(t) + \tau(t)\left\{\prod_J(t) - E_J(t)\right\} = C_J(t) + I_J(t) \tag{20.10}$$

replaces (20.3). In words, (20.10) notes that revenue generated for region J by the sale of permits can be allocated to consumption, investment, or both, but expenditures required by region J to purchase permits must be deducted from consumption, investment, or both.

20.2.4 Characterizing the solution

The complexity of even the simplest model of the interaction of climate with the economy makes it extraordinarily difficult to record solutions to the optimization problem. Indeed, solution techniques generally search iteratively across time as the future unfolds to specify trajectories that support maximum levels of discounted utility to predesignated levels of precision. Underlying economic theory can, however, sustain intuitive descriptions of the optimality conditions for critical control variables. In the basic model portrayed in section 20.2.1, for example, investment and the control rate of emissions were identified as control variables. Investment over time will support optimization if it proceeds from year to year so that the rate of return on capital matches the sum of the discount rate on utility and the contemporaneous rate of capital depreciation. Satisfying this condition will guarantee that the marginal cost of the last unit of investment, expressed in terms of the marginal utility of the last unit of foregone consumption, matches the discounted value of the marginal utilities of higher future consumption that optimally will be sustained by the resulting increase in the capital stock.

Similarly, "investment" in reduced emissions will support optimization if the marginal utility of the last unit of consumption foregone matches the discounted value of the marginal utilities of additional consumption that optimally will be sustained over future years by the resulting reduction in climate related damages.

Adding the regional disaggregation described in section 20.2.3 imposes a third requirement on the solution: the marginal cost of carbon abatement must be equal across all countries. This condition can be achieved by changing the control variable on emissions into a tax or a permit price. The marginal cost of investment in emissions reduction in any country will then be set equal to the tax or the permit price, so they will all be equal to each other. It follows that setting either economic (shadow) price (directly or by properly defining the number of permits) equal to the discounted value of the marginal utility of reduced future damage at any point in time will guarantee that condition (20.2) is satisfied.

Finally, adding energy services to the mix as in section 20.2.2 produces a fourth requirement: the Hotelling [6] rent on carbon-based energy must be a true scarcity rent in the sense that the extraction costs are inflated by a factor that climbs at the rate of interest. Rent then equals the discounted value of the intertemporal social effect of marginal extraction on the cost of extraction.

Table 20.1. *Baseline emissions trajectory and optimal control: RICE-99 results* [31].

Year	Baseline emissions*	Optimal emissions	Optimal control rate
1995	6.2	5.9	3.9%
2005	7.2	6.9	4.8%
2015	7.9	7.5	5.6%
2025	8.4	7.9	6.2%
2035	8.9	8.3	6.9%
2045	9.5	8.7	7.5%
2055	10.0	9.2	8.2%
2065	10.6	9.7	8.8%
2075	11.2	10.2	9.4%
2085	11.9	10.7	10.0%
2095	12.6	11.3	10.5%
2105	13.3	11.8	10.9%

*Emissions measured in gigatons of carbon per year (Gt/yr).

20.2.5 Some results

Table 20.1 displays baseline and economically optimal time series for emissions and control rates reported by [31] for the RICE-99 model described above. They depict smooth and modest reductions in emissions that are typical of results drawn from these sorts of dynamic optimization exercises. Specifically, the table reports an 8.2% reduction in cumulative emissions through 2100 that improve global welfare by something on the order of $250 billion 1990 U.S. dollars (substantially less than 1% of total world product in 2100) with a benefit-cost ratio of 3.02. Nordhaus and Boyer [31, Chapter 7] provide descriptions of time series of other state variables as well as detailed descriptions of the requisite underlying calibrations for regional production functions, regional damage functions, appropriate equations of motion, and the other model components. All of these calibrations play a critical role in determining the numerical results, of course, and their potential sensitivity to alternative specifications raises a litany of derivative questions even on a global level. Are the optimal trajectories sensitive to model specification? to calibrations that affect the underlying baselines? to different assumptions about climate sensitivity? to different assumptions about the shape of the damage function? The answer to each of these questions is emphatically affirmative.

A series of experiments conducted by participants in the Energy Modeling Forum (EMF) and others can provide some insight into the answers to the first of these questions. Figure 20.1, for example, contrasts comparable results reported to the EMF [2] by five other models with the RICE-99 results; details of the EMF process can be found in [4]. All of these models were run with consistent calibrations for the EMF experiment, but they differ significantly in modeling approach and emphasis. The HCRA model (see [5]), for instance, is based on aggregate constructions of the sort described above. Others, like CETA [32, 33], MARIA [26], and MERGE [22], have relatively elaborate energy sectors, and ICAM [1]

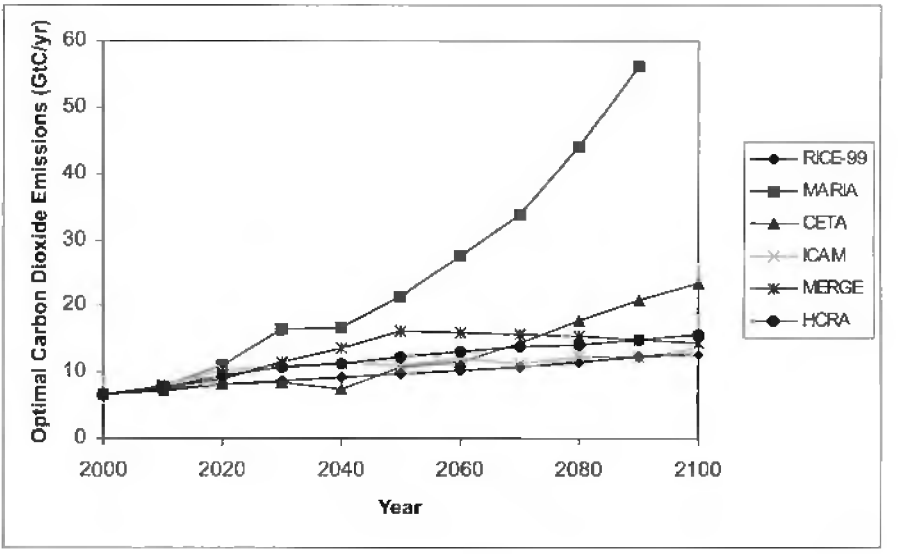


Figure 20.1. *Optimal carbon emissions for RICE-99 and five other models using common calibrations.*

incorporates an enormous amount of disaggregated detail across multiple economic sectors and multiple household decision makers. All of them produce trajectories that are similar to the Nordhaus results (also portrayed in the figure for reference), but it is clear that outcomes diverge substantially by the middle of the next century. Modeling structure, or model uncertainty in the parlance of the climate literature, clearly matters quantitatively if not qualitatively.

Results reported by Yohe and Garvey [46] showed that different assumptions about baseline economic activity and climate sensitivity can both produce dramatically different optimal control results well before the year 2050 even from the same model. Figure 20.2 displays the RICE-99 baseline in addition to four representative baseline emissions trajectories drawn from a Monte Carlo simulation of possible futures. High emissions trajectories were driven by high population growth, limited substitutability out of carbon-based energy sources, and large fossil fuel reserves; low emissions trajectories were the result of opposite configurations. Optimal interventions were then computed along each scenario for different climate sensitivities ranging from 1.5°C to 4.5°C for a doubling of carbon equivalent atmospheric concentrations. Panel A of Figure 20.3 portrays the results graphically in terms of cumulative emissions reductions; panel B displays corresponding estimates of the initial marginal cost of carbon emissions in 1995 for different control targets. For reference, the RICE-99 results set this initial marginal cost equal to $9.13 per ton of carbon for its optimally targeted 8.2% cumulative control rate. The RICE-99 results therefore fall on the lower boundary of the range depicted. Notice, as well, that different emissions futures and/or different climate sensitivities not reflected in RICE-99 can easily support wildly variant results.

Roughgarden and Schneider [34] produced similar variation from the same DICE-

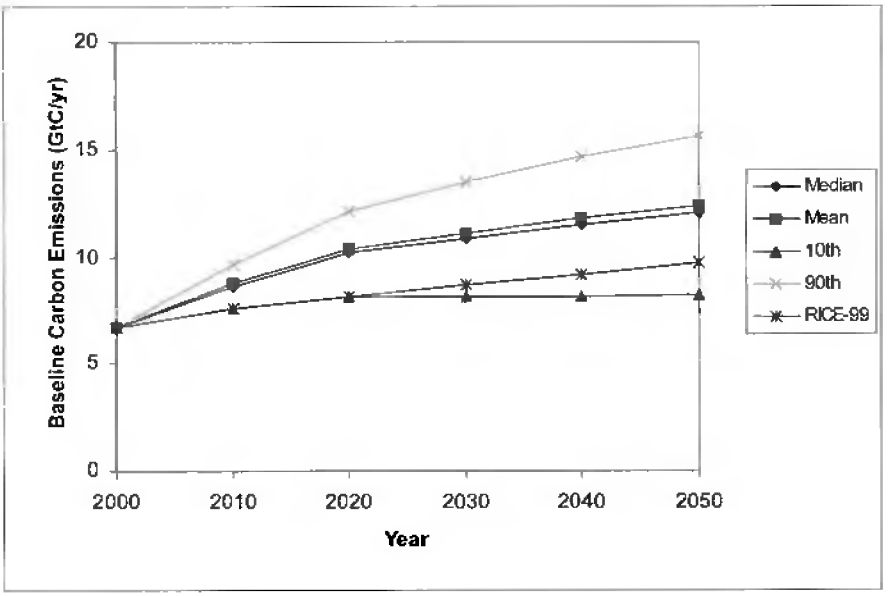
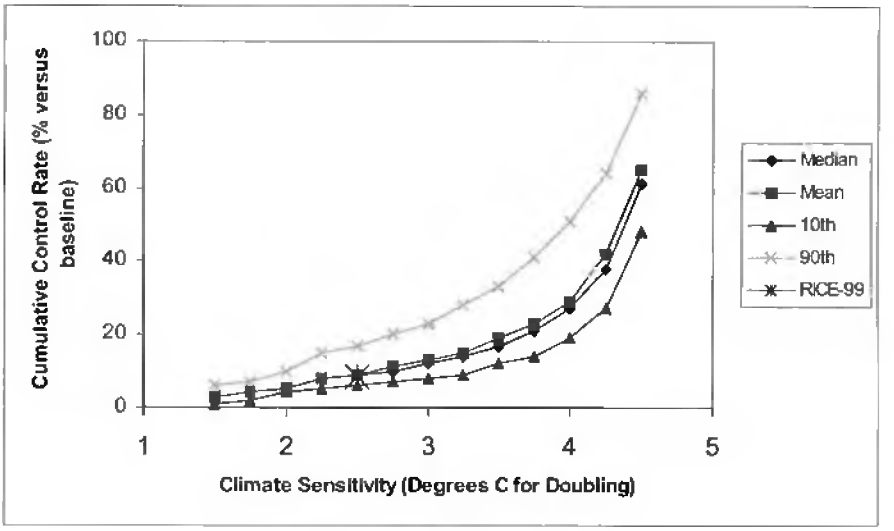


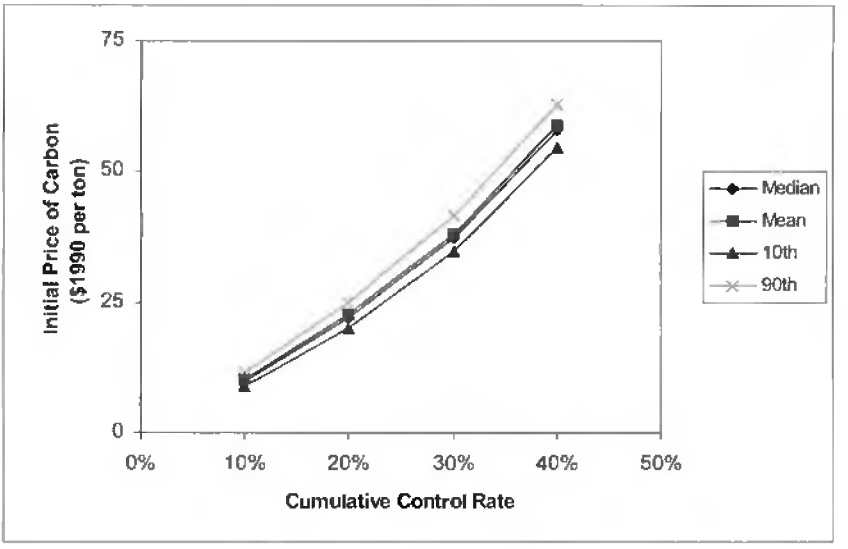
Figure 20.2. *Baseline unregulated carbon emissions for RICE-99 and representative scenarios drawn from a Monte Carlo simulation exercise of a DICE-like antecedent model.*

like model by using alternative damage estimates, some of which include widely different perceptions about the potential for abrupt climate change. Figure 20.4 summarizes their approach and their results. Panel A displays different damage functions drawn from a survey conducted by Nordhaus [29, 25]. The RICE-99 calibration lies everywhere above the average social scientist view but below estimates offered by environmental scientists, and natural scientists were the most pessimistic. Panel B shows associated optimal control rates; these are solutions to the optimization problem for the different damage functions. Panel B makes it obvious that different perceptions of damage clearly sustain dramatically different views of how optimally to control emissions over time.

This is not a surprise, but two less obvious insights can be drawn from panel B. First, two control trajectories are displayed there for the median damage estimate. The lower trajectory would be optimal if the pure rate of time preference were set equal to 3% per year (i.e., if $\rho(t) = 0.03$ for (20.1b) and (20.1c) for all t). The higher trajectory (almost 75% higher for every year) would be optimal, though, if nothing changed but the rate of time preference. In particular, the lower median trajectory reflects an economically optimal solution if $\rho(t) = 0.015$ for all t. Lower discount rates mean that future damages cause more harm, in discounted value; panel B shows that they therefore mean that increased control rates must be imposed. Second, optimal control rates from the RICE-99 are the lowest although the underlying damage function is not. RICE-99 includes both regional diversification and an ability to substitute endogenously out of carbon-based energy as its relative price climbs while the earlier DICE models did not. These two innovations make it easier to reduce emissions along the regulated scenario, but they also apparently make it



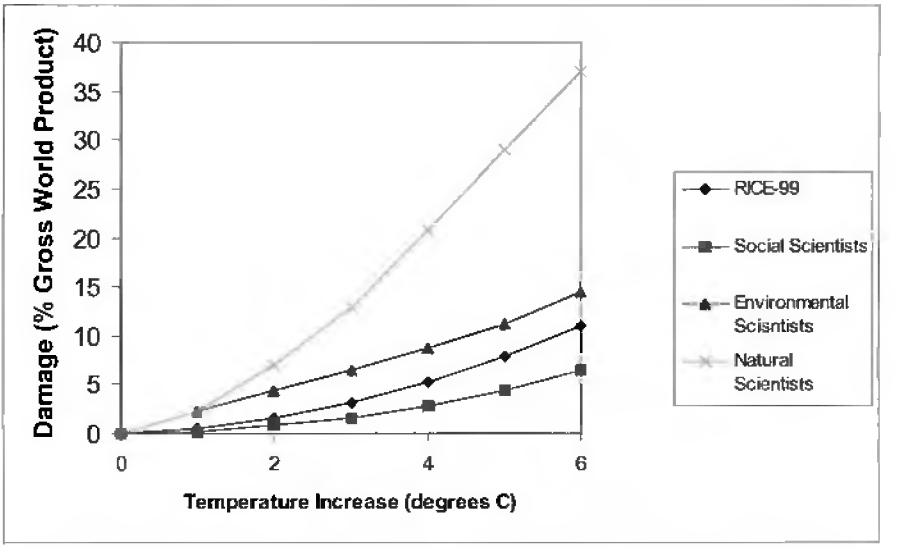
A



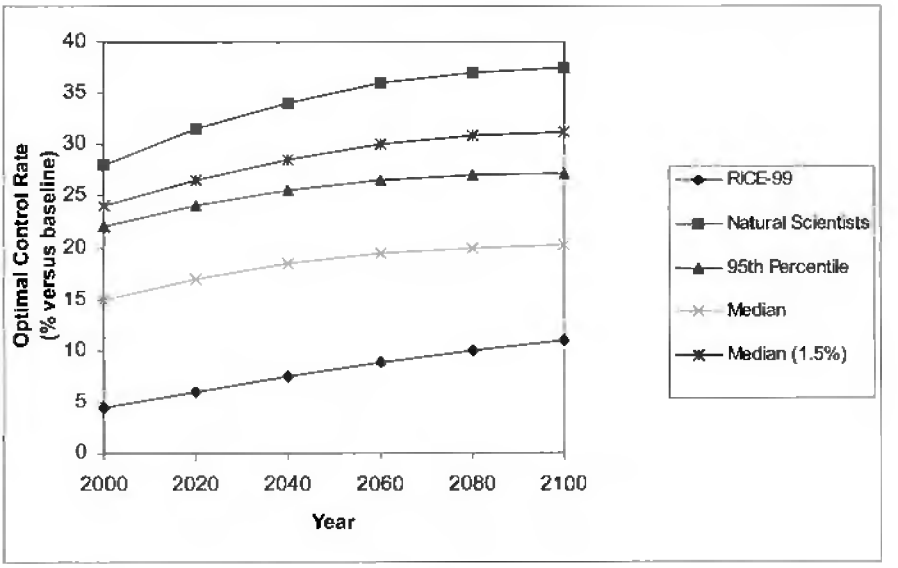
B

Figure 20.3. (A) *Optimal cumulative control rates for representative baseline unregulated emissions scenarios and alternative calibrations for climate sensitivity.* (B) *Initial carbon prices for optimal emissions control along representative baseline unregulated scenarios.*

easier to reduce emissions along the unregulated baselines against which control rates are measured; panel B shows that the optimal control rate need not, as a result, be as aggressive.



A



B

Figure 20.4. (A) *Damage functions from alternative perspectives.* (B) *Optimal transient control rates for alternative damage specifications.*

20.3 Optimization in an uncertain environment

Uncertainty cascades through any attempt comprehensively to represent the climate system. Such a description must begin with a depiction of future human activity. It must work its

way through some very complicated atmospheric and natural science to produce portraits of global or regional impacts. It must correlate those impacts with human adaptation to depict vulnerability. And finally, it must try to solve a dynamic programming problem with some sort of representation of global welfare. Simple applications of the dynamic optimization approach to this uncertain environment have been attempted, and they have offered some insight into the sensitivity of near-term and medium-term control rates to both uncertainty and alternative welfare functions. We begin with a brief review of this tactic before spending more time on related second-best exercises.

Most social welfare functions for regionally disaggregated models take the form

$$W\{c(t), L(t)\} = \sum(\omega_j U_j(u_j(c_j(t)))), \quad j = 1, \ldots, n, \tag{20.11}$$

where ω_j represent regional weights. In writing (20.11), moreover, unsubscripted variables depict vectors, so that

$$c(t) = \{c_1(t), \ldots, c_n(t)\},$$
$$L(t) = \{L_1(t), \ldots, L_n(t)\};$$

the subscripts identify per capita consumption and population in regions 1 through n. A Benthamite version of social welfare would, for example, set

$$\omega_j \equiv \left\{\frac{L_j}{(L_j + \cdots + L_j)}\right\}. \tag{20.12}$$

If regional utility were defined, as usual, as the natural log of regional per capita consumption, then application of this structure shows that maximizing discounted expected utility would add modestly to the strength of optimal mitigation along the median emissions scenario. In other words, recognizing stochastic elements would add a risk premium (of as much as 30%) to the optimal marginal cost of emissions along the median possibility (see, for example, [39]).

Fankhauser, Tol, and Pearce [3] have, however, argued that regional weights of the form

$$\omega_j \equiv \left\{\frac{\mu_c}{c_j}\right\}, \quad j = 1, \ldots, n, \tag{20.13}$$

would be more appropriate weights if regional utility functions were logarithmic, as in the original formulation in section 2 and if (20.11) were applied as the global objective function. It is important to note that relative risk aversion for a logarithmic objective function displays constant relative risk aversion equal to -1 $(= u''(c)c/u'(c))$ but a declining absolute measure of risk aversion that declines with consumption (i.e., $u''(c)/u'(c) = (1/c)$). It is, however, widely known that constant relative risk aversion leads individuals to allocate a constant proportion of their income or wealth to the purchase of insurance, so the logarithmic function does pick up a benchmark aversion to risk. Here, μ_c represents average global per capita consumption so that ω_j can be greater than or less than one depending on whether a region's per capita consumption is below or above the global average, respectively. Poor countries receive more weight, as a result, but to what end? Table 20.2 shows the marginal cost of carbon that maximized discounted expected utility for four time horizons reported by Tol

Table 20.2. *The optimal marginal cost of carbon dioxide emissions for simple sum* (20.12) *and equity-based* (20.13) *weights (constant dollars per ton of carbon with standard errors in parentheses)* [38, *Table* 4].

	0% Discount rate		1% Discount rate		3% Discount rate	
Year	Simple sum	Equity weight	Simple sum	Equity weight	Simple sum	Equity weight
2050	3.2 (5.2)	3.2 (1.5)	2.6 (4.0)	2.4 (1.1)	1.8 (2.5)	1.4 (0.6)
2100	5.9 (3.9)	8.8 (5.8)	3.9 (3.0)	5.1 (3.1)	2.1 (2.2)	2.1 (1.0)
2150	11.4 (7.0)	24.9 (96.3)	5.5 (3.1)	9.2 (24.2)	2.2 (2.1)	2.4 (2.2)
2200	25.0 (57.9)	∞ (∞)	7.7 (9.1)	∞ (∞)	2.3 (2.1)	∞ (∞)

[38] for three different discount rates and the regional weights given by (20.12) and (20.13). Near- to middle-term costs can be smaller for the equity-weighted solution, especially with a relatively high discount rate. It would seem that near-term climate policy can effect poor countries, too. However, the marginal cost of carbon that defines the equity-weighted solution is higher in the distant future in all cases. Why? The expected value calculation of social welfare includes a few extreme scenarios that drive poor regions toward subsistence, and so the weights applied to their plights climb as their utility plummets. As a result, adding equity weights to the stochastic problem can produce optimal risk premiums well in excess of 100%.

As interesting as these insights may be, they may be drawn from answers to what is fundamentally an ill-conceived question. It is obvious that one cannot simply compute the mean of the solutions for probabilistically weighted scenarios that span an uncertain range of possible futures to see what should be done over the next few decades. To do so would be appropriate only if everything in sight were linear, and it surely is not. It may, however, be equally inappropriate to saddle the actors with the full range of uncertainty and compute future control rates that maximize expected welfare. Expected value calculations have trouble accommodating high-consequence events to which current understanding would assign low probabilities, but the problem is much more subtle than that. The bottom row of Table 20.3, replicated from [38], shows infinite marginal costs attributed to carbon for equity-weighted social welfare functions. It seems that marginal damages could become undefined over finite periods of time along many "not implausible" futures that include sudden surprises although absolute damages are finite. All that is required is that some region with a finite weight collapse to its subsistence level. If that happens, our arithmetic would surely have trouble summing across those futures to set the expected marginal cost emissions reduction equal to the corresponding expected marginal damage even with significant discounting of the distant future.

In addition, it is unrealistic to assume that policy makers in an uncertainty environment would be willing to sit down with their analysts to make optimal control decisions that would be in force forever. That is, no policy maker would accept the notion that he or

Table 20.3. *Definition of the parameters that produce representative scenarios* (20.1)–(20.7) [46].

Scenario	Subjective probability	Population growth	Technological change	Depletion	Substitution elasticity
1	0.27	High	High	Low	High
2	0.13	High	Medium	Medium	High
3	0.23	Low	Medium	Medium	Medium
4	0.19	Medium	Medium	Low	Medium
5	0.09	Medium	Low	High	Low
6	0.05	High	Low	Medium	Low
7	0.04	High	Low	Low	Low

she or successors will never return to reconsider those decisions as more information about what the future might hold is delivered to their desks. Realistic investigations of how to proceed must, therefore, include some representation of midcourse corrections. For these reasons, and others that could be added to a growing list of concerns, extreme care must be taken in defining appropriate and realistic questions as the analysis moves into a stochastic environment.

Variability in the results drawn from deterministic analyses can, however, provide some guidance in this endeavor. Figures 20.1–20.4 immediately identify sources of uncertainty that both satisfy a cautionary rule of thumb articulated by Lave in the mid 1980s and offer some access to meaningful stochastic programming problems that involve learning and contingent adjustments in control rates. What is the Lave rule? It is, "Don't spend much time worrying about uncertainty about a variable or a structure that cannot influence outcomes by less than a factor of two (one way or the other) *because their effects all lie within the noise of our understanding of the large climate system.*" Figures 20.1–20.4 show that model uncertainty, baseline uncertainty, climate sensitivity uncertainty, and damage uncertainty all satisfy the rule. Moreover, it is not unreasonable to believe that global decision makers will learn more about each of these uncertainties over the next few decades; thus it makes sense to model how these decision makers might respond to improved information in the future and explore how they might act in the meantime. This section reports on the results of a policy experiment designed to explore answers to these questions in a stylized environment of hedging and learning with respect to climate and damage uncertainty across a not-implausible range of baseline uncertainty.

20.3.1 Specification of a stochastic environment: A hedging experiment

Figure 20.2 displayed representative unregulated global emissions trajectories through 2050 drawn from a Monte Carlo simulation of a DICE-like model of aggregate economic activity reported in [47]; they were selected by employing a methodology reported initially in [43].

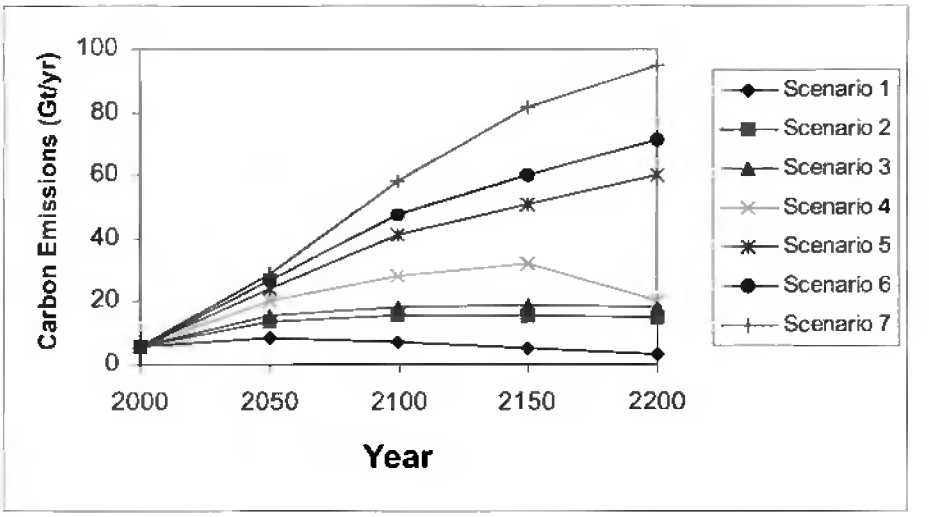


Figure 20.5. *Baseline unregulated carbon emissions for representative scenarios drawn from a Monte Carlo simulation exercise of a DICE-like antecedent model; their subjective relatively likelihoods are recorded in Table* 20.2.

Figure 20.5 offers an alternative portrait of the same simulation results through the year 2200. Seven distinct trajectories are portrayed there, and Table 20.3 provides some insight into why they are so different. High emissions were produced in scenarios (20.5)–(20.7) by combinations of high rates of population growth with low rates of technological change, low abilities to substitute out of carbon-based fuel (i.e., a low elasticity of substitution between carbon- and noncarbon-based fuel), and/or a low price trajectory for carbon-based fuel. Low emissions similarly emerged from scenarios (20.1) and (20.2); they were the result of high rates of technological change and/or a high elasticity of energy substitution even with high population growth rates. The second column records the subjective likelihood of each representative scenario derived from the simulation.

Panel A in Table 20.4 reproduces the underpinnings of an experimental design distributed among members of the Uncertainty Working Group of the EMF by Manne and Nordhaus in 1996. Manne [21] and Weyant [41] describe this process based on a structure initially portrayed in [23, 24]. It shows the summary statistics of two surveys of expert opinion on climate sensitivity and worldwide damage associated with a 3°C increase in global mean temperature [25, 29]. Based on these statistics, the experiment asked participants to quantify high-consequence cases by scaling their baseline values up to the conditional means of the upper 5% tail of the surveys. For a base case climate sensitivity of 2.5°C taken from [8], adding the 2.3°C dispersion reported in panel A produced a 5% extreme value of 4.8°C. Similar scaling by the 7.8-to-1 ratio also reported in panel A produced a 5% extreme value for potential damage equal to 12.48% of gross world product from a base case damage estimate of 1.6%. Under the assumption that estimates of damages and climate sensitivity were independently distributed, cases U_1, U_2, and U_3 identified in panel B of Table 20.4 could therefore be assigned relative likelihoods of 0.25%, 5%, and 5%, respectively, when

Table 20.4. *Definition of the extremes* [24, 47]. *Parameterization of the states of nature.**

A. Characterization of extreme outcomes

Variable	Survey median	Standard deviation	Mean of the top 5%
Climate sensitivity	2.8°C	1.4°C	5.1°C
Damage with a 3°C warming	1.75%	3.3%	13.6%

B. Specification of extreme events

Case	Description	Specification	Subjective likelihood
U_0	Climate sensitivity = 2.5°C Damages = 1.6% of GWP	$\Theta_1 = 0$; $\Theta_2 = (0.016/9)$; $\eta/\lambda = 2.5$;	$\{1 - \text{prob}(U_k)\}$
U_1	Climate sensitivity = 4.8°C Damages = 12.48% of GWP	$\Theta_1 = 0$; $\Theta_2 = (0.1248/9)$; $\eta/\lambda = 4.8$;	$0 < \text{prob}(U_k) < 0.05$
U_2	Climate sensitivity = 4.8°C Damages = 1.6% of GWP	$\Theta_1 = 0$; $\Theta_2 = (0.016/9)$; $\eta/\lambda = 4.8$;	0.05
U_3	Climate sensitivity = 2.5°C Damages = 12.48% of GWP	$\Theta_1 = 0$; $\Theta 2 = (0.1248/9)$; $\eta/\lambda = 2.5$;	0.05

*Notation applies to (20.5)–(20.7).

compared to case U_0. The probability of U_1 could climb to as high as 5% relative to U_0 if high sensitivity and high damages were perfectly correlated.

A set of simple stochastic programming problems can now be defined for hedging in pairwise comparisons of U_0 with U_1, U_2, and U_3. Using the global model, the solution chooses investment trajectories $I*(t)$ and control rate trajectories $\mu*(t)$ that maximize

$$\sum_{t=0}^{20} E\left\{\frac{U[c_J(t), L(t)]}{R(t)}\right\} + E\left\{\sum_{t=21}^{n} \frac{U[c_J(t), L(t)]}{R(t)}\right\},$$

where the expected value operator incorporates the subjective probabilities of case U_0 versus U_J (for $J = 1, 2$, or 3) recorded in panel B of Table 20.4. The states of nature for each case are also depicted in Table 20.4. The trajectory of the per capita consumption state variable that appears in both parts of this objective function is constrained by the structure and the equations of motion that defined the simple variant of the complete model described in section 20.2. The initial conditions for the first part are the same as in the deterministic case, but the initial conditions for the capital stock and the stock of atmospheric concentrations of carbon dioxide that anchor the second part of the objective function are determined endogenously by outputs of the hedging trajectory over the first 20 years. All uncertainty

will be resolved at that time, but the actual outcome of that resolution is unknown at time zero—thus the application of the expected value operator in the second term.

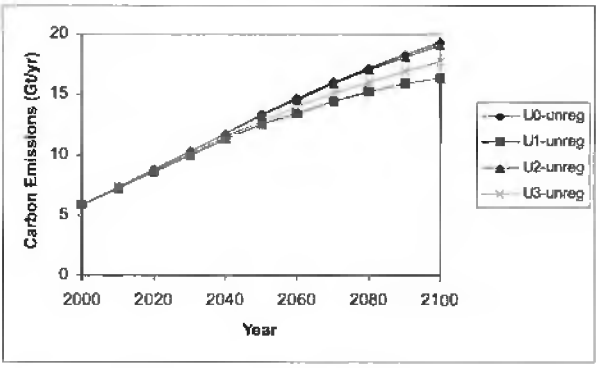
20.3.2 Some results from the stochastic environment

Figure 20.6 focuses initial attention temporarily on comparisons of the three cases along scenario (20.3). Notice from Table 20.3 that scenario (20.3) is the functional representative of a median emissions trajectory absent any regulatory intervention and any incorporation of climate-related damages. Unregulated emissions trajectories are displayed in panel A of Figure 20.6 for the four cases (U_0–U_3); they are different for the four cases where climate-related damages are included even though the underlying economic drivers are the same. The deviations portrayed there may not satisfy the Lave rule for significance, but their implications cannot be ignored. Panel B shows why by displaying optimal emissions paths for U_0–U_3. Panel C corroborates this insight with a portrayal of associated emissions control rates. The results displayed in these panels are, quite simply, the solutions to the deterministic problem under the assumption that decision makers can correctly identify the correct damage-sensitivity combination before initiating the long-term optimal intervention.

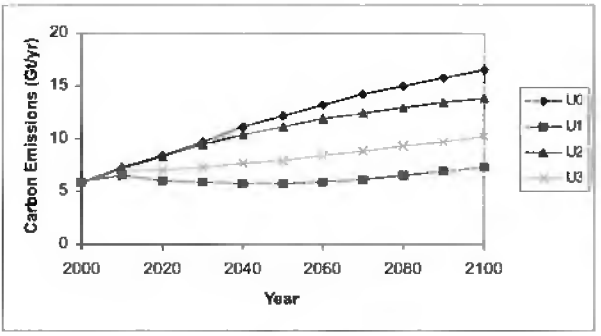
Uncertainty can now be brought to bear on those decisions through pairwise comparisons of possible futures along different emissions scenarios. In Figure 20.7, for example, decision makers confronting scenario (20.3) do not know if case U_0 or U_1 will materialize until the year 2020, but they do expect that their estimates of economic damage and climate sensitivity are independently distributed. They can, therefore, presume that case U_1 has a 0.25% chance of materializing. Panel A contrasts optimal emissions trajectories for cases in which U_0 or U_1 futures were accurately foreseen with trajectories that would optimally hedge until 2020. The hedging paths, of course, converge along turnpikes to the optimal path for actual state of the world after it is revealed. Panel B displays the corresponding control rate trajectories. Notice in both representations that convergence to the U_0 optimal path would be relatively rapid; the near-term hedging trajectories do not deviate significantly from the U_0 optimum through 2020, so little correction would be required. By way of contrast, convergence to the U_1 optimal path would be slow and extended because near-term emissions would have been significantly and suboptimally high for two entire decades until uncertainty was resolved.

The alternative futures portrayed in Figure 20.7 are typical for comparisons of case U_0 with cases U_1, U_2, and U_3 along all seven emissions trajectories. However, estimates of the expected discounted value of the costs associated with deviating from the optimal paths before and after 2020 vary widely across those futures. Table 20.5 expresses those expected costs as estimates of the expected discounted value of perfect information. That is to say, Table 20.5 records the expected discounted value of control costs and climate damages that could be avoided in each case if the decision maker could have known in 2000 whether U_0 or some other U_j case would materialize with certainty. Each was computed using optimal discount rates based on a pure rate of time preference of 3% with a logarithmic utility function in per capita consumption, and each assumes that climate-related damage and climate sensitivity are independently distributed.

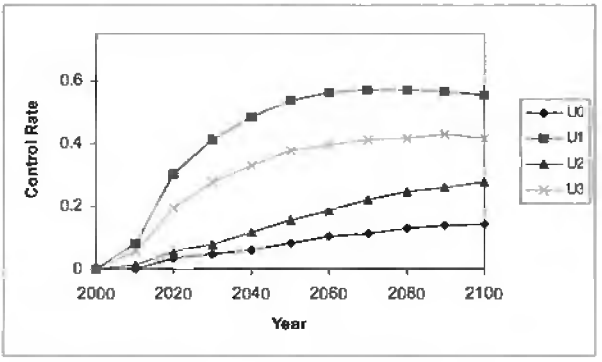
Instructive patterns emerge in Table 20.5, but they are not obvious. For example, the cost estimates are higher for the higher emissions of scenario 2 relative to scenario 1 for all



A

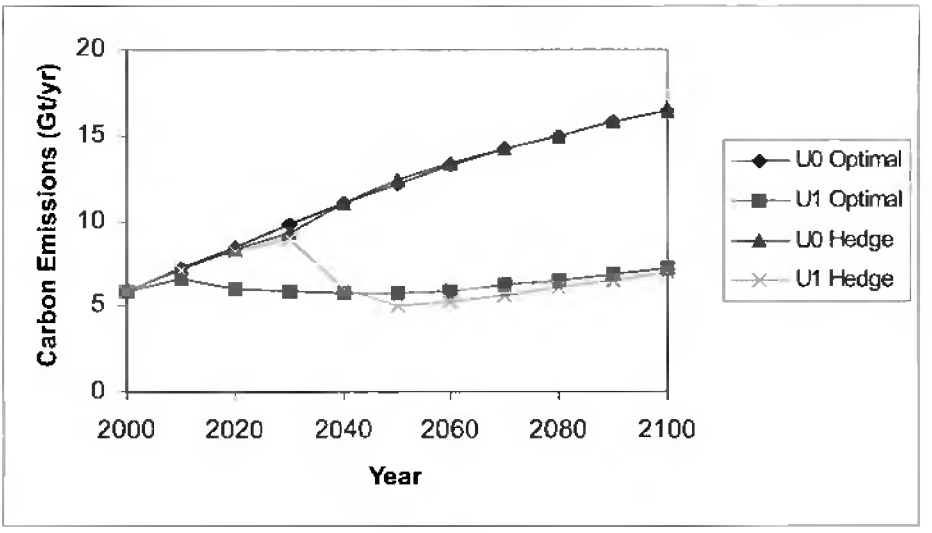


B

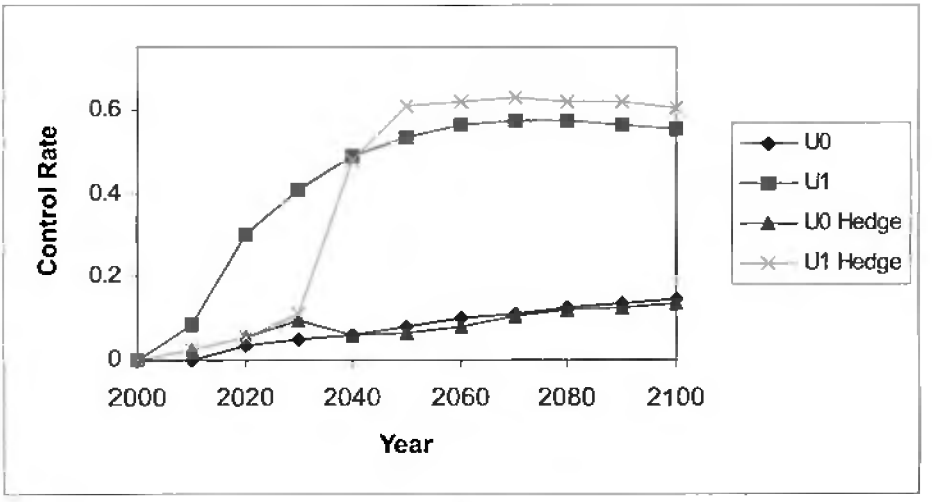


C

Figure 20.6. (A) *Unregulated carbon emissions along scenario* (20.3) *for alternative assumptions about climate sensitivity and economic damage associated with a* 3°*C increase in global mean temperature. These are the* U_0–U_3 *cases identified in Table* 20.3. (B) *Optimally regulated carbon emissions along scenario* (20.3) *for alternative deterministic assumptions about climate sensitivity and economic damage associated with a* 3°*C increase in global mean temperature. These are the* U_0–U_3 *cases identified in Table* 20.3. (C) *Optimally regulated control rates along scenario* (20.3) *for alternative deterministic assumptions about climate sensitivity and economic damage associated with a* 3°*C increase in global mean temperature. These are the* U_0–U_3 *cases identified in Table* 20.3.



A



B

Figure 20.7. *Adjustment paths showing the corrections required from the hedging trajectories relative to the optimal deterministic solutions. Emissions depicted in panel A fall below the optimal after correction for the lower U1 case because, as seen in panel B, the hedge would exert too much control in the early years. Also, emissions would rise above the optimal for the highter U0 case because the hedge would then exert too litle control.*

three comparisons although both scenarios assume a high elasticity of energy substitution. The same pattern can be seen for middle-elasticity scenarios 3 and 4 and for low-elasticity scenarios 5–7. Moreover, cost estimates climb even when higher emissions can be attributed in part to lower elasticities. High estimates for expected cost can therefore be anticipated

Table 20.5. *Expected discounted value of perfect information (billions of constant* 1990 *dollars)* [47].

Scenario	U_0 versus U_1	U_0 versus U_2	U_0 versus U_3
1	5.32	1.20	38.55
2	10.04	2.02	48.55
3	5.29	0.70	19.70
4	6.92	0.98	31.65
5	5.97	0.92	30.60
6	7.80	1.18	37.15
7	9.16	1.52	48.90

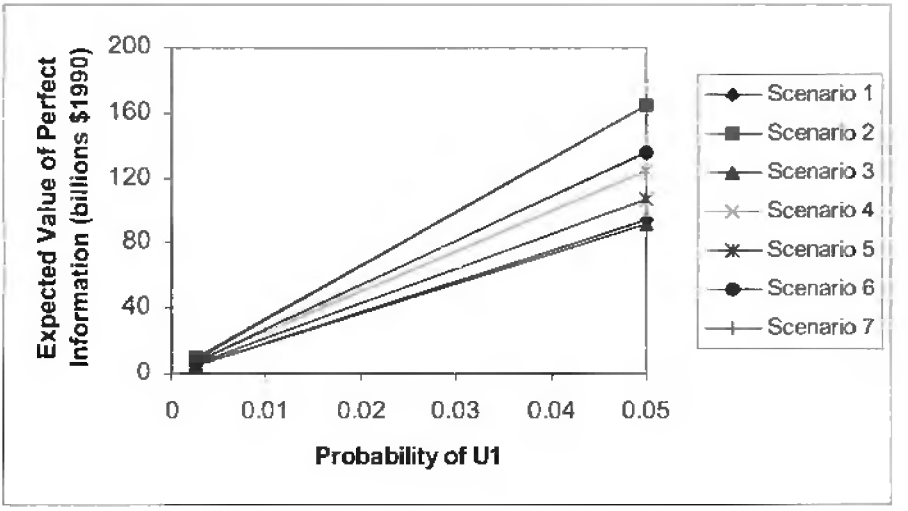


Figure 20.8. *The expected value of perfect information against optimal hedging for a U_0–U_1 comparision along scenario* (20.3) *for alternative likelihoods that U_1 will materialize. Cases U_0 and U_1 are described in Table* 20.3; *the values for probability* 0.0025 *are recorded in Table* 20.4.

not only in cases where emissions are high because substitution out of carbon-based fuel would be limited but also more generally in cases where emissions are higher for whatever reason. The errors involved in misallocating mitigation efforts over the near term are simply exaggerated when emissions are high.

Figure 20.8 displays a second general result by depicting the degree to which the value of information for the U_0–U_1 comparison depends on the assumed independence between the high-damage and high-sensitivity extremes. If U_2 and U_3 were independent events with individual subjective likelihoods equal to 5%, then their joint likelihood in case U_1 would equal 0.25% and the values reported in Table 20.5 would be appropriate. If U_2 and U_3 were

positively correlated, however, then their joint likelihood could climb to a maximum of 5%. The expected discounted value of perfect information should therefore climb. Figure 20.8 shows this result, but it should be noted that even a 20-fold increase in the likelihood of U_1 (from 0.25% to 5.0%) would not cause a 20-fold increase in expected costs along any scenario. More important, however, increasing the likelihood of the U_1 case means that emissions along the optimal short-term hedging trajectory would track further from the U_0 certainty path.

20.3.3 Real-world context and applied stochastic programming

The stochastic process depicted here is very simple. Why not model an annual process that could produce expanding sets of multiple states of nature as the future unfolds? Because the climate research community has not been convinced that there would be much point to undertaking such an exercise. Climate researchers try to be careful to keep track of who would know what, when they would know it, and what they would do after they found out. They argue that modeling decisions in the context of the simplistic stochastic process described above is extraordinarily optimistic if the noise is so large that the "factor of two" criterion applies even in approximation. They expect that improvement in our understanding of climate processes and our abilities to separate signal from noise will be slow, so a 20-year time horizon for resolving any uncertainty (much less all uncertainty) about extreme events (much less quantifying damages associated with those events) is illustrative at best. Moreover, they have observed that the international institutions that would be required to administer a global climate policy do not yet exist, and they expect that those institutions will be extremely cumbersome when they are invented. Global climate policy will always reflect enormous inertia toward maintaining the status quo. In short, they see no good reason to incorporate more elaborate processes in their optimization models that will not inform the decisions of any of the actors that they depict in those models. Maybe someday, but not now.

It would, of course, also be possible to add more two-decade intervals to the learning process with Bayesian updating before the uncertainty would be resolved. While this might be a reasonable next step, the community still questions whether their addition would produce new insight into what factors are important sources of uncertainty and where to spend scarce research resources (money *and* time) so that we most effectively weigh expected benefits and cost. This is a researchable question, to be sure, but it has yet to be confronted as a stochastic programming problem notwithstanding the possibility, indicated elsewhere in this volume, that such an approach might produce fundamentally different solutions.

20.4 An alternative approach: Cost effectiveness

The modeling structure described in section 20.2 need not be employed exclusively in the pursuit of economically efficient approaches to the climate problem. Its most useful application may in fact be to search for control trajectories that minimize the cost of achieving specific environmental targets with full recognition that those targets may not be the product of economic analysis. Signatory nations to the Framework Convention of Climate Change have, for example, committed themselves to limiting atmospheric concentrations

of greenhouse gases to a level, yet to be determined, that will prevent "dangerous" anthropogenic interference with the climate system. The Kyoto Protocol, to take another example, would restrict the emissions of the world's industrial countries to specific targets by the year 2012, and negotiations for second-round limitations have begun (notwithstanding the paucity of nations who have actually committed themselves to the Kyoto timetable). See http://www.unfccc.de for details. It is becoming clear that these sorts of targets may be determined by ethical considerations, political maneuverings, applications of an ecological precautionary principle, or any one of a multitude of other perspectives that are not rooted in economic efficiency. As a result, economists have come to recognize that they should try to contribute to the policy process by examining solutions to dynamic optimization problems that take these targets as constraints.

This section reports some preliminary results for concentration thresholds of the sort envisioned by the Framework Convention. The damage side of the modeling structure cannot be ignored in these cost minimization exercises. Figure 20.6 showed how economic activity, and therefore emissions, can be influenced by damage associated with residual climate change even in a policy environment. The objective function will now be discounted economic cost instead of discounted utility, but those costs must include climate costs, and they are measured against economic activity along an unregulated trajectory. Researchers who confront these constrained optimization problems must bring their entire models to the table.

20.4.1 Deterministic results

Figure 20.9 mimics Figure 20.1 in the sense that cost-minimizing emissions trajectories for limiting greenhouse gas concentrations to 550 parts per million in volume (ppmv in equivalent CO_2) are displayed for the same set of models under the same set of standardizing calibrations; 550 ppmv is a popular target for this work simply because it is approximately twice the concentrations that prevailed across the globe before the industrial revolution. Two observations immediately jump out from Figure 20.9. First, and notwithstanding a common inverted U-shape to each trajectory (plus or minus an occasional wiggle in one or two paths), it is obvious that model uncertainty persists in the cost-minimization problem even when different models accept comparable specifications of underlying processes. Secondly, model-to-model comparisons of Figures 20.1 and 20.9 reveal that the cost-minimizing trajectories track their corresponding optimal trajectories over the near term only to diverge significantly after 2030. Fixing concentrations at 550 ppmv would not be economically efficient (for smooth, regular assessments of moderate damages, given modest climate sensitivity, at least). The deterministic optimal solution to the extreme U_1 case described in section 20.3 would limit concentrations to roughly 650 ppmv. That is, economic efficiency would allow atmospheric concentrations to climb 100 ppmv above the popular 550 ppmv even if it were known with certainty that climate sensitivity were 4.8°C and baseline damages at 3°C were 12.5% of gross world product.

20.4.2 A stochastic experiment

The second observation drawn from Figure 20.9 also suggests a second line of inquiry—what sort of emissions control over time would minimize the expected cost of meeting a

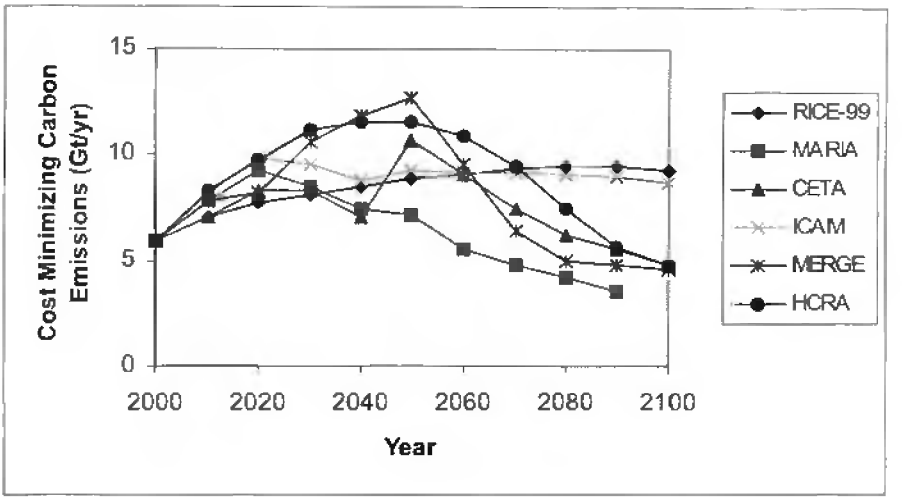


Figure 20.9. *Carbon emissions trajectories that minimize the discounted economic cost (cum climate damage) of limiting atmospheric concentrations of greenhouse gases to the radiative equivalent of 550 parts per million (volume) of carbon. Results from RICE-99 and five other models using common calibrations are displayed.*

concentration threshold that would not be determined until sometime in the future? Yohe and Wallace [47] explored exactly this question in two steps in the spirit of the hedging exercise described in section 20.3. They worked with the emissions scenarios described in section 20.3.1, and the damage side of their model was calibrated to match a 2.5°C increase in global mean temperature with damages equal to 1.6% of gross world product.

Table 20.6 reports the results of their first step—computations of the minimum discounted expected cost of imposing controls that would create least cost deviations from each of the seven representative scenarios for seven different concentration targets until the year 2020. Each target was considered to be equally likely until 2020, when the actual target would be imposed. After 2020, of course, the least-cost control turnpike converged to the control trajectory that would have minimized discounted costs if the actual concentration target had been known in 2000. Costs through 2200 were included in the discounting, but near-term deviations were most influential. Planning as if 850 ppmv would be the target minimized expected costs for scenarios (20.1)–(20.5), preceding as if 800 ppmv were the target was best along scenario (20.6), and 700 ppmv was the least-cost choice along scenario (20.7).

The second step finally takes recognition of one source of uncertainty that has thus far escaped attention even though its significance was highlighted in section 20.2. Specifically, Yohe and Wallace extended the scope of uncertainty through 2020 to include ambiguity in the underlying emissions trajectory through 2020. Table 20.7 displays these results under the assumption that decision makers would make their near-term policy decisions by following a sequential procedure that recognizes the content of Table 20.6. They would first compute, for each scenario, the expected discounted cost across all scenarios and concentration targets that would result from imposing the controls identified in Table 20.6 through 2020 and then

Table 20.6. *The expected discounted value of costs through* 2200 *for alternative concentration targets (percent of gross world product in* 2000; *one percentage point is* $270 *billion in constant year* 2000 *dollars)* [47]. **'s indicate the minimum numbers in each column.*

	Scenario number						
Concentration target (ppmv)	(1)	(20.2)	(20.3)	(20.4)	(20.5)	(20.6)	(20.7)
550	0.00	0.77	2.33	5.69	22.39	26.17	33.31
600	0.00	0.77	2.15	5.66	21.74	25.54	32.77
650	0.00	0.77	2.07	5.66	21.46	25.28	32.52
700	0.00	0.76	2.02	5.64	21.31	25.18	32.49*
750	0.00	0.67	2.00	5.64	21.25	25.17	32.60
800	0.00	0.67	1.98	5.64	21.22	25.16*	32.65
850	0.00*	0.67*	1.97*	5.64*	21.21*	25.18	32.69

Table 20.7. *The expected discounted value of costs through* 2200 *for alternative concentration targets across uncertain scenarios (percent of gross world product in* 2000; *one percentage point is* $270 *billion in constant year* 2000 *dollars)* [47].

Scenario	Assumed near-term target	550	600	650	700	750	800	850
1	850	16.51	12.99	5.94	4.02	2.84	2.42	1.59
2	850	16.51	12.99	5.94	4.02	2.84	2.42	1.59
3	850	16.40	12.96	6.06	4.17	3.02	2.61	1.79
4	850	16.69	13.16	6.10	4.20	3.06	2.65	1.82
5	850	16.50	13.15	6.45	4.61	3.49	3.08	2.27
6	800	16.43	13.16	6.61	4.79	3.68	3.28	2.47
7	700	16.40	13.30	7.09	5.32	4.24	3.84	3.05

optimally converging to the policy that would have been correct if they had known the scenario and the target.

The results of Table 20.6 provided seven (actually six, because near-term controls along scenarios (20.1) and (20.2) were identical) alternative control trajectories through 2020 that can be interpreted as least-cost hedging strategies across concentration targets until all uncertainty is resolved.

Table 20.7 reveals that planning for 850 ppmv as if scenario (20.1) or (20.2) would materialize (by imposing *no emissions controls through* 2020) would minimize overall discounted expected cost across scenarios and targets. This strategy is remarkably robust across alternative concentration targets. Indeed, a different near-term plan would be more cost effective only if the concentration target turned out to be less than 600 ppmv. Modest controls minimized overall expected costs if all seven representative emissions scenarios were taken to be equally likely.

20.5 Concluding remarks

This discussion has focused attention on dynamic optimization in the climate arena with and without acknowledgement of stochastic elements, and it has offered one example of how the same techniques might be applied to a second-best question. A long series of other second-best questions have been examined in like manner. Weyant [42] reviews EMF experiments that capture the strict short-term limits of the Kyoto Protocol under a variety of assumptions about sinks and the structure of accounting frameworks for other greenhouse gases. Nordhaus and Boyer [31] also devote a chapter to these questions. Parallel research initiatives have addressed cost-minimizing trajectories for emissions and control rates for specified and/or uncertain temperature targets. Perhaps the most involved of these applications contemplates the ramifications of adopting a tolerable windows approach where limits are enumerated in terms of both temperature increases and the pace of temperature change (see, for example, [40]). Still other researchers, like Lempert and Schlesinger [17, 18] and Lempert et al. [19], have examined the relative efficacy of contingent policy strategies. Modeled conceptually on the way the Federal Reserve System modulates the rate of growth of the money supply, these studies depend critically on divining how policy makers might separate signal from noise in the climate record, and so they highlight questions about exactly what to monitor and more generally about how international institutions might be created to administer midcourse corrections. Much as the perfectly competitive model in economics serves as an efficiency benchmark against which to measure the economic cost of market distortions and other market imperfections in traditional economic theory, the dynamic efficiency of deterministic solutions and/or stylized hedging strategies of the sort described above serve as benchmarks in all these studies against which to measure relative efficacy of alternative policy designs. It is in this benchmarking capacity that they display their largest utility.

Bibliography

[1] H. Dowlatabadi and M. G. Morgan, 1995, *Integrated Assessment Climate Assessment Model* 2.0, Technical Discussion, Department of Engineering and Public Policy, Carnegie Mellon University, Pittsburgh, PA.

[2] Energy Modeling Forum, *Second Round Scenario Results*, Stanford University, Stanford, CA, 1996.

[3] S. Fankhauser, R. S. J. Tol, and D. W. Pearce, *The aggregation of climate change damages: A welfare theoretic approach*, Environ. Resource Econ., 10 (1997), pp. 249–266.

[4] D. Gaskins and J. Weyant, *EMF-12: modeling comparisons of the costs of reducing CO_2 emissions*, Amer. Econ. Rev., 83 (1993), pp. 318–323.

[5] J. K. Hammit, R. J. Lempert, and M. E. Schlesinger, *A sequential decision-strategy for abating climate change*, Nature, 357 (1992), pp. 315–318.

[6] H. Hotelling, *The economics of exhaustible resources*, J. Political Econ., 39 (1931), pp. 137–175.

[7] IPCC, *Climate Change: The Supplementary Report to the IPCC Scientific Assessment*, Cambridge University Press, Cambridge, UK, 1992.

[8] IPCC, *The Science of Climate Change, Contribution of Working Group* I *to the Second Scientific Assessment of the Intergovernmental Panel on Climate Change*, Cambridge University Press, Cambridge, UK, 1996.

[9] IPCC, *Impacts, Adaptation and Mitigation of Climate Change: Scientific-Technical Analyses, Contribution of Working Group* II *to the Second Scientific Assessment of the Intergovernmental Panel on Climate Change*, Cambridge University Press, Cambridge, UK, 1996.

[10] IPCC, *Climate Change* 1995*—Economic and Social Dimensions of Climate Change, Contribution of Working Group* III *to the Second Scientific Assessment of the Intergovernmental Panel on Climate Change*, Cambridge University Press, Cambridge, UK, 1996.

[11] IPCC, *The Science of Climate Change—The Contribution of Working Group* I *to the Third Scientific Assessment of the Intergovernmental Panel on Climate Change*, Cambridge University Press, Cambridge, UK, 2001.

[12] IPCC, *Impacts, Adaptation, and Vulnerability—The Contribution of Working Group* II *to the Third Scientific Assessment of the Intergovernmental Panel on Climate Change*, Cambridge University Press, Cambridge, UK, 2001.

[13] IPCC, *Mitigation—The Contribution of Working Group* III *to the Third Scientific Assessment of the Intergovernmental Panel on Climate Change*, Cambridge University Press, Cambridge, UK, 2001.

[14] F. Joos, M. Bruno, R. Fink, U. Siegenthaler, T. F. Stocker, C. L. Quere, and J. L. Sarmiento, *An efficient and accurate representation of complex oceanic and biospheric models of anthropogenic carbon uptake*, Tellus, 48 (1996), pp. 397–417.

[15] R. K. Kaufmann, *Assessing the DICE model: Uncertainty associated with the emission and retention of greenhouse gases*, Climatic Change, 35 (1997), pp. 435–448.

[16] J. F. Kasting and C. G. Walker, *The geochemical carbon cycle and the uptake of fossil fuel* CO_2, in Global Warming: Physics and Facts, B. G. Levi, D. Hafemeister, and R. Scribner, eds., Amer. Inst. Phys. Conf. Proc. 247, 1992, pp. 175-200.

[17] R. J. Lempert and M. E. Schlesinger, *Robust strategies for abating climate change*, Climatic Change, 45 (2000), pp. 387–401.

[18] R. J. Lempert, M. E. Schlesinger, and S. C. Bankes, *When we don't know the costs or the benefits: Adaptive strategies for abating climate change*, Climatic Change, 33 (1996), pp. 235–274.

[19] R. J. Lempert, M. E. Schlesinger, S. C. Bankes, and H. G. Andronova, *The impact of variability on near-term climate change policy choices and the value of information*, Climatic Change, 45 (2000), pp. 129–161.

[20] E. MAIER-REIMER AND K. HASSELMAN, *Transport and storage of carbon dioxide in the ocean, and an organic ocean-circulation carbon cycle model*, Climate Dynamics, 2 (1987), pp. 63–90.

[21] A. S. MANNE, *A Summary of Poll Results—EMF 14 Subgroup on Analysis for Decisions under Uncertainty*, Stanford University, Stanford, CA, 1995.

[22] A. MANNE, R. MENDELSON, AND R. RICHELS, *MERGE: A model for evaluating regional and global effects of GHG reduction policies*, Energy Policy, 23 (1992), pp. 17–34.

[23] A. MANNE AND R. RICHELS, *Buying Greenhouse Insurance: The Economic Costs of CO_2 Emissions*, MIT Press, Cambridge, MA, 1992.

[24] A. MANNE AND R. RICHELS, *The costs of stabilizing global CO_2 emissions: A probabilistic analysis based on expert judgements*, Energy J., 15 (1994), pp. 31–56.

[25] M. G. MORGAN AND D. KEITH, *Subjective judgements by climate experts*, Environ. Sci. Technol., 29 (1994), pp. 468–476.

[26] T. MORITA, Y. MATSUOKA, K. JIANG, T. MASUI, K. TAKAHASHI, M. KAINUMA, AND R. PANDEY, *Quantification of IPCC-SRES Storylines Using the AIM/Emission-Linkage Model*, National Institute for Environmental Studies, Japan, 1998.

[27] G. MYHRE, E. J. HIGHWOOD, K. P. SHINE, AND F. STORDAL, *New estimates of radiative forcing due to well mixed greenhouse gases*, Geophys. Res. Lett., 25 (1998), pp. 2715–2718.

[28] W. D. NORDHAUS, *To slow or not to slow: The economics of the greenhouse effect*, Economic J., 101 (1991), pp. 920–937.

[29] W. D. NORDHAUS, *Expert opinion on climatic change*, American Sci., 82 (1994), pp. 45–51.

[30] W. D. NORDHAUS, *Managing the Global Commons: The Economics of Climate Change*, MIT Press, Cambridge, MA, 1994.

[31] W. D. NORDHAUS AND J. BOYER, *Warming the World: Economics Models of Climate Change*, MIT Press, Cambridge, MA, 2000.

[32] S. PECK AND T. TEISBERG, *CETA: A model for carbon emissions trajectory assessment*, Energy J., 13 (1992), pp. 55–77.

[33] S. PECK AND T. TEISBERG, *Optimal CO_2 control policy with stochastic losses from temperature rise*, Climatic Change, 31 (1995), pp. 19–34.

[34] T. ROUGHGARDEN AND S. SCHNEIDER, *Climate change policy: Quantifying uncertainties for damages and optimal carbon taxes*, Energy J., 27 (1999), pp. 415–427.

[35] M. E. SCHLESINGER AND X. JIANG, *Simple model representation of atmoshpheric-ocean GCMs and estimation in the timescale of CO_2-induced climate change*, J. Climate, 1 (1990), pp. 12–15.

[36] P. SCHULTZ AND J. F. KASTING, *Optimal reductions in CO_2 emissions*, Energy Policy, 25 (1997), pp. 491–500.

[37] S. SCHNEIDER, *The changing climate*, in Managing the Planet—Readings from Scientific American, W. H. Freeman, New York, 1989, pp. 25–39.

[38] R. S. J. TOL, *Is the uncertainty about climate change too large for expected cost-benefit analysis?*, Climatic Change, 56 (2002), pp. 265–289.

[39] R. S. J. TOL, *New Estimates of the Damage Costs of Climate Change, Parts* I *and* II, Working Papers D99-01 and D99-02, Institute for Environmental Studies, Vrije Universiteit, Amsterdam, 1999.

[40] F. L. TOTH, T. BRUCKNER, H. M. FUSSEL, M. LEIMBACH, G. PETSCHEL-HELD, AND J. J. SCHELLNHUBER, *The tolerable windows approach to integrated assessments*, in Climate Change and Integrated Assessment Models—Bridging the Gap, O. K. Cameron, K. Fukuwatari, and T. Morita, eds., Center for Global Environmental Research, Environmental Agency of Japan, Tsukuba, Japan, 1998.

[41] J. WEYANT, *Incorporating uncertainty in integrated assessment of climate change*, in Elements of Climate Change 1996, Aspen Global Change Institute, Aspen, CO, 1997, pp. 268–283.

[42] J. WEYANT, ED., *The Costs of the Kyoto Protocol: A Multi-Model Evaluation*, Energy J. (special issue), 1999.

[43] G. YOHE, *Selecting "interesting" scenarios with which to analyze policy-responses to potential climate change*, Climate Res., 1 (1991), pp. 169–177.

[44] G. YOHE, *Exercises in hedging against extreme consequences of global change and the expected value of information*, Global Environ. Change, 6 (1996), pp. 87–101.

[45] G. YOHE, *More trouble for cost-benefit analysis*, Climatic Change, 56 (2002), pp. 235–244.

[46] G. YOHE AND B. GARVEY, *Incorporating uncertainty and non-linearity into the calculus of an efficient response to the threat of global warming*, Internat. J. Global Energy Issues, 7 (1995), pp. 34–47.

[47] G. YOHE AND R. WALLACE, *Near term mitigation policy for global change under uncertainty: Minimizing the expected cost of meeting unknown concentration thresholds*, Environ. Modeling Assessment, 1 (1996), pp. 47–57.

[48] G. YOHE AND M. E. SCHLESINGER, *The economic geography of the impacts of climate change*, J. Econ. Geography, 2 (2002), pp. 311–341.

This page intentionally left blank

Chapter 21

Groundwater Pollution Control

*David W. Watkins, Jr.,** *Daene C. McKinney,*† *and David P. Morton*‡

21.1 Introduction

Groundwater is an important source of potable water because it is abundant and readily available in many locations and often requires little or no treatment. In 1995, groundwater accounted for approximately 20% of potable water use in the United States, and approximately 50% of the U.S. population relied on groundwater for their source of drinking water. In most European countries, groundwater accounts for 10% to 50% of potable water use [1].

Unfortunately, various human activities have resulted in the overuse or degradation of many groundwater resource systems. Overpumping (or groundwater mining, the extraction of groundwater at rates higher than natural recharge rates) has led to increased pumping costs, land subsidence, saltwater intrusion into freshwater aquifers, and limited ability to achieve sustainable social and economic systems. Large-scale groundwater pollution has resulted from the use of agricultural chemicals, and localized pollution has resulted from industrial discharges, improper hazardous waste disposal, landfill seepage, and leaky underground storage tanks.

Since the management of a groundwater system can be a complex task, a systems analysis framework is frequently used to address groundwater pollution problems. In particular, numerical groundwater simulation models are now commonly used by engineers and scientists to address a wide range of problems involving water supply management,

*Department of Civil and Environmental Engineering, Michigan Technological University, Houghton, MI 49931 (dwatkins@mtu.edu).

†Department of Civil Engineering, The University of Texas at Austin, Austin, TX 78712 (daene_mckinney @mail.utexas.edu).

‡Graduate Program in Operations Research, The University of Texas at Austin, Austin, TX 78712 (morton @mail.utexas.edu).

pollution control, and ecosystem protection or restoration. These models are capable of simulating groundwater flow and contaminant transport and predicting the impact of human stresses—pumping or recharge modifications—on the groundwater system. Inputs to these models include the extent of the groundwater system to be modeled, boundary conditions, a discretization scheme, and model parameters describing the transmission and storage capabilities of the aquifer. Model outputs typically include groundwater levels (or hydraulic pressures), flow velocities, and contaminant concentrations at various locations in the aquifer.

In designing a solution to a groundwater management problem, numerical simulation models are typically used in a repetitive manner to evaluate various alternatives and scenarios and to select an alternative that meets constraints and best achieves one or more objectives. As a complement to simulation models, groundwater optimization models are used to directly consider management objectives and various policy constraints. Typical decision variables in these models are pumping-well locations and pumping rates.

Although deterministic optimization models have proven useful in the preliminary design of groundwater remediation systems (e.g., [2]), the variability and complexity of the subsurface environment motivates the application of stochastic programming models to select design alternatives which hedge against uncertainty. In general, stochastic programming models for groundwater management (1) seek a low-cost, low-risk design; (2) consider uncertainty and imperfect predictive capability due to incomplete knowledge of the subsurface environment; (3) primarily address predictive uncertainty due to spatially variable aquifer parameters through the use of geostatistical methods; and (4) invoke the concept of reliability in quantifying the economic impacts of uncertainty [8].

In this chapter, we consider the problem of hydraulically containing a groundwater contaminant plume in the presence of hydrogeologic uncertainty. A brief introduction to the equations of groundwater flow, numerical simulation modeling, and geostatistics is first presented. A deterministic optimization model for hydraulic control is formulated, and this model is then extended to a two-stage optimization model to incorporate uncertainty. The two-stage model is solved approximately using a sampling-based technique that provides confidence intervals on the optimality gap with respect to a candidate solution. Trade-offs among expected cost, expected cost overruns, and reliability are investigated.

For additional background on groundwater flow management, see [1]. For more detailed discussions of groundwater management modeling under uncertainty, see [31, 24, 8].

21.2 Groundwater flow equations

Groundwater occupies the void space of subsurface formations, with voids ranging in size from tiny pores to large openings such as caverns. Connected pores act as conduits for fluid flow driven by gravity, pressure, or surface tension forces. Subsurface formations containing useful quantities of groundwater are called aquifers, and these formations consist of unconsolidated rocks (mainly sands and gravels) and are usually of large areal extent. In general, aquifers are classified as confined or unconfined, depending on the presence or absence of an overlying, relatively impermeable formation that confines the groundwater under pressure. These types of aquifers are illustrated in Figure 21.1.

Aquifers perform two important functions. They store water, acting as reservoirs, and

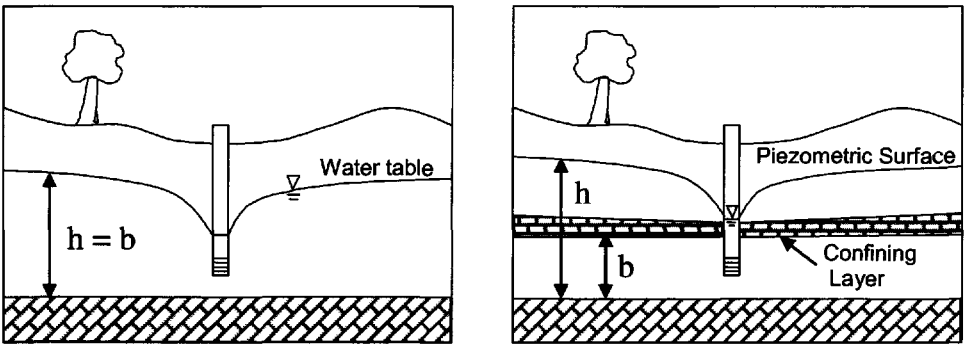


Figure 21.1. *Subsurface distribution of water.* (a) *Unconfined (phreatic) aquifer;* (b) *confined (artesian) aquifer.*

they transmit water, acting as pipelines. The parameters that describe an aquifer's ability to store and transmit water are storativity, S, and hydraulic conductivity, K, respectively.

Storativity, S, represents the change in the volume of water stored in a formation due to a change in the hydraulic pressure in the formation. It is the volume of water released (taken up) from a vertical column of the aquifer with a unit cross-sectional area per unit decline (rise) in the hydraulic head—the sum of elevation and pressure expressed in units of length. In a confined aquifer, the compressibility of the water in the pore space and the compressibility of the solid matrix determine the storage capabilities of the aquifer. In an unconfined aquifer, pore space drainage is the dominant factor determining the storage characteristics of the aquifer. In this case, a certain amount of water will be retained in the pores due to surface tension (capillary action) and molecular forces between the water molecules and the solid matrix. Thus, in an unconfined aquifer, the storativity is slightly less than the percent of total pore volume. The storativity of an unconfined aquifer is often called the specific yield. Due to compressibility effects, the storativity of a confined aquifer ($S \approx 10^{-5}$) is much less than the specific yield of an unconfined aquifer ($S \approx 10^{-1}$).

Hydraulic conductivity, K, represents the ability of a subsurface formation to transmit water under a gradient in hydraulic pressure. With units of length per time $[L/T]$, K is the flow rate of water through a unit cross-sectional area of the aquifer under a unit hydraulic gradient, as defined by a relationship known as Darcy's law. Darcy [6] published a study of the design of sand filters for the city of Lyon, France. He found that the flow rate, Q $[L^3/T]$, through a vertical sand filter was proportional to (1) the cross-sectional area of the filter, A $[L^2]$, and (2) the decrease in hydraulic head through the filter, Δh $[L]$, and inversely proportional to the length of the flow path, L $[L]$, or

$$\frac{Q}{A} = -K\frac{\Delta h}{L},$$

where K, the hydraulic conductivity, is the constant of proportionality. The average velocity

through the area of porous medium is

$$V = \frac{Q}{nA},$$

where n is the percent of total pore volume. The units of hydraulic conductivity are typically reported in m/s or gal/(day-ft^2), where 1 gal/(day-ft^2) $= 4.72 \times 10^{-7}$ m/s. Values of K have a wide range for various types of porous media, being on the order of 1×10^{-2} m/s for gravels and 1×10^{-10} m/s for clays.

Combining Darcy's law with conservation of mass, and defining transmissivity, T, $[L^2/T]$ as Kb, where b is the thickness of the aquifer, the governing partial differential equation (PDE) for two-dimensional flow in a confined aquifer is

$$\frac{\partial(-Q^x)}{\partial x} + \frac{\partial(-Q^y)}{\partial y} = \frac{\partial}{\partial x}\left(T^x \frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(T^y \frac{\partial h}{\partial y}\right) - q = S\frac{\partial h}{\partial t}. \tag{21.1}$$

Here, T^x and T^y are transmissivities and Q^x and Q^y are the flow rates in the x and y directions, h is the hydraulic head $[L]$, and q is the volumetric pumping or extraction rate per unit area $[L/T]$. Analytical solutions of this PDE are difficult for aquifers with irregular boundaries, various boundary conditions, and heterogeneous and anisotropic porous media properties, and most aquifers exhibit all of these characteristics. Fortunately, numerical solutions to the governing PDE provide a useful and convenient method of handling these complications.

The finite-difference method replaces the governing PDE by a numerical approximation. Specifically, the continuous derivatives of the PDE are replaced by discrete approximations. The result of this discretization is a set of simultaneous equations that must be solved for the values of the unknown variables at discrete locations in the modeled domain. To accomplish the discretization process, a mesh or grid must be defined that covers the domain. The grid consists of a series of intersecting, orthogonal, straight lines, as illustrated in Figure 21.2. We solve for the unknown state variables at the center of each grid block.

We assume steady state flow, i.e., $\partial h/\partial t = 0$. Then, applying a finite-difference approximation to the outer derivatives in (21.1) yields

$$\frac{\left(T^x \frac{\partial h}{\partial x}\right)_{i+\frac{1}{2},j} - \left(T^x \frac{\partial h}{\partial x}\right)_{i-\frac{1}{2},j}}{\Delta x} + \frac{\left(T^y \frac{\partial h}{\partial y}\right)_{i,j+\frac{1}{2}} - \left(T^y \frac{\partial h}{\partial y}\right)_{i,j-\frac{1}{2}}}{\Delta y} - q_{i,j} = 0.$$

Applying a finite-difference approximation to the remaining derivatives yields

$$\frac{\left(T^x_{i+\frac{1}{2},j}\frac{h_{i+1,j}-h_{i,j}}{\Delta x}\right) - \left(T^x_{i-\frac{1}{2},j}\frac{h_{i,j}-h_{i-1,j}}{\Delta x}\right)}{\Delta x} + \frac{\left(T^y_{i,j+\frac{1}{2}}\frac{h_{i,j+1}-h_{i,j}}{\Delta y}\right) - \left(T^y_{i,j-\frac{1}{2}}\frac{h_{i,j}-h_{i,j-1}}{\Delta y}\right)}{\Delta y} - q_{i,j} = 0, \tag{21.2}$$

where

$$T^x_{i+\frac{1}{2},j} = 2\frac{T^x_{i+1,j}T^x_{i,j}}{T^x_{i+1,j} + T^x_{i,j}} \quad \text{and} \quad T^y_{i,j+\frac{1}{2}} = 2\frac{T^y_{i,j+1}T^y_{i,j}}{T^y_{i,j+1} + T^y_{i,j}}$$

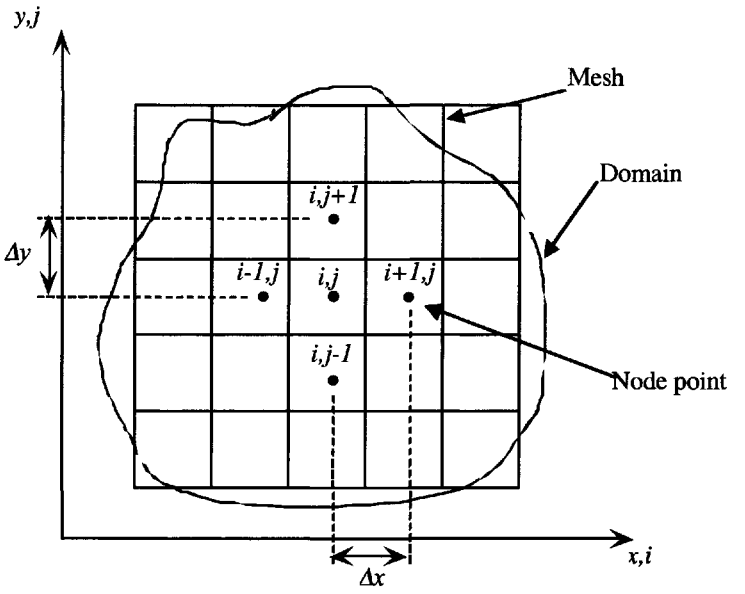


Figure 21.2. *Finite-difference mesh.*

are, respectively, the harmonic averages of the x-direction transmissivities between cells (i, j) and $(i + 1, j)$ and the y-direction transmissivities between cells (i, j) and $(i, j + 1)$. We rewrite (21.2) as

$$\begin{aligned} &A_{i,j}\left(h_{i+1,j} - h_{i,j}\right) + B_{i,j}\left(h_{i-1,j} - h_{i,j}\right) \\ &+ C_{i,j}\left(h_{i,j+1} - h_{i,j}\right) + D_{i,j}\left(h_{i,j-1} - h_{i,j}\right) - q_{i,j} = 0, \end{aligned} \tag{21.3}$$

where

$$A_{ij} = \frac{2}{\Delta x^2}\frac{T^x_{i+1,j}T^x_{i,j}}{T^x_{i+1,j} + T^x_{i,j}}, \quad B_{ij} = \frac{2}{\Delta x^2}\frac{T^x_{i-1,j}T^x_{i,j}}{T^x_{i-1,j} + T^x_{i,j}},$$

$$C_{ij} = \frac{2}{\Delta y^2}\frac{T^y_{i,j+1}T^y_{i,j}}{T^y_{i,j+1} + T^y_{i,j}}, \quad D_{ij} = \frac{2}{\Delta y^2}\frac{T^y_{i,j-1}T^y_{i,j}}{T^y_{i,j-1} + T^y_{i,j}}.$$

The finite-difference approximation yields an equation of the form (21.3) for each node (i, j) on the interior of the solution domain. Solution of these equations, however, requires complete specification of boundary conditions. Two types of boundary conditions are typically encountered in groundwater flow problems: constant or prescribed head conditions, and prescribed flux conditions.

Constant-head boundary conditions over the domain C_1 in a steady state system are

$$h(x, y) = f_1(x, y), \quad (x, y) \in C_1 \tag{21.4}$$

for a specified function f_1. For example, the finite-difference equation (21.3) for node (1, 1) at the lower left-hand corner of the modeled domain (see Figure 21.2) includes two terms, $h_{0,1}$ and $h_{1,0}$, that are known from the boundary conditions.

Prescribed flux conditions over the domain C_2 in a steady state system are represented as

$$\frac{\partial h(x, y)}{\partial x} = f_2(x, y) \quad \text{and} \quad \frac{\partial h(x, y)}{\partial y} = f_2(x, y), \quad (x, y) \in C_2. \tag{21.5}$$

While many models assume that the aquifer is homogeneous and isotropic, with physical parameters that are independent of spatial location and direction, most subsurface formations are extremely heterogeneous and anisotropic [7, 5]. Aquifers are typically composed of unconsolidated geologic deposits of complex arrays of lenses or strata of essentially unknown geometry and variable hydraulic properties. The degree of heterogeneity and the spatial structure of the hydraulic properties have a large effect on the flow and mass transport characteristics of the aquifer. In cases of predictable heterogeneity, the conventional approach to groundwater modeling is to subdivide the formation into a number of homogeneous subzones, each with a different equivalent value for the parameter of interest.

From available data, it is expected that groundwater velocity in many formations will vary irregularly over scales of approximately 1 to 10 cm in the vertical direction and 1 to 2 m in the horizontal direction. Since it would be impractical to make detailed measurements over the scale of hundreds or thousands of meters, the complexity of most common groundwater systems has led to the consideration of the physical properties of the aquifer as spatial stochastic processes or spatial random functions with random hydraulic parameters [5, 10]. Average macroscopic or "equivalent" parameters are derived for making flow calculations on a large scale. In practice, this problem is handled by modeling the unknown parameters (here, hydraulic conductivity) as a spatial random field. In our work the hydraulic conductivity is modeled as a multivariate lognormal distribution. The parameters of this distribution are estimated from measured data with the spatial persistence of the random field being captured via the covariance parameters.

21.3 Optimization modeling for hydraulic control

We address the problem of hydraulically controlling a groundwater contaminant plume threatening drinking water supplies or other aquatic resources by installing and operating one or more pumping wells. These pumping wells are selected to maintain an inward hydraulic gradient to prevent migration of the plume. A typical optimization model of this type selects the least-cost set of well locations and pumping rates that will maintain the required hydraulic gradients at one or more monitoring locations. Figure 21.3 illustrates a hypothetical contaminant plume, along with a number of potential well and gradient monitoring locations. Assuming perfect knowledge of hydraulic conductivity, our deterministic optimization model along these lines is formulated as follows.

Indices and sets

$i \in I$ two-dimensional groundwater model cells,

IF subset of boundary cells with specified (fixed) head,

IN subset of boundary cells where no flow is allowed,

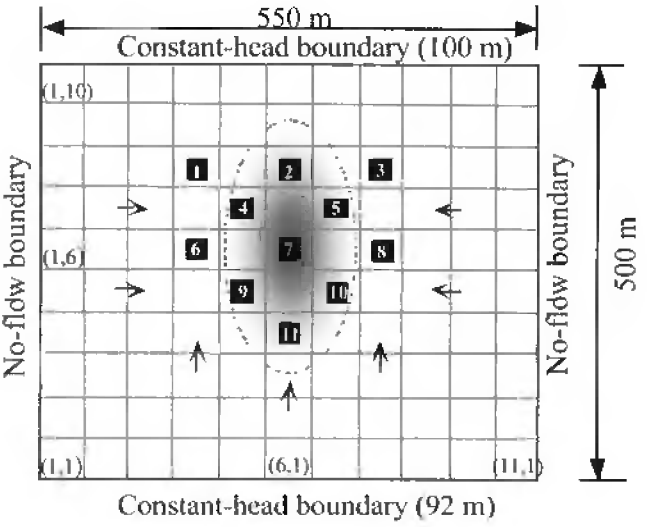


Figure 21.3. *Finite-difference grid, hypothetical contaminant plume, potential well locations* (■).

IA subset of active (nonboundary) cells,

$IW \subset IA$ subset of cells where a well can be built,

IC subset of cells on the capture curve,

N_i set of neighboring cells of cell i,

C_i set of neighboring cells of $i \in IC$ with which a specified head-gradient is required.

Data

B maximum number of wells that can be installed,

PC unit pumping cost [$\$T/L^2$],

K_i capital cost of installing a well at site i [\$],

PU_i maximum pumping rate of a well at site i [$\$/L$],

HB_i fixed value of head at boundary cell $i \in IF$ [L],

$TR_{ij} \equiv TR_{ji}$ transmissivity coefficient [$1/T$] between cells i and j,

EL_i land surface elevation at i [L],

TC target installation and operating cost [\$],

λ_c weight on penalty for exceeding target cost.

Decision variables

x_i takes value 1 if a well is built at site i and 0 otherwise,

q_i pumping from cell i $[L/T]$,

h_i aquifer head at cell i $[L]$,

v amount by which target cost is exceeded [\$].

Boundary conditions

$q_i \equiv 0, i \notin IW$, cannot pump in i if a well cannot be built in i,

$h_i \equiv BH_i, i \in IF$, fix head in i to boundary head value in i.

$$\min_{x,q,h,v\geq 0} \sum_{i\in IW} K_i x_i + PC \sum_{i\in IW} (EL_i - h_i)q_i + \lambda_c v \tag{21.6a}$$

$$\text{s.t.} \quad \sum_{i\in IW} x_i \leq B, \tag{21.6b}$$

$$q_i \leq PU_i x_i \quad \forall i \in IW, \tag{21.6c}$$

$$\sum_{j\in N_i} TR_{ij}(h_j - h_i) - q_i = 0 \quad \forall i \in IA, \tag{21.6d}$$

$$h_i \leq h_j \quad \forall i \in IC, \quad j \in C_i \tag{21.6e}$$

$$h_i = h_j \quad \forall i \in IN, \quad j \in N_i, \tag{21.6f}$$

$$v \geq \sum_{i\in IW} K_i x_i + PC \sum_{i\in IW} (EL_i - h_i)q_i - TC, \tag{21.6g}$$

$$x_i \in \{0, 1\} \quad \forall i \in IW.$$

The objective function (21.6a) minimizes a weighted sum of installation and pumping costs and costs exceeding the target cost. This function is nonlinear due to the consideration of pumping costs which are proportional to the height the water must be lifted. Constraint (21.6b) limits the number of installed wells, and (21.6c) limits the pumping rate in cell i if a well is built there and prevents pumping if not. Constraints (21.6d) provide for groundwater flow continuity and correspond to the finite-difference equations (21.3). The groundwater contaminant plume is contained by the hydraulic-gradient constraints (21.6e). Constraints (21.6f) enforce equal heads at boundary cells where no flow is permitted and are an example of a prescribed flux condition (21.5). Similarly, the boundary conditions, $h_i \equiv BH_i, i \in IF$, are an example of (21.4). Constraint (21.6g) defines the target-cost exceedance variable v.

We more compactly enumerate the cells of the form (i, j) from section 21.2 with a single index $i \in I$ and notate the "up-down-left-right" neighborhood structure of a cell via N_i. With this specification of N_i, the transmissivity coefficients TR_{ij} in (21.6d) correspond to the A, B, C, and D coefficients in (21.3) for the four neighbors of cell i.

In the presence of geologic uncertainty (e.g., uncertain hydraulic conductivity), a solution found using this deterministic optimization model is likely to be optimistic (i.e.,

low cost but not reliable). On the other hand, a solution based on a worst-case scenario may be overly pessimistic (i.e., overdesigned). In reality, there is a trade-off between pumping cost and the reliability of plume containment, and multiobjective stochastic programming provides a convenient means of generating a trade-off curve that can be viewed by a decision maker in selecting the desired level of reliability.

The stochastic programming model we develop is based on a zonal and geostatistical description of hydraulic conductivity [25, 26]. This model seeks a low-cost, reliable design that hedges against the possible realizations of the random hydraulic conductivity field and is given as follows:

$$w^* = \min_x \sum_{i \in IW} K_i x_i + Ef(x, \boldsymbol{TR}) \tag{21.7a}$$

$$\text{s.t.} \sum_{i \in IW} x_i \leq B, \tag{21.7b}$$

$$x_i \in \{0, 1\} \quad \forall i \in IW, \tag{21.7c}$$

where

$$f(x, \boldsymbol{TR}) = \min_{q,h,z,v \geq 0} \quad PC \sum_{i \in IW} (EL_i - h_i) q_i + \lambda_c v + \lambda_e \sum_{i \in I} z_i^2 \tag{21.8a}$$

$$\text{s.t.} \quad q_i \leq PU_i x_i \quad \forall i \in IW, \tag{21.8b}$$

$$\sum_{j \in N_i} \boldsymbol{TR}_{ij} (h_j - h_i) = q_i \quad \forall i \in IA, \tag{21.8c}$$

$$h_i \leq h_j + z_j \quad \forall i \in IC, \quad j \in C_i, \tag{21.8d}$$

$$h_i = h_j \quad \forall i \in IN, \quad j \in N_i, \tag{21.8e}$$

$$v \geq \sum_{i \in IW} K_i x_i + PC \sum_{i \in IW} (EL_i - h_i) q_i - TC. \tag{21.8f}$$

Since there may be no design that satisfies all the plume containment constraints for all scenarios—and it may not even be desirable to do so—the hydraulic-gradient constraints (21.8d) are relaxed from their analogs in the deterministic formulation by allowing a violation to occur via decision variables z_i, $i \in I$. Squared violations are penalized in the objective with an additional environmental weighting factor λ_e [27]. In the stochastic setting, realizations of cost that exceed the target TC are penalized to incorporate a measure of the risk of cost overruns. This provides a piecewise linear disutility function, as shown in Figure 21.4. The three objectives are combined using a weighting method for multiobjective programming in the spirit of other related work in stochastic programming (e.g., [19]).

This stochastic programming formulation is rooted in two main assumptions. First, the spatial distribution of the hydraulic conductivity is considered a random field that has a known probability distribution. Second, the design problem is represented by a two-stage decision process: the well locations must be selected in the first stage, but pumping rates can be adjusted in the second stage as more information about the hydraulic conductivity field becomes available, perhaps through additional conductivity measurements or observed groundwater heads. In reality, complete information on the subsurface environment cannot be obtained, but the two-stage model is deemed appropriate for selecting promising well locations.

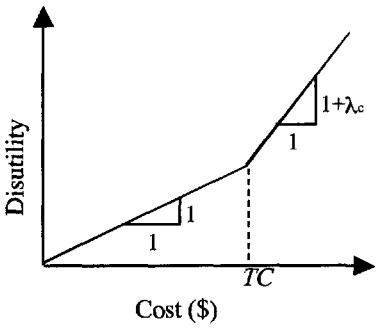


Figure 21.4. *Piecewise linear disutility function, representing risk aversion to cost overruns.*

A more sophisticated multistage model would involve sequentially selecting well sites and conductivity-measurement sites. In such a setting, the probability distribution governing hydraulic conductivity would depend on these earlier decisions. General models of this type are difficult to analyze and solve, but for work along these lines, see, e.g., [3, 9, 13].

The nonlinear program (21.8) is not a convex program because of bilinear terms in the pumping rates, q_i, and head values, h_i. That said, (21.8) can be reformulated as a convex program by replacing (21.8c) with

$$\sum_{j \in N_i} \boldsymbol{TR}_{ij}(h_j - h_i) \geq 0 \quad \forall i \in IW, \tag{21.9a}$$

$$\sum_{j \in N_i} \boldsymbol{TR}_{ij}(h_j - h_i) = 0 \quad \forall i \in IA \setminus IW \tag{21.9b}$$

and substituting $\sum_{j \in N_i} \boldsymbol{TR}_{ij}(h_j - h_i)$ for q_i, $i \in IW$, in the objective function (21.8a) and in constraint (21.8f). The resulting mathematical program is convex if $F(h)$ is convex, where

$$F(h) = \sum_{i \in IA} (EL_i - h_i) \sum_{j \in N_i} TR_{ij}(h_j - h_i). \tag{21.10}$$

The Hessian of F has diagonal terms, $[\nabla^2 F(h)]_{ii} = 2\sum_{j \in N_i} TR_{ij}$, and nonzero off-diagonal terms, $[\nabla^2 F(h)]_{ij} = [\nabla^2 F(h)]_{ji} = -2TR_{ij}$, $j \in N_i$. All the transmissivity coefficients satisfy $TR_{ij} > 0$, $i \in IA$, $j \in N_i$, and so the Hessian has positive diagonal elements and is diagonally dominant, i.e., $[\nabla^2 F(h)]_{ii} \geq \sum_{j \neq i} |[\nabla^2 F(h)]_{ij}|$. It follows from Gershgorin's circle theorem (e.g., [11, Theorem 7.2-1]) that $\nabla^2 F(h)$ has nonnegative eigenvalues, and thus, $F(h)$ is a convex function and the reformulated (21.8) is a convex program.

21.4 Solution method

We cannot exactly solve the stochastic program (21.7) under the continuous distribution that governs the random transmissivity field, $\boldsymbol{TR}$. Instead, we generate independent and

identically distributed (i.i.d.) observations $\boldsymbol{TR}^1, \ldots, \boldsymbol{TR}^n$, from the distribution of $\boldsymbol{TR}$ and solve the associated approximating problem. Denote the former "true" problem (GPC) and the latter approximating problem (GPC_n). (GPC_n) is a large-scale mixed-integer nonlinear program (MINLP). For fixed values of the binary well-location variables, the operations subproblem separates into a convex quadratic program for each of the n scenarios. As a result, we can solve (GPC_n) using the stochastic extension of generalized Benders decomposition described in [28]. The principal advantage of such a decomposition method, over general-purpose solution algorithms, is that we can computationally handle (GPC_n) with larger numbers of scenarios n.

Another solution algorithm for (GPC_n) is the piecewise-quadratic L-shaped method of [15]. Alternatively, the sampling-based branch-and-bound method applied to a groundwater management model in [14] could be applied directly to (GPC). Lucero in [16] exploits special structure in the numerical solution of the nonlinear operations subproblems of a stochastic groundwater remediation model, employing the progressive hedging algorithm [21] in a parallel environment.

Solving (GPC_n) yields a candidate design decision for well locations, say $\hat{x}$. A number of analysts have solved analogous problems with a modest number of scenarios and then applied postoptimality Monte Carlo simulation to assess the actual reliability of the candidate design (e.g., [23, 26, 22]). This approach provides valuable information but does not provide a statement about the quality of $\hat{x}$ relative to an optimal solution of (GPC). To assess the quality of $\hat{x}$ we use the procedure of [17], which constructs a one-sided confidence interval on the optimality gap, $K\hat{x} + Ef(\hat{x}, \boldsymbol{TR}) - w^*$, where w^* is the optimal value of (GPC). We summarize our approach below and refer the reader to [17] and [29] for further details.

Solution Procedure GPC

Input. Data for (GPC). Batch size n, number of batches n_g, and n_x which is the size of the approximating problem used to obtain the candidate solution. Confidence level $1 - \alpha$ and t distribution quantile $t_{n_g-1,\alpha}$.

Output. Candidate design decision $\hat{x}$ and approximate $(1 - \alpha)$-level confidence interval $[0, \bar{G}_{n_g} + \epsilon_g]$ on $K\hat{x} + Ef(\hat{x}, \boldsymbol{TR}) - w^*$.

1. Sample observations $\boldsymbol{TR}^1, \ldots, \boldsymbol{TR}^{n_x}$ and solve (GPC_{n_x}) to obtain $\hat{x}$.
2. Sample i.i.d. batches $\boldsymbol{TR}^{i1}, \ldots, \boldsymbol{TR}^{in}$ for $i = 1, \ldots, n_g$.
3. For each $i = 1, \ldots, n_g$ calculate

$$G_n^i = K\hat{x} + \frac{1}{n}\sum_{j=1}^{n} f(\hat{x}, \boldsymbol{TR}^{ij}) - \min_{x \in X}\left[Kx + \frac{1}{n}\sum_{j=1}^{n} f(x, \boldsymbol{TR}^{ij})\right]. \qquad (21.11)$$

4. Let $\bar{G}_{n_g} = \frac{1}{n_g}\sum_{i=1}^{n_g} G_n^i$, $s_g^2 = \frac{1}{n_g - 1}\sum_{i=1}^{n_g}(G_n^i - \bar{G}_{n_g})^2$, and $\epsilon_g = t_{n_g-1,\alpha} s_g / \sqrt{n_g}$.
5. Print("Candidate well-location design solution:", $\hat{x}$, "Confidence interval on optimality gap:", $[0, \bar{G}_{n_g} + \epsilon_g]$).

The optimization in (21.11) over $x \in X$ corresponds to constraints (21.7b) and (21.7c). In our implementation of this solution procedure we solve the MINLPs in steps 1 and 3 using the generalized Benders decomposition scheme described in [28] and implemented in GAMS [4] using MINOS 5 to solve the nonlinear subproblems and CPLEX 7.0 to solve the mixed-integer master programs. Times required to solve typical instances of 200-scenario, 500-scenario, and 1000-scenario problems to within 0.1% of optimality on a 1.7 GHz Pentium IV machine with 1 GB of RAM are about 1.25 minutes, 6 minutes, and 10 minutes (with lack of proportionality due to different numbers of major iterations of the decomposition algorithm).

21.5 Results and discussion

We analyze solutions to (GPC) and describe our computational experience. Realizations of the heterogeneous hydraulic conductivity field were generated assuming a lognormal distribution, whose underlying normal distribution has mean parameter 1.14×10^{-4} m/s, standard deviation 1.45 m/s, and a correlation length of 100 m in all directions. The stochastic hydraulic control model was then approximately solved with $n_x = 1000$ scenarios to obtain the candidate design, $\hat{x}$, with a range of values of the environmental weight, λ_e. Some candidate solutions with associated expected cost, cost overruns, and environmental penalty are summarized in Table 21.1, which shows (unbiased) point estimates and confidence intervals for each of these terms of the objective and also shows estimates of the reliability of the design, i.e., the probability there are no violations of the environmental constraints.

First-stage decisions range from do nothing (with a very small weight on the hydraulic gradient penalty term) to the installation of three pumping wells (with a high weight on the penalty term). For different values of weights that lead to the same first-stage decisions, e.g., $\lambda_e \in [0.5, 10]$, the differences in the values of the objective function terms result from more or less pumping. For this problem, the installation (capital) costs are significantly larger than the pumping (operating) costs, and small increases in pumping costs, relative to the installation costs, can have a large impact on the reliability. For example, as λ_e increases from 0.50 up to 10, expected pumping costs grow from \$7560 to \$8000, but estimated reliability increases from 0.860 to 0.968. A similar (and more dramatic) effect occurs as we move from $\lambda_e = 0.01$ to $\lambda_e = 0.25$.

The solutions and associated output analysis from Table 21.1 allow a decision maker to quantitatively consider trade-offs between expected costs, along with the risk of cost overruns, and environmental penalties and associated reliability. In our computations, decreasing environmental penalties (which are explicit in the model) correspond to increasing reliability (which is not). However, this need not be the case as the latter does not consider the magnitude of the violation.

We computed confidence intervals on the optimality gap, $K\hat{x} + Ef(\hat{x}, \boldsymbol{TR}) - w^*$, with respect to a subset of the problem instances in Table 21.1. The point estimate of the optimality gap, $\bar{G}_{n_g}$, and the error due to sampling, ϵ_g, were computed using batch sizes of $n = 200$ and $n_g = 30$ batches. For example, with $\lambda_e = 1$ we obtained $\bar{G}_{n_g} = 0.77$, $\epsilon_g = 0.62$ for a confidence interval of [0.1.39] or about 1% of the point estimate of the objective function value in Table 21.1. Regardless of the value of λ_e, the variability associated with

Table 21.1. *(GPC_{n_x}) was solved with $\lambda_c = 1.5$ and λ_e ranging from 0.0001 to 50. Our candidate solutions $\hat{x}$ are specified in the Wells column (see Figure 21.3) and are based on $n_x = 1000$. The Objective column gives a point estimate (95% confidence interval half-width) of $K\hat{x} + Ef(\hat{x}, \boldsymbol{TR})$ based on 6000 observations of $\boldsymbol{TR}$, i.e., based on observations independent of those used to find $\hat{x}$. The next three columns specify similar values for the expected cost, expected cost overruns relative to $TC = 100$, and expected environmental penalty. The last column estimates the probability that there were no hydraulic gradient violations. Costs are in units of* \$1000, *and the hydraulic gradient penalty is in units of* cm^2.

λ_e	Wells	Objective	Cost	Overruns	Environ.	Reliability
0.0001	none	0.896(0.029)	0(0)	0(0)	8963(290)	0.008(0.017)
0.01	[7]	60.0(0.27)	56.8(0.12)	0(0)	317.9(18.3)	0.013(0.026)
0.10	[7]	78.2(1.8)	59.9(0.16)	0.0007(0.001)	182.6(17.6)	0.327(0.0096)
0.25	[7]	105.2(4.5)	60.5(0.16)	0.005(0.003)	178.7(17.6)	0.618(0.015)
0.50	[6,8]	125.0(1.8)	107.6(0.15)	7.56(0.15)	12.3(3.3)	0.860(0.009)
1.0	[6,8]	131.0(3.4)	107.7(0.16)	7.71(0.16)	11.7(3.3)	0.930(0.010)
5	[6,8]	177.0(16.3)	107.9(0.17)	7.94(0.17)	11.4(3.2)	0.963(0.005)
10	[6,8]	234.1(32.5)	108.0(0.17)	8.00(0.17)	11.4(3.2)	0.968(0.005)
50	[2,6,8]	347.6(57.6)	157.9(0.17)	57.9(0.17)	2.1(1.1)	0.988(0.003)

pumping costs and cost overruns is modest (see Table 21.1). This may provide comfort to a fiscally minded decision maker. However, due to its quadratic nature, the environmental penalty term has considerable variability, which degrades our ability to compute a tight confidence interval for more strident values of λ_e. For example, with $\lambda_e = 10$ we obtain $\bar{G}_{n_g} = 24.4$ and $\epsilon_g = 17.1$, which leads to a confidence interval width of about 17% of the objective value estimate. (The authors in [20] use importance sampling to improve reliability estimation.)

21.6 Conclusions

Groundwater quantity and quality management will be an increasingly important problem in the future as population and development lead to water demands which approach or exceed water availability in many places throughout the world. Numerical simulation and optimization models have proven to be useful tools for predicting and minimizing the impact of human stresses on the groundwater systems. In particular, stochastic programming models are well suited for identifying promising design alternatives that properly balance cost, expected performance, and reliability. In this chapter, we have shown how a multiobjective stochastic programming model can incorporate hydrogeologic uncertainty and can be used to quantify trade-offs for decision makers.

In closing, we mention a number of possible extensions of this work, motivated by the need to provide reliable information to designers and decision makers in an expedient manner. In this work, as in nearly all applications of stochastic programming, the scenar-

ios were generated assuming a known distribution, but there was no means of acquiring additional information about this distribution until the second stage, when it was assumed that the random parameters became known. In reality, groundwater management problems lend themselves to multistage stochastic programs in which imperfect information can be acquired (at some cost) in the early stages of the decision process. In such a case, the scenarios used in later stages of the decision process would be conditioned on the observations made in an earlier stage. While solution of such a multistage stochastic program would be challenging at this time, a combination of decision analysis and stochastic programming appears to be a reasonable approach for integrating measurement decisions with design and operation decisions.

Acknowledgments

The third author's research was supported by the National Science Foundation through grants DMI-9702217 and DMI-0217927, and by the State of Texas Advanced Research Program through grant 003658-0405.

Bibliography

[1] D. AHLFELD AND A. MULLIGAN, *Optimal Management of Flow in Groundwater Systems*, Academic Press, San Diego, CA, 2000.

[2] D. AHLFELD, R. PAGE, AND G. PINDER, *Optimal ground-water remediation methods applied to a superfund site: From formulation to implementation*, *Groundwater*, 33 (1995), pp. 58–70.

[3] Z. ARTSTEIN AND R. WETS, *Sensors and information in optimization under stochastic uncertainty*, Math. Oper. Res., 18 (1993), pp. 523–547.

[4] A. BROOKE, D. KENDRICK, A. MEERAUS, AND R. RAMAN, *GAMS, A User's Guide*, GAMS Development Corporation, Washington, DC, 1998.

[5] G. DAGAN, *Flow and Transport in Porous Formations*, Springer-Verlag, New York, 1989.

[6] H. DARCY, *Les fontaines publiques de la ville de Dijon*, Victor Dalmont, Paris, 1856.

[7] R. FREEZE AND J. CHERRY, *Ground Water*, Prentice–Hall, Englewood Cliffs, NJ, 1979.

[8] R. FREEZE AND S. GORELICK, *Convergence of stochastic optimization and decision analysis in the engineering design of aquifer remediation*, Groundwater, 37 (1999), pp. 934–954.

[9] A. FUTSCHIK AND G. PFLUG, *Optimal allocation of simulation experiments in discrete stochastic optimization and approximative algorithms*, Eur. J. Oper. Res., 101 (1997), pp. 245–260.

[10] L. GELHAR, *Stochastic Subsurface Hydrology*, Prentice–Hall, Englewood Cliffs, NJ, 1993.

[11] G. GOLUB AND C. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983.

[12] Y. HAIMES AND W. HALL, *Multiobjectives in water resource systems analysis: The surrogate tradeoff method*, Water Resources Res., 10 (1974), pp. 615–624.

[13] T. JONSBRÅTEN, R. WETS, AND D. WOODRUFF, *A class of stochastic programs with decision dependent random elements*, Ann. Oper. Res., 82 (1998), pp. 83–106.

[14] B. J. LENCE AND A. RUSZCZYŃSKI, *Managing water quality under uncertainty: Application of a new stochastic branch and bound method*, in Risk, Reliability, Uncertainty and Robustness of Water Resources Systems, Cambridge University Press, Cambridge, UK, 2002.

[15] F. LOUVEAUX, *Piecewise convex programs*, Math. Program., 15 (1978), pp. 53–62.

[16] S. LUCERO, *A Stochastic Programming Model for Groundwater Remediation*, University of California, Davis, CA, 2002.

[17] W. MAK, D. MORTON, AND R. WOOD, *Monte Carlo bounding techniques for determining solution quality in stochastic programs*, Oper. Res. Lett., 24 (1999), pp. 47–56.

[18] M. MAKOWSKI, L. SOMLYÓDY, AND D. WATKINS, *Multiple criteria analysis of water quality management in the Nitra basin*, Water Resources Bull., (1996), pp. 937–951.

[19] J. MULVEY, R. VANDERBEI, AND S. ZENIOS, *Robust optimization of large-scale systems*, Oper. Res., 43 (1995), pp. 264–281.

[20] S. RANJITHAN, J. EHEART, AND J. GARRETT, JR., *Neural network-based screening for groundwater reclamation under uncertainty*, Water Resources Res., 29 (1993), pp. 563–574.

[21] R. ROCKAFELLAR AND R. WETS, *Scenarios and policy aggregation in optimization under uncertainty*, Math. Oper. Res., 16 (1991), pp. 119–147.

[22] C. TIEDEMAN AND S. GORELICK, *Analysis of uncertainty in optimal groundwater contaminant capture design*, Water Resources Res., 29 (1993), pp. 2139–2153.

[23] Y. TUNG, *Groundwater management by chance-constrained model*, J. Water Resources Planning Management, 112 (1986), pp. 1–19.

[24] B. WAGNER, *Recent advances in simulation-optimization ground-water management modeling*, Rev. Geophys., 33 (1995), pp. 1021–1028.

[25] B. WAGNER AND S. GORELICK, *Optimal groundwater quality management under parameter uncertainty*, Water Resources Res., 23 (1987), pp. 1162–1174.

[26] B. WAGNER AND S. GORELICK, *Reliable aquifer remediation in the presence of spatially variable hydraulic conductivity: From data to design*, Water Resources Res., 25 (1989), pp. 2211–2225.

[27] J. WAGNER, U. SHAMIR, AND H. NEMATI, *Groundwater quality management under uncertainty: Stochastic programming approaches and the value of information*, Water Resources Res., 28 (1992), pp. 1233–1246.

[28] D. WATKINS, JR. AND D. MCKINNEY, *Finding robust solutions to water resources problems*, J. Water Resources Planning Management, 123 (1997), pp. 49–58.

[29] D. WATKINS, JR., D. MORTON, AND D. MCKINNEY, *Monte Carlo techniques for estimating solution quality in stochastic groundwater management models*, in Proceedings of the XII International Conference on Computational Methods in Water Resources, Crete, 1998, pp. 67–74.

[30] A. YANG, *Stochastic Heterogeneity and Dispersion*, Ph.D. dissertation, University of Texas, Austin, TX, 1990.

[31] W. YEH, *Systems analysis in ground-water planning and management*, J. Water Resources Planning Management, 118 (1986), pp. 224–237.

Chapter 22

Catastrophic Risk Management: Flood and Seismic Risks Case Studies

Tatiana Ermolieva and Yuri Ermoliev**

22.1 Introduction

Losses from human-made and natural catastrophes are rapidly increasing. Within the last three decades the direct damages only from natural disasters have increased ninefold (see [25, 34]). The main reason for this is the clustering of people and capital in hazard-prone areas as well as the creation of new hazard-prone areas, a phenomenon that may be aggravated by a lack of knowledge of the risks. It is estimated (see [35]) that within the next 50 years more than a third of the world population will live in seismically and volcanically active zones. Analysis of insurance companies shows that because of economic growth in hazard-prone areas, damages due to natural catastrophes have grown at an average annual rate of 5% [19]. The possibility of more frequent catastrophes dominates discussions of current global changes. In fact, the main point of the climate change debates concerns the increasing frequency of extreme floods, droughts, and windstorms rather than the increasing global mean temperature which can be within the difference between the average temperature of cities and their surrounding rural areas. Another source of catastrophes is associated with increasing interdependencies among different countries. Dantzig [9] compares our society with a busy highway where disruptions in one of its parts may lead to fundamental traffic jams in other parts.

The increasing vulnerability of our society calls for new integrated approaches to economic developments and risk management with an explicit emphasis on the possibility of catastrophes. The standard economic theory is dominated by truncated models of uncertainties, represented by a finite manageable number of contingencies well known to the whole society, which can, therefore, be priced and spread over the whole society through

*International Institute for Applied Systems Analysis (IIASA), A-2361 Laxenburg, Austria (ermol@iiasa.ac.at, ermoliev@iiasa.ac.at).

markets. Under such assumptions of certainty, catastrophes pose no special problems [3].

Insurance risk theory has developed independently of the fundamental economic ideas (see discussion in [3, 20]). The central problem of this theory is modeling the probability distribution of total future claims [23], which is then used to evaluate ruin probabilities, premiums, reinsurance arrangements, etc. This theory essentially relies on the assumption of independent, frequent, low-consequence (conventional) risks, such as car accidents, for which decisions on premiums, estimates of claims, and likelihood of insolvency (probability of ruin) can be calculated by using rich historical data. The frequent conventional risks also permit simple "more-risks-are-better" strategies with simple trial-and-error or learning-by-doing procedures for adjusting insurance decisions.

Catastrophes produce losses highly mutually dependent in space and time, which challenges the standard risk pooling concepts and the standard extremal value theory [11]. The law of large numbers does not operate (in general), and the probability of ruin can be reduced not just by pooling risks but only if insurers deliberately select the dependent fractions of catastrophic risks they will cover. The existing extremal value theory deals also primarily with independent events assuming these events are quantifiable by a single number. Definitely catastrophes are not quantifiable events in this sense. They may have quite different spatial and temporal patterns, which cause significant heterogeneity of losses in space and time. These losses can be dramatically affected by risk mitigation decisions (say, by construction of a dike or a flood retention area) and loss spreading schemes within a country or on the international level through the insurance or financial markets.

Catastrophe modeling [43] is becoming increasingly important in risk management for estimating dependent catastrophic losses and making decisions on the allocation of coverage, premiums, reinsurance agreements, and the effects of mitigation measures.

The aim of this paper is to show that the choice of decisions in the presence of catastrophic risks can be regarded as a stochastic optimization problem. This discussion closely follows the papers [1, 2, 14, 15, 17, 18]. Section 22.2 illustrates the peculiarities of emerging stochastic optimization problems by using a typical model of risk theory. Section 22.3 provides more motivations by outlining important case studies. Sections 22.4 and 22.5 discuss a rather general model attempting to bridge decision-oriented economic theory with risk theory and catastrophe modeling. Risk management decisions are evaluated from perspectives of the welfare growth in the region. We use such economically sound risk measures as expected costs of overpayments and borrowing, which have strong connection with standards in the insurance business, the insolvency and stability constraints, and the conditional-value-at-risk (CVaR) type of risk measures. Section 22.6 discusses the results of the case study in the seismic-prone Toscana region of Italy. Section 22.7 outlines the computational procedure.

22.2 The standard insurance risk model

Consider a simple model of growth under shocks, which is a stylized version of insurance business [10]. The main variable of concern is the risk reserve r^t at time t: $r^t = r_0 + \pi^t - A^t$, $t \geq 0$, where π^t, A^t are aggregated premiums and claims, and r_0 is the initial risk reserve. The process $A^t = \sum_{k=1}^{N(t)} S_k$, where $N(t)$, $t \geq 0$, is a counting process for a number of claims in interval $[0, t]$ (e.g., a Poisson process) with $N(0) = 0$, and $\{S_k\}_1^\infty$ is a sequence of

independent and identically distributed random variables (claims), in other words, replicates of a random variable S. The inflow of premiums π^t pushes r^t up, whereas the random outflow A^t pushes r^t down.

The main problem of risk theory [10, 23] is the evaluation of the ruin probability $\Psi = P\{r^t \leq 0 \text{ for some } t, t > 0\}$ under different assumptions on π^t, A^t. There are several cases where Ψ can be explicitly given, or at least given in a form suited for numerical calculations. An important case arises when the claim distribution is a mixture of exponential distributions and claims occur according to a Poisson process. There are numerous approximations for the probability distribution of A^t. Most of them provide satisfactory results only in the area of mean values and cannot be applied to catastrophes.

The typical actuarial analysis is based on the following. Assume that $N(t)$ and S_k are independent, $N(t)$ has intensity α, i.e., $E\{N(t)\} = \alpha t$, and $\pi^t = \pi t$, $\pi > 0$. Then the expected profit over the interval $[0, t]$ is $(\pi - \alpha ES)t$; that is, the expected profit increases in time for $\pi - \alpha ES > 0$. The difference $\pi - \alpha ES$ is the "safety loading." The strong law of large numbers implies that $[\pi^t - A^t]/t \to [\pi - \alpha ES]$ with probability 1. Therefore, in the case of positive safety loading, $\pi > \alpha ES$, we have to expect that the real random profit $\pi^t - A^t$ for large enough t would also be positive under the appropriate choice of premium $\pi = (1 + \rho)\alpha ES$, where ρ is the "relative safety" loading, $\rho = (\pi - \alpha ES)/\alpha ES$. But this holds only if the ruin does not occur before time t. This is a basic actuarial principle: premiums are calculated by relying on the mean value of aggregated claims increased by the (relative) safety loading. Thus, practical actuarial approaches ignore complex interdependencies among timing of claims, their sizes, and the possibility of ruin, $r^t \leq 0$. The random jumping process r^t is simply replaced by a linear in t function $\bar{r}^t = r_0 + (\pi - \alpha ES)t$.

Various decision variables affect Ψ. Claim size S depends on the coverages of the insurer from different locations. Important decision variables are r_0, π, and reinsurance arrangements, for example, the "excess of loss" reinsurance contract. In this case, the insurer retains only a portion, $S(x) = \min\{S, x\}$, $x \geq 0$, of a claim S, and the remaining portion is passed to the reinsurer. The reinsurance contracts with deductibles are defined by two variables $x = (x_1, x_2)$. In this case $S(x) = \max\{x_1, \min[S, x_2]\} - x_1$, $x_1 \geq 0$, $x_2 \geq 0$ is retained by the insurer. The reduction of Ψ to acceptable levels can be viewed as a chance constraint problem [38]. The complexity is associated with the jumping process A^t with analytically intractable dependencies of A^t on decision variables, which restricts the straightforward use of conventional stochastic optimization (STO) methods. The direct sample mean estimation of $\Psi(x)$ requires a very large number of observations and leads to discontinuous functions. The following simple idea can be used for rather general problems to overcome these difficulties.

Consider $t = 0, 1, \ldots$, and assume that r^t can be subdivided into a "normal" part (including r_0), M^t, associated with ordinary claims, and a "catastrophic" part B^t, $\pi^t = \pi t$, where π is the rate of premiums related to catastrophes; the probability of a catastrophic event p is characterized by a probability distribution in an interval $[\underline{p}, \overline{p}]$, and the probability distribution $V_t(z) = P[M^t < z]$ can be evaluated. Assume also that ruin may occur only due to a catastrophe. Then the probability of ruin after the first catastrophe and with the excess of loss contract is defined as a function

$$\Psi(x) = E \sum_{t=1}^{\infty} p(1-p)^{t-1} V_t(\min\{x, B^t\} - \pi t). \tag{22.1}$$

In general cases we can use an arbitrary "small" auxiliary random variable instead of M^t [14, 15]. Definitely, (22.1) enables us to evaluate $\Psi(x)$ much faster (fast Monte Carlo sampling) in contrast to direct evaluation of $\Psi(x)$ by straightforward Monte Carlo sampling. This is the main idea in dealing with rare catastrophic events. The search for a desirable x can be based on methods outlined in section 22.7.

22.3 Overview of case studies

To better understand the main features of the model described in the next section, let us sketch out some case studies [1, 2, 17] carried out at the International Institute for Applied Systems Analysis (IIASA). The main concern of these case studies is related to issues emphasized by Froot in [19]. Froot admitted that most of the catastrophic losses

> are paid ex-post by some combination of insurers and reinsurers (and their investors), insured, state and federal agencies and taxpayers, with only some of these payments being explicitly arranged ex-ante. This introduces considerable uncertainty about burden sharing into the system, with no particular presumption that the outcome will be fair. The result is incentives for players to shift burdens towards others, from the homeowner who builds on exposed coastline, to insurers who write risks that appear highly profitable in the absence of a large event.... But most importantly, bad or inefficient risk sharing raises the cost of capital for companies and requires returns for households, reducing the amount of profitable investments and the rate of growth of the economy... it is worth noting that the gains from higher growth rate are huge....

For Hungary [24], facing special problems of a poor and immobile population, ex ante mechanisms to fund the costs of recovery and, in particular, the establishment of a multipillar flood loss-sharing program, are especially important. In the analysis of the Upper Tisza river pilot region [17] it is assumed, in particular, that for the first pillar the government would provide compensation of a limited amount to all households that suffer losses from flooding. As the second pillar, a special regional fund would be established through a mandatory public flood insurance on the basis of location-specific risk exposures. It is assumed that the governmental financial aid is regulated through this fund. As a third pillar, a contingent credit may also be available to provide an additional injection of capital to stabilize the system. In the latter case, the lender charges a fee that the borrower (in our case, the fund) pays as long as the trigger event does not occur. If the event does occur, the borrower rapidly receives the fund. The advantages of this financial arrangement in contrast to catastrophic bonds are discussed, e.g., in [37].

Such a program would increase the responsibility of individuals and local governments for flood risks and losses. Local governments may be more effective in the evaluation and enforcement of loss-reduction and loss-spreading measures, but this is possible only through location-specific analysis of potential losses, the mutual interdependencies of these losses, and the sensitivities of the losses to new risk management strategies.

The lack of historical data is a main challenge in dealing with rare catastrophes. Purely adaptive, learning-by-doing types of approaches may be very expensive and dangerous. The development of models can be viewed as a mitigation measure enabling the simulation of probable catastrophes for designing ex ante preparedness programs.

The analysis of possible gains and losses from different arrangements of the program outlined above is a multidisciplinary task, which has to take into account the frequency and intensity of hazards, the stock of capital at risk, its structural characteristics, and different measures (in particular, engineering, financial) of vulnerability. These efforts require the development of so-called catastrophe models [43]. For this purpose, Ermolieva et al. [17] discuss a GIS-based catastrophe model developed for the Upper Tisza pilot region that, in the absence of historical data, simulates samples of mutually dependent potential losses at different locations. The model emphasizes the cooperation of various agents in dealing with catastrophes. The solution to catastrophic risk management, especially for small economies with limited risk absorption capacity, cannot be accomplished (see [2, 8, 37]) without pooling of risk exposures. The proposed model involves pooling risks through mandatory flood insurance based on location-specific exposures, partial compensation to the flood victims by the central government, and a contingent credit to the pool. This program encourages accumulation of regional capital to better buffer international reinsurance market volatility. To stabilize the program such economically sound risk indicators as expected overpayments by individuals (cells of flood-prone areas) and an expected shortfall of the mandatory insurance are used. These indicators together with the so-called stopping times orient the analysis toward the most destructive scenarios. It was shown (see [14, 15] and references therein) that the explicit introduction of ex post borrowing (see also section 22.5) as a measure against insolvency enables us to approximate the insolvency constraint by a convex optimization problem, whereas the use of the contingent credit leads to the CVaR type of risk measures.

For the seismic-prone Irkutsk region in Russia [2], the focus of the analysis was on the feasibility of an insurance pool to cover catastrophic losses subject to strong standard insolvency regulations. In contrast to the case study in Hungary, the contribution of different insurers to the stability of the pool was explicitly analyzed by taking into account the transaction costs and effects of mutual dependencies among claims from different locations (cells).

Many authors have stressed the need for better models to improve established insurance practices for evaluating catastrophe coverages (see, for instance, [43]). Such models can be even more critical for guidance and setting regulations in countries that are moving toward market economies. In Russia [2] new legislative instruments and government resolutions are creating a framework for risk management similar to that existing in the OECD countries. However, in Russia and other transition countries the emergence of a viable insurance industry is slow and subject to insolvency risks due to problems of the national economies, the lack of consolidated experience and practicable guidance, and the lack of sufficient risk reserves of the existing companies. For example, when insurance is available in seismic regions, premiums neither are based on the probability of occurrence of earthquakes nor do they differentiate among geological situations and construction type. The model proposed in [1, 2] is a pilot exercise, which, however, can create the basis for cooperation among researches, insurers, and regulatory bodies in transition countries. In the case of Russia's emerging insurance industry, cooperation among insurers will undoubtedly play an important role in stabilizing the insurance market. A key problem, however, is the lack of necessary information on the distribution of losses among locations. For this purpose the region-specific earthquake generator (see [5, 42]) was designed and incorporated within the STO model [1, 2].

The case study for the seismic-prone Italian region especially illustrates the fact that neither the market nor the government will be acceptable as the mechanism for catastrophic risk management. Thus, some form of a public-private partnership may be appropriate [30].

A well-known example of a government acting as a primary insurer is the United States National Flood Insurance Program (NFIP), which seeks to provide insurance at actuarially fair premiums combined with incentives for communities and homeowners to take appropriate loss-reducing measures. Given the size of the United States and the large number of persons living in flood plains, the program is sufficiently diversified to cover most regional losses with premium payments. In contrast to the NFIP, some government insurance schemes in Europe, e.g., the French national insurance program, cross subsidize claims. This is because the French constitution (1946, 1958) established the principle of "the solidarity and equality of all French citizens facing the expenses incurred through national calamities" [21].

However, even if many governments are pursuing policies to reduce their role in compensating victims, a study [31] confirms that the victims and their governments bear the major losses from natural disasters and, worldwide, there is only moderate risk transfer with insurance. An important consideration for national insurance strategies is linking private insurance with mitigation measures to reduce losses. Insurers, however, are reluctant to enter markets that expose them to a risk of bankruptcy. In the United States, for example, many insurers have pulled out of catastrophic risk markets in response to their large losses from natural catastrophes in the last decade [25].

To reduce their risk of insolvency, insurers' strategies may be based on modeling tools that account for the complexity implied by the manifold dependencies in the stochastic process of catastrophic events, decisions, and losses. For example, to study the problem in its complexity for the Toscana region, a spatial-dynamic, stochastic optimization model has been developed in [13, 14, 16, 18] and is described below.

In Italy, a law for integrating insurance in the overall risk management process was proposed only in late 1997 (within the Design of Law 2793: "Measures for the stabilization of the public finance"). This opened a debate, which has not yet been concluded by a legislative act. Therefore policy options for a national insurance strategy are still open to investigation. The Institute for Research on Seismic Risk of the Italian National Research Council made data from a previous study available [36]. These have been incorporated into a Monte Carlo catastrophe model, which simulates occurrence of earthquakes affecting the region, calculates attenuation according to the geological characteristics, and finally determines the acceleration at the ground in each municipality. The model explicitly incorporates the vulnerability of the built environment, with data on number and types of buildings in each municipality of the region.

Since the discussion of GIS-based catastrophe models deserves a separate paper, our analysis in the next sections is concentrated primarily on the evaluation of different catastrophic risk management policies assuming that mutually dependent spatial losses are generated in time and space (cells) by a catastrophe model.

22.4 STO model

Catastrophes may lead to large costs, social disruption, and economic stagnation. A catastrophe would ruin many agents if their risk exposures were not properly managed. To design

safe catastrophic risk management strategies it is necessary to define at least the following: patterns of possible disasters in space and time, a map of regional values and their vulnerability, and feasible decisions, e.g., insurance coverages. The model of this section uses this information. It emphasizes the collective nature of catastrophe risk management. The aim of this model is to address only main features of the problem. Basically, we assume that the goal of the insurance sector is to maximize its wealth while maintaining survival and stability of growth (see also the discussion in [41]). In a similar manner, other agents are concerned with their sustained wealth growth, whereas the main concern of the government is the sustained welfare growth of the region. The model emphasizes catastrophe risk management as a long-term business rather than as subject of annual accounting and taxation. Accordingly, catastrophe reserves should be accumulated over years.

Assume that the study region is divided into subregions or cells $j = \overline{1, m}$. A cell may correspond to a collection of households, a zone with similar seismic activity, a watershed, a grid with a segment of a gas pipeline, etc. The choice of cells provides a desirable representation of losses. For each cell j there exists an estimate of its wealth at time t that may include the value of infrastructure, houses, factories, etc. A sequence of random catastrophic events $\omega = \{\omega_t, t = \overline{0, T-1}\}$ affects different cells $j = \overline{1, m}$ and generates at each $t = \overline{0, T-1}$ mutually dependent losses $L_j^t(\omega)$, i.e., damages of the wealth at j; T is a time horizon. These losses can be modified by various decision variables. Some of the decisions reduce losses, say, a dike, whereas others spread them on a regional, national, and international level, e.g., insurance contracts, catastrophe securities, credits, and financial aid. If x is the vector of the decision variables, then the losses $L_j^t(\omega)$ are transformed into $L_j^t(x, \omega)$. For example, we can think of $L_j^t(x, \omega)$ as $L_j^t(\omega)$ being affected by the decisions of the insurance to cover losses from a layer $[x_{j1}, x_{j2}]$ at a cell j in the case of a disaster at time t:

$$L_j^t(x, \omega) = L_j^t(\omega) - \max\{x_{j1}, \min[x_{j2}, L_j^t]\} + x_{j1} + \pi_j^t,$$

where $\max\{x_{j1}, \min[x_{j2}, L_j^t]\} - x_{j1}$ are retained by insurance losses, and π_j^t is the premium.

In the most general case, the vector x comprises decision variables of different agents, including governmental decisions, such as the height of a new dike or a public compensation scheme defined by a fraction of total losses $\sum_{j=1}^m L_j^t$. The insurance decisions concern premiums paid by individuals and the payments of claims in the case of catastrophe. There are complex interdependencies among these decisions, which call for the cooperation of agents. For example, the partial compensation of catastrophe losses by the government enforces decisions on loss reductions by individuals and, hence, increases the insurability of risks and helps the insurance industry to avoid insolvency. On the other hand, the insurance combined with risk-reduction measures can reduce losses, compensations, and governmental debt and stabilize the economic growth of the region and the wealth of individuals.

We assume that ω is an element of a probability space $(\Omega, \mathcal{F}, P)$, where Ω is a set of all possible ω and $\mathcal{F}$ is a σ-algebra of measurable (with respect to probability measure P) events from Ω. Let $\{\mathcal{F}_t\}$ be a nondecreasing family of σ algebras, $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}$, $\mathcal{F}_t \subseteq \mathcal{F}$. Random losses $L_j^t(\omega)$ are assumed to be $\mathcal{F}_t$-measurable; i.e., they depend on the observable catastrophes until time t. In the following we specify dependencies of these variables on ω, although sometimes we do not use ω when these dependencies are clear from the text.

Catastrophe losses are shared by many participants, such as individuals (cells), governments, insurers, reinsurers, and investors. In the model we call them "agents," since the

main balance equations of our model are similar for all of them. For each agent i a variable of concern is the wealth W_i^t at time $t = \overline{0, T}$

$$W_i^{t+1}(\omega) = W_i^t(x, \omega) + I_i^t(x, \omega) - O_i^t(x, \omega), \quad i = \overline{1, n}, \quad t = \overline{0, T-1}, \quad \omega \in \Omega, \tag{22.2}$$

where W_i^0 is the initial wealth. This is a rather general process of accumulation, which, depending on the interpretation, can describe the accumulation of reserve funds, the dynamics of contamination, or processes of economic growth with random disturbances (shocks), reserves of the insurance company at moment t, the gross national product of a country, or the accumulated wealth of a specific region. In more general cases, when catastrophes may have profound effects on economic growth, this model can be generalized to an appropriate version of economic-demographic model (see, for example, in [32]) enabling it to represent movements of individuals and the capital accumulation processes within the economy.

For the simplicity of the exposition we do not discuss discount rates in these equations since catastrophes require nonstandard approaches. In particular, induced by catastrophes, discount rates become important, which is evident from the evaluation (22.1). We use also the same index i for quite different agents. Therefore, the variables $I_i^t(x, \omega)$, $O_i^t(x, \omega)$ may have quite a different meaning. For example, for each insurer i we can think of I_i^t as premiums π_i^t which are ex ante arranged and do not depend on ω, whereas O_i^t is defined by the claim size S_i^t and possible transaction costs which triggers a random jump of the risk reserve W_i^t (usually denoted as R_i^t) downward at random times of catastrophic events (as in the simple model of section 22.2). If i corresponds to a cell, then income I_i^t may be affected by a catastrophic event ω generated by a catastrophe model. The incomes I_i^t can be defined by a set of scenarios or through a regional growth model with geographically explicit distribution of the capital among cells. The term O_i^t may include losses L_i^t, taxes and premiums paid by i. For central or local governmental agent i (e.g., mandatory insurance, catastrophe fund) I_i^t may include a portion of taxes collected by the government (compensations of losses by the government), and O_i^t may consist of mitigation costs, debts, loans, and fees paid for ex ante contingent credits.

Catastrophes may cause strong dependencies among claims S_i^t for different insurers i. These claims are defined by decisions on coverages of losses L_j^t from different locations j. For example, let us denote by x_{ij}^t a searched fraction of L_j^t covered by insurer i, e.g., assume $i = \overline{1, n}$. Then

$$\sum_{i=1}^{n} x_{ij}^t \le 1, x_{ij}^t \ge 0, \quad j = \overline{1, m}, \tag{22.3}$$

and claims S_i^t are linear functions of $x = \{x_{ij}^t, i = \overline{1, n}, j = \overline{1, m}, t = \overline{0, T-1}\}$:

$$S_i^t(x, \omega) = \sum_{j=1}^{m} L_j^t x_{ij}^t, \quad i = \overline{1, n}, \quad t = \overline{0, T-1}.$$

If I_i^t, O_i^t simply correspond to premiums π_i^t and claims S_i^t, then the wealth of insurer i (its risk reserves) are calculated for $t = \overline{0, T-1}$, $\omega \in \Omega$ as follows:

$$R_i^{t+1}(x, \omega) = R_i^t(x, \omega) + \sum_{j=1}^{m} \pi_{ij}^t x_{ij}^t - \sum_{j=1}^{m} L_j^t(\omega) x_{ij}^t, \tag{22.4}$$

where π_{ij}^t are rates of premiums per unit of coverage.

For each i consider a stopping time τ_i for process $W_i^t(x, \omega)$, i.e., a random variable with integer values, $t = \overline{0, T}$. The event $\{\omega : \tau_i = t\}$ with fixed t depends only on the history till t, and it corresponds to the decision to stop process $W_i^t(x, \omega)$ after time t. Therefore, τ_i in the case of W_i^t defined according to (22.4) depends on $\{x_{ij}^k, i = \overline{1, n}, j = \overline{1, m}, k = \overline{0, t}\}$; i.e., it is a function $\tau_i(x, \omega)$. Examples of τ_i may be $\tau_i = T$, the time of the first catastrophe, or the time of the ruin before a given time T : $\tau_i(x, \omega) = \min\{T, \min[t : W_i^t(x, \omega) < 0, t > 0]\}$. The last example defines τ_i as a rather complex implicit function of x.

Assume that each agent i maximizes (possibly negative) wealth at $t = \tau_i$. The notion of wealth at t requires exact definition since it must represent, in a sense, the whole probability distribution W_i^t. The traditional expected value EW_i^t may not be appropriate for probability distributions of W_i^t affected by rare catastrophes of high consequences. As a result they may have a multimode structure with "heavy tails." We can think of the estimate for W_i^t as a maximal value V_i^t, which does not overestimate, in a sense, random value W_i^t, i.e., cases when $\min_{s \le t}(W_i^t(q, \omega) - V_i^t) < 0$. Formally, V_i^t can be chosen by maximizing

$$V + \gamma E \min\{0, W_i^t - V\} \tag{22.5}$$

or a more general function $V + \gamma E d(W_i^t - V)$ for appropriate function $d(\cdot)$ and $\gamma > 0$. The second term can be considered as the risk of overestimating the wealth $W_i^s(x, \omega)$ for $s = 0, 1, \ldots, t$. This concept corresponds to the CVaR risk measure (see [4, 27, 40]). The maximization of (22.5) is a simple example of the so-called stochastic maximin problems. It is easy to see from the optimality conditions for this problem (see [12, pp. 165, 416]) that for continious distributions the optimal value V satisfies condition $P[W_i^t \le V] = 1/\gamma$. For the normal distribution and $\gamma = 2$, it coincides with the traditional mean value EW_i^t. In the case of quadratic function $d(\cdot)$ and $\gamma = \infty$, i.e., the maximization of $E(W_i^t - V)^2$, the optimal $V = EW_i^t$.

Besides the maximization of wealth, the agent i is concerned with the risk of insolvency, i.e., when $W_i^s < 0$ for some $s = 0, 1, \ldots, t$, as well as the lack of sustained growth, i.e., when $I_i^s - O_i^s < 0$ for some $s = 0, 1, \ldots, t$. In accordance with this consider the stochastic goal functions

$$\begin{aligned} f_i^t(x, V, \omega) = V_i^t &+ \gamma_i \min\left\{0, \min_{s \le t}[W_i^s(x, \omega) - V_i^s]\right\} + \delta_i \min\left\{0, \min_{s \le t} W_i^s(x, \omega)\right\} \\ &+ \beta_i \min\left\{0, \min_{s \le t}[I_i^s(x, \omega) - O_i^s(x, \omega)]\right\}, \\ F_i(x, V) &= E f_i^{\tau_i(x,\omega)}(x, V, \omega), \end{aligned} \tag{22.6}$$

where nonnegative γ_i, δ_i, β_i are substitution coefficients between wealth V_i^t and risks of overestimating wealth, insolvency, and overestimating sustained growth. If a catastrophe is considered as the most distractive event, then we can use in the definition of f_i^t simply $s = t$ instead of $\min_{s \le t}$. These requirements reflect survival and stability constraints of agents. In (22.6) we use a modified form of (22.5), which is more appropriate for dynamic problems. Each agent attempts to maximize $F_i(x, V)$.

Pareto optimal improvements of risk situations with respect to goal functions $F_i(x, V)$

of different agents can be achieved by maximizing

$$W(x, V) = \sum_{i=1}^{n} \alpha_i F_i(x, V) \tag{22.7}$$

for different weights $\alpha_i \geq 0$, $\sum_{i=1}^{n} \alpha_i = 1$. These weights reflect the importance of the agents. The maximization of $W(x, V)$ for different weights α_i, $i = \overline{1, n}$, corresponds to a stochastic version of the welfare analysis [22].

When $n > 1$, this model generalizes Borch's [7] fundamental ideas of risk sharing to the case of catastrophic risks. In the Borch model risks from different locations are substitutable, and the insurance pool is concerned only with the redistribution of the total risk mass. According to (22.3), our model emphasizes differences among risks from different locations, i.e., $m > 1$ in contrast to $m = 1$ of Borch's model.

Random functions $f_i^t(x, V, \omega)$ have a complex nested analytically intractable structure defined by simulated patterns of catastrophes. Their nonsmooth character is due to the presence of operators min and stopping times τ_i, which may be complex implicit functions of (x, ω). When $f_i^t(x, V, \omega)$ are concave functions in x as min of linear functions, then expectations $F_i(x, V) = Ef_i^t(x, V, \omega)$ are also concave functions in x for fixed t. The use of stopping times, $t = \tau_i$, generally destroys their concavity and even continuity. If stopping times do not depend on x, then these expectations are also concave. The use of such risk functions as in (22.5) is similar to the [33] mean-semivariance model and the [29] model with absolute deviations. Connections of problems (22.5) with the CVaR risk measure are established in [40].

The choice of weights (risk coefficients) γ_i, δ_i, β_i, provides different trade-offs between wealth and risks. The increase of these parameters better eliminates corresponding risks.

22.5 Insolvency, stopping time, and nonsmooth risk functions

A key issue for selecting catastrophic risk portfolios is the financial ruin of insurers. It was shown (see [13]), that when risk coefficients γ_i, δ_i, β_i in (22.7) become large enough, then the probability of associated risks, in particular the probability of ruin, drops below a given level p:

$$P\left[\min_{s \leq \tau_i} W_i^s < 0, \ i = \overline{1, n}\right] \leq p. \tag{22.8}$$

The maximization problem defined by (22.6)–(22.7) is much simpler than the problem defined in terms of the chance constraint (22.8). The functions $F_i(x, V)$ defined according to (22.6) for $W_i^t(x, \omega) := R_i^t(x, \omega)$ are concave, whereas constraints (22.8) for the same case may have discontinuitous character, e.g., if ω has a discrete distribution. The problem defined in terms of the chance constraints (22.8) has a convex feasible set only under a strong assumption on the probability measure.

The discontinuous nature of the problem (22.6)–(22.7) may still be connected with the stopping time defined as the ruin (insolvency) moment. Different smoothing techniques

for this case are analyzed in [15]. In particular, a rather natural idea of smoothing consists of introducing the possibility of borrowing money in the case of insolvency. It is natural to expect that when the payment for borrowing is high, agents will tend to exclude such a necessity through a reasonable selection of the portfolios, i.e., to keep constraints on the insolvency within reasonable limits. Let us slightly modify the process (22.2):

$$W^{t+1}(x, y, \omega) = W^t(x, y, \omega) + I^t(x, \omega) - O^t(x, \omega) + y_{t+1} - (1 + \beta_t)y_t, \qquad (22.9)$$

where for the simplicity of notation we do not use here index i, y_t is a value of borrowing on the interval $[t-1, t)$, β_t is the bank interest for the credit on the interval $[t-1, t)$, and $y = \{y_0, \ldots, y_T\}$. According to (22.9), the borrowing taken out at the moment t to maintain solvency should be paid off at the next instant of time $t+1$ with interest β_t. If the reserves of the company are not sufficient for this purpose, then new loans are taken. The following fact is the key for dealing with discontinuities of the stopping time effects and the insolvency constraints. Let us represent the process $W^t(x, y, \omega)$ as

$$W^t(x, y, \omega) = \widetilde{W}^t(x, \omega) - \sum_{s=1}^{t-1} \beta_s y_s + y_t, \quad \widetilde{W}^t(x, \omega) = W^0 + \sum_{s=1}^{t} (I^s(x, \omega) - O^s(x, \omega))$$

and let $(x^*(\beta), V^*(\beta), y^*(\beta))$ be a solution of the following problem: maximize

$$F(x, V) = E \max_{y \geq 0} [f^T(x, V, y, \omega) - (1 + \beta_T)y_T], \quad W^t(x, y, \omega) \geq 0, \quad 0 \leq t \leq T, \qquad (22.10)$$

where $f^t(x, V, y, \omega)$ is defined as in (22.6) for $W^t(x, y, \omega)$ defined according to (22.9).

Theorem 22.1 (see [15]). *Assume that* $\overline{R}_t(0) \geq 0$, $P[\widetilde{W}^t(x, \omega) = 0] = 0$, *for any* $x \in X$, $t = \overline{1, T}$. *Then the probability of borrowing can be arbitrarily small by taking interest coefficients* β_t, $t = \overline{1, T}$, *large enough, i.e.,* $P[\widetilde{W}^t(x^*(\beta_t), \omega) \geq 0, t = \overline{1, T}] \to 1$ *almost surely for* $\min_{1 \leq t \leq T} \beta_t \to \infty$.

22.6 The Tuscany region case study

We now specify the general model described in section 22.4 to the Tuscany region. The region has been subdivided into $M \approx 300$ subregions, which correspond to the number of its municipalities. For each municipality j, number and types of buildings, their vulnerability, and number of built cubic meters are available. These represent the estimate of wealth W_j in the municipality j. Using data and models in [36], a catastrophe generator has been created (see [2, 5, 42]) using the Gütenberg–Richter law and the attenuation characteristics of the region (see Figure 22.1). This enables us to generate the occurrences of earthquakes at random time moments, intensities, and accelerations in each municipality. The generator could be easily adapted to incorporate different kinds of distributions, non-Poissonian catastrophic processes, as well as microzoning within a municipality. It produces earthquake scenarios at random time moments according to geophysical characteristics of faults and soil type.

Simulated in time and space, earthquakes $\omega_0, \ldots, \omega_t$ may occur in different municipalities, inside or outside the region, have random magnitudes, and, therefore, affect a random number of municipalities.

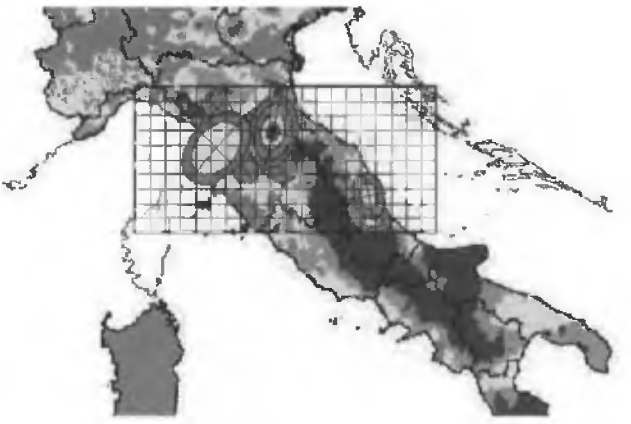


Figure 22.1. *Earthquake generator.*

In municipalities affected at time t the vulnerability relations between accelerations and losses [36] according to the type (masonry or reinforced concrete), age, and maintenance of the buildings are used to estimate the number of cubic meters of destroyed properties. The economic loss of destroyed cubic meters of a building is defined as the cost for their reconstruction. Then it is possible to be independent of contingent pricing by considering the cost of reconstruction per cubic meters to be the monetary unit. In this way the simulation of time histories for possible earthquakes in the region produces the sets of economical losses and enables the design of an insurance program. It also enables us to determine in which way preventive retrofitting could decrease the losses: this is easily done by a consequent decrease of the vulnerability indices in the loss model. In this way it is possible to study the interplay between structural measures and risk sharing for an integrated risk management approach and to design an insurance system linked to incentives for retrofitting of the built environment. Our analysis was primarily concerned with the following. In its early version the Italian Design of Law 2793 (1998), to reduce the impact of natural disasters on the governmental budget, included in provisions for an insurance program against all natural hazards. It was intended not to make this insurance mandatory, but to make mandatory the extension of a fire insurance policy to all natural hazards, in a way similar to the French system (see section 22.3). In addition to tax incentives for such an insurance, it stipulated a maximum exclusion layer of 25%, the creation of a pool of insurance companies with an appropriate reserve fund, e.g., corresponding to the annual average government payment for compensating losses (with some forms of state guarantee to be specified further), and linking of the premium to the premium for fire policy. This article was withdrawn, and later proposals are still the subject of discussion.

Starting from these principles, the case study intends to demonstrate how the model analyzes and offers the decision makers different policy options. Let us assume that an insurance company (this might be a pool of companies or the government itself acting as an insurer) covers a fraction, e.g., $q = 0.75$, of earthquake losses. The rest $\nu = 1 - q$, according to the Italian Design of Law, would be compensated by the state. The state would also be expected to feed the reserve funds in case of excessive losses.

The company has an initial catastrophe fund or a risk reserve R^0, which in general is characterized by a random variable dependent on past catastrophic events. It is also possible to analyze needs for the future as R^0 a policy variable. For example, taking $R^0 = 0$ enables us to evaluate the capacity of the region to accumulate risk reserves in the future. Assume that the time span consists of $t = \overline{1, T}$, $T = 50$, time intervals. The stopping time τ is the time of the first catastrophe in the region within the time horizon T. The risk reserve (wealth) R^t of the pool at time $t = \overline{1, T}$ is calculated according to (22.4):

$$R^t = R^{t-1} + \sum_{j=1}^{m} \pi_j - \sum_{j=1}^{m} L_j^t(\omega_t)q,$$

where q defines the coverage of the pool in affected municipalities j at time t, π_j is the premium rate from the municipality j, and $L_j^t(\omega_t)$ is the loss (damage) at j caused by the simulated catastrophic event ω_t at time t. The value $L_j^t(\omega_t)$ depends on the event ω_t, the content of j, mitigation measures, and deterioration of the built environment. The analytical structure of the probability distribution of the random variable R^t is intractable; therefore, the methodology relies on Monte Carlo simulation.

Standard actuarial approaches calculate premiums in accordance with loss expectations. Therefore this study analyzed two policy options based on similar principles:

1. premiums based on the average damage over all municipalities (solidarity principle, bringing less exposed locations to pay premiums equal to more severely exposed ones, as in the spirit of the proposed insurance program); and

2. location-specific premiums based on average damage in the particular municipality, i.e., risk-based premiums.

However, the use of average losses may be misleading in the case of heavy-tailed distributions which are typical for catastrophic losses. The stochastic optimizations allows the analysis of different criteria and takes into account dependencies among location-specific losses. As an important example, a third policy option has been considered:

3. premiums calculated in a way that equalizes in a fair manner the risk of instability for the insurance company and the risk of premium overpayment for exposed municipalities. Besides this, it was important to analyze location-specific coverages and the amount of governmental compensation as a decision variable.

For option 3 it was assumed that the pool maximizes its wealth (risk reserves) taking into account the risks of the insolvency under the constraint on "fair" premiums. Fair premiums are defined according to the specified probability (say, once in every 100 years) of cases when paid premiums exceed actual claim sizes.

Accordingly, the goal function (22.6) for the pool at $t = \tau$, $R^0 = 0$, is defined as

$$f^\tau(x, V, \omega) = V + \gamma \min\{0, R^\tau(x, \omega) - V\} + \delta \min\{0, R^\tau(x, \omega)\}.$$

The stability of the welfare growth of municipalities can be written in the form of the chance constraints on overpayments

$$P\{(1 - q)L_j^t + L_j^t q_j < \pi_j\} \le p, \quad \sum_{j=1}^{m} q_j = q,$$

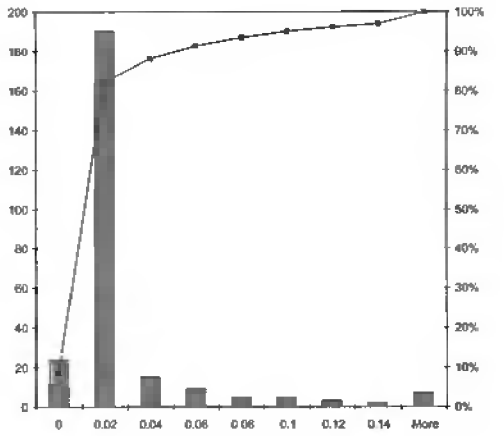


Figure 22.2. *Distribution of municipality-specific premiums (per building volume/ municipality, percent terms).*

where $x = (\pi_1, \ldots, \pi_n, q_1, \ldots, q_n), x \geq 0$, p is a given "safety" level. The difference $q - q_j$ defines the partial coverages of some municipalities, which generates the demand for further increase of the compensation by the government. The wealth of municipalities at $t = \tau$ changes due to the insurance program from $W_j^\tau - L_j^\tau$ to $W_j^{\tau+} = W_j^\tau - L_j^\tau + (1 - q + q_j)L_j^\tau - \pi_j$. The stochastic goal function (22.6) for municipality j at $t = \tau$ is

$$f_j^\tau(x, V_j, \omega) = V_j + \gamma_j \min\{0, W_j^{\tau+} - V_j\} + \delta_j \min\{0, (1 - q + q_j)L_j^\tau - \pi_j\}.$$

Figures 22.2–22.6 illustrate some numerical results. The number of simulations is shown on the vertical axis.

For option 1, where the burden of losses is equally distributed over the population, the simulation of catastrophic losses showed that the annual premium is equal to the flat rate of 0.02 monetary units (m.u.) per cubic meter of building.

For option 2, Figure 22.2 shows the distribution of municipality-specific premiums based on average damage in each municipality (or according to the municipality-specific risk). There is a prevailing number of municipalities (about 220) that have to pay 0.02–0.03 m.u., which is close to the flat rate of 0.02, as in option 1. About 20 municipalities are at no risk at all (0 rate). Municipalities more exposed to the risk have to pay 0.04 and higher rates (more than 50 municipalities).

Figure 22.3 shows the distribution of the insurers' reserve (cumulated at τ within 50 years) at premiums of option 2. The volume of capital is shown on the horizontal axis. The probability of insolvency (when the risk reserve accumulated up to the catastrophe is not enough to compensate incurred losses) is indicated on the right-hand ordinate axis. There is a rather high probability of "small" insolvency (values -90, -40 occurred 190 and 90 times out of 500 fast simulations, as discussed in sections 22.2 and 22.7). High solvency (more than 500 m.u.) occurred in about 10% of the simulations. The size of insolvency would represent the cost to the government to cover losses uncovered by the pool. Another option may be to transfer a fraction of losses to international financial markets, as analyzed in [17].

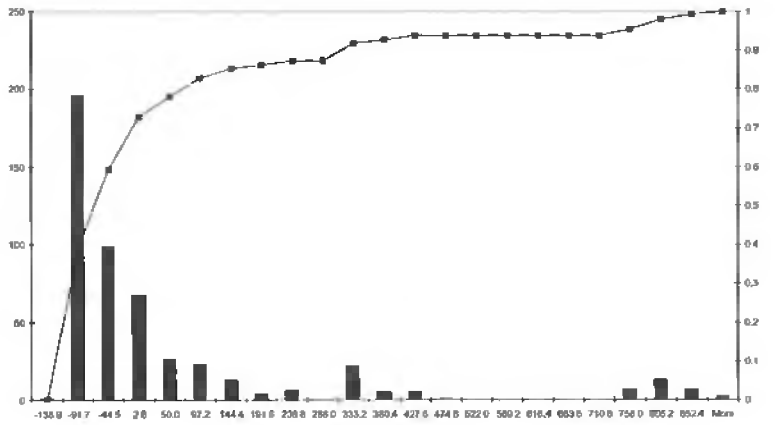


Figure 22.3. *Distribution of insurer's reserve, options* 1 *and* 2 (*thousands m.u.,* 50 *years*).

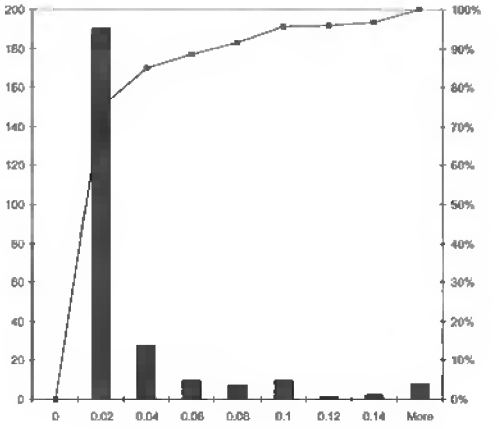


Figure 22.4. *Distribution of fair premiums, option* 3 (*per building volume/ municipality, percent terms*).

Figure 22.4 shows the distribution of premiums for option 3. According to this principle, most of the municipalities (190) have to pay close to the flat rate of 0.02–0.03 m.u. per cubic meter of a building. Rates of 0.04 and higher have to be paid by about 100 municipalities. In this case the highest premium rate is 0.5, which is much lower than the highest rate of 1.2 of option 2. The distribution of the insurer's reserve in Figure 22.5 indicates also the improvement of the insurer's stability: the frequency of insolvency is considerably reduced.

Figure 22.6 is very illustrative. For each municipality it shows the optional premiums to be paid: the flat premium rate of 0.02, the option 2 municipality-specific rate, and the fair premium of option 3. Many municipalities in all three options have to pay the premium rate, which is about the flat rate (0.015–0.03). For quite a number of municipalities in option 2, the rate significantly exceeds the flat rate. For these municipalities special attention should

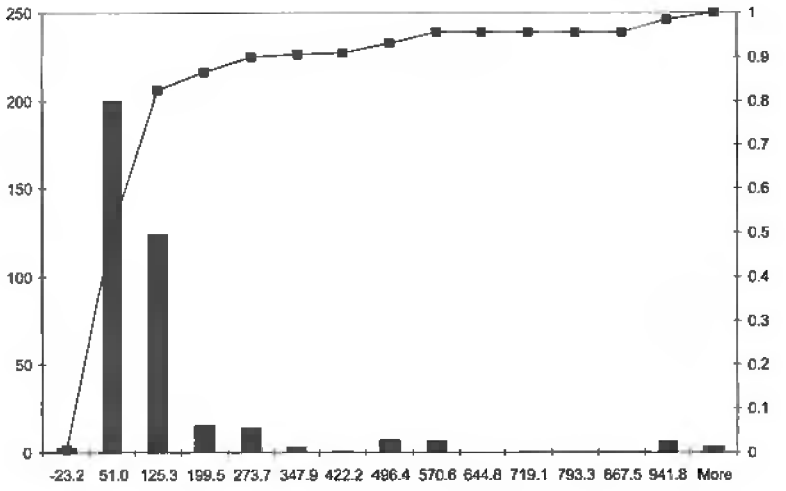


Figure 22.5. *Distribution of insurers' reserve, option* 3 *(thousands m.u., over* 50 *years).*

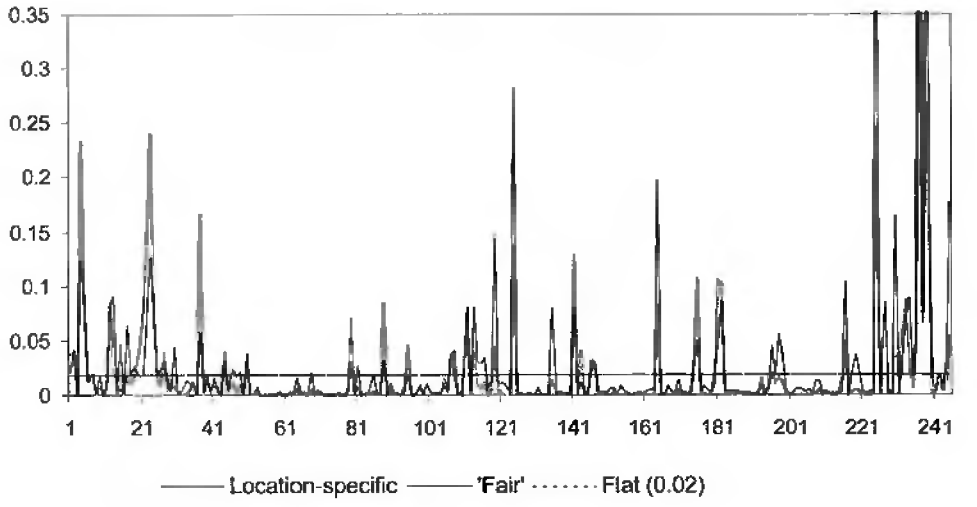


Figure 22.6. *Comparison of options: municipality-specific, fair, and flat* (0.02) *premiums.*

be given as to whether they are able to pay such high premiums. Option 3 allows one to take such individual constraints on overpayments into account and work out the efficient premiums both for insurer and municipalities.

22.7 The solution procedure

From the discussion in section 22.4, it follows that the welfare function $W(x, V)$ for the case study in the Tuscany region is a concave function assuming that τ and L_j^{τ} do not depend

on x. In this case the minimization of $W(x, V)$ can be approximately solved by linear programming methods (see general discussion in [15]). The resulting linear approximation may prove to have extremely large dimensions due to the large number of scenarios for estimating the function being optimized. The main challenge arises in the case when τ and L_j^τ are implicit functions of x. Then we can only use the stochastic quasi-gradient (SQG) methods (see [12, 6]). Let us outline only the main idea of these techniques. More details and further references are in [14, 15].

Assume that vector x incorporates not only risk management decision variables x but also V and decisions affecting the efficiency of the sampling itself (for more detail, see [16, 39]). An adaptive Monte Carlo optimization procedure (SQG method) searching for a solution minimizing $W(x)$ starts at any reasonable guess x^0. It updates the solution sequentially at steps $k = 0, 1, \ldots$, by the rule $x^{k+1} = x^k - \rho_k \xi^k$, where numbers $\rho_k > 0$ are predetermined step sizes satisfying the condition $\sum_{k=0}^{\infty} \rho_k < \infty$, $\sum_{k=0}^{\infty} \rho_k^2 = \infty$. For example, the specification $\rho_k = 1/(k+1)$ would formally suit. Random vector ξ^k is an estimate of the gradient $W_x(x)$ or its analogs for nonsmooth function $W(x)$. This vector is easily computed from random observations of $W(x)$. For example, let W^k be a random observation of $W(x)$ at $x = x^k$ and $\widetilde{W}^k$ be a random observation of $W(x)$ at $x = x^k + \delta_k h^k$. The numbers δ_k are positive, $\delta_k \to 0$, $k \to \infty$, and h^k is an independent observation of the vector h with components independent and uniformly distributed on $[-1, 1]$. Then ξ^k can be chosen as $\xi^k = [(\widetilde{W}^k - W^k)/\delta_k]h^k$. There is significant flexibility in choosing ξ^k for estimating the gradient of $W(x)$ at $x = x^k$. Some of them may lead to fast convergence; others produce slow oscillating behavior. For example, the straightforward estimation of function $\Psi(x)$ in section 22.2 is time consuming. But due to formula (22.1) we can use the following procedure. Consider any sequence of numbers $\mu_t > 0$, $t \geq 1$, $\sum_{t=1}^{\infty} \mu_t = 1$. Step $k+1$: choose t_k with a probability μ_t from set $t \in \{1, 2, \ldots\}$; generate $p_k \in [\underline{p}, \overline{p}]$ and simulate claim $B_k^{t_k}$ by a catastrophe model. Calculate $\xi^k = \mu_{t_k}^{-1}[p(1-p)^{t_k-1} V'_{t_k}(\min\{x^k, B_k^{t_k}\} - \pi t^k)]\eta^k$, where V'_t denotes the derivative of $V_t(\cdot)$, and $\eta^k = 1$ if $x^k \leq B_k^{t_k}$, and $\eta^k = 0$ otherwise. It is easy to see, e.g., from the discussion of the stochastic minimax problems in [12, p. 165], that $\mu_{t_k}^{-1}[p(1-p)^{t_k-1} V'_{t_k}(\min\{x^k, B_k^{t_k}\} - \pi t^k)]\eta^k$ is an estimate of $\Psi'(x^k)$; i.e., its expected value is $\Psi'(x^k)$. The rate of asymptotic convergence of this method (when the number of observations $k \to \infty$) is similar to other sampling based procedures.

Bibliography

[1] A. Amendola, Y. Ermoliev, and T. Ermolieva, *Earthquake risk management: A case study for an Italian region*, in Proceedings of the Second EuroConference on Global Change and Catastrophe Risk Management: Earthquake Risks in Europe, J. Linnerooth-Bayer and A. Amendola, eds., International Institute for Applied Systems Analysis, Laxenburg, Austria, 2000.

[2] A. Amendola, Y. Ermoliev, T. Ermolieva, V. Gitits, G. Koff, and J. Linnerooth-Bayer, *A system approach to modeling catstrophic risks and insurability*, Natural Hazards J., 21 (2000), pp. 381–393.

[3] K. ARROW, *The theory of risk-bearing: Small and great risks*, J. Risk Uncertainty, 12 (1996), pp. 103–111.

[4] P. ARTZNER, F. DELBAEN, J.-M. EBER, AND D. HEATH, *Coherent measures of risk*, Math. Finance, 9 (1999), pp. 203–228.

[5] S. BARANOV, B. DIGAS, T. ERMOLIEVA, AND V. ROZENBERG, *Earthquake Risk Management: Scenario Generator*, Interim Report 02-025, International Institute for Applied Systems Analysis, Laxenburg, Austria, 2002.

[6] J. BIRGE AND F. LOUVEAUX, *Introduction to Stochastic Programming*, Springer Series in Operation Research, Springer-Verlag, New York, 1997.

[7] K. BORCH, *Equilibrium in a reinsurance market*, Econ. J., 30 (1962), pp. 424–444.

[8] J. CUMMINS AND N. DOHERTY, *Can Insurer Pay for the "Big One"?*, Working Paper, Wharton Risk Management and Decision Processes Center, University of Pennsylvania, Philadelphia, 1996.

[9] G. DANTZIG, *The Role of Models in Determining Policy for Transition to a More Resilient Technological Society*, IIASA Distiguished Lecture Series 1, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1979.

[10] C. DAYKIN, T. PENTIKAINEN, AND M. PESONEN, *Practical Risk Theory for Actuaries*, Monogr. Statist. Appl. Probab. 53, Chapman and Hall, London, 1993.

[11] P. EMBRECHTS, C. KLUEPPELBERG, AND T. MIKOSCH, *Modeling Extremal Events for Insurance and Finance: Applications of Mathematics, Stochastic Modeling and Applied Probability*, Springer-Verlag, Heidelberg, 2000.

[12] Y. ERMOLIEV AND R. WETS, EDS., *Numerical Techniques of Stochastic Optimization: Computational Mathematics*, Springer-Verlag, Berlin, 1988.

[13] Y. ERMOLIEV, T. ERMOLIEVA, G. MACDONALD, AND V. NORKIN, *Insurability of catastrophic risks: The stochastic optimization model*, Optim. J., 47 (2000), pp. 251–265.

[14] Y. ERMOLIEV, T. ERMOLIEVA, G. MACDONALD, AND V. NORKIN, *Stochastic optimization of insurance portfolios for managing exposure to catastrophic risks*, Ann. Oper. Res., 99 (2000), pp. 207–225.

[15] Y. ERMOLIEV, T. ERMOLIEVA, G. MACDONALD, AND V. NORKIN, *Problems on insurance of catastrophic risks*, Cybernet. Syst. Anal., 2 (2001), pp. 220–234.

[16] T. ERMOLIEVA, *The Design of Optimal Insurance Decisions in the Presence of Catastrophic Risks*, Interim Report 97-028, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1997.

[17] T. ERMOLIEVA, Y. ERMOLIEV, J. LINNEROOTH-BAYER, AND I. GALAMBOS, *The role of financial instruments in integrated catastrophic flood management*, in Conference Proceedings of the Eighth Annual Conference of the Multinational Finance Society, Garda, Italy, P. Theodossiou, ed., School of Business, Rutgers University, Camden, NJ, 2001.

[18] T. ERMOLIEVA, Y. ERMOLIEV, AND V. NORKIN, *Spatial Stochastic Model for Optimization Capacity of Insurance Networks under Dependent Catastrophic Risks: Numerical Experiments*, Interim Report 97-028, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1997.

[19] K. FROOT, *The Limited Financing of Catastrophe Risk: An Overview*, Harvard Business School, Cambridge, MA, 1997.

[20] O. GIARINI AND H. LOUBERG, *The Diminishing Returns of Technology*, Pergamon Press, Oxford, UK, 1978.

[21] C. GILBER AND C. GOUY, *Flood management in France*, in Flood Response and Crisis Management in Western Europe: A Comparative Analysis, U. Rosenthal and P. Hart, eds., Springer-Verlag, Berlin, 1998.

[22] V. GINSBURG AND M. KEYZER, *The Structure of Applied General Equilibrium Models*, MIT Press, Cambridge, MA, 1997.

[23] J. GRANDELL, *Aspects of Risk Theory*, Springer Series in Statistics: Probability and Its Applications, Springer-Verlag, Berlin, 1991.

[24] IIASA, *Proposal for the Project on Flood Risk Management Policy in the Upper Tisza Basin: A System Analytical Approach*, International Institute for Applied Systems Analysis, Laxenburg, Austria, 2000.

[25] *The Impact of Catastrophes on Property Insurance*, Insurance Service Office, New York, 1994.

[26] N. JOBST AND S. ZENIOS, *The tail that wags the dog: Integrating credit risk in asset portfolios*, J. Risk Finance, 3 (2001), pp. 31–43.

[27] N. JOBST AND S. ZENIOS, *The Tail That Wags the Dog: Integrating Credit Risk in Asset Portfolios*, Working Paper 01-24, Wharton Financial Institutions Center, University of Pennsylvania, Philadelpha, 2001.

[28] P. KLEINDORFER AND H. KUNREUTHER, *The Complementary Roles of Mitigation and Insurance in Managing Catastrophic Risks*, Working Paper 98-14, Wharton School, Center for Financial Institutions, University of Pennsylvania, Philadelphia, 1997.

[29] H. KONNO AND H. YAMAZAKI, *Mean absolute deviation portfolio optimization model and its application to Tokyo stock market*, Management Sci., 37 (1991), pp. 519–531.

[30] H. KUNREUTHER AND R. ROTH, *Paying the Price: The Status and Role of Insurance against Natural Disasters in the United States*, Joseph Henry Press, Washington, DC, 1998.

[31] J. LINNEROOTH-BAYER AND A. AMENDOLA, *Global change, catastrophic risk and loss spreading*, Geneva Papers Risk Insurance, 25 (2000), pp. 203–219.

[32] L. MacKellar and T. Ermolieva, *The IIASA Social Security Project Multiregional Economic-Demographic Growth Model: Policy Background and Algebraic Structure*, Interim Report 99-007, International Institute for Applied Systems Analysis, Laxenburg, Austria, 1997.

[33] H. Markowitz, *Mean Variance Analysis in Portfolio Choice and Capital Markets*, Oxford, UK, Blackwell, 1987.

[34] *Climate Change and Increase in Loss Trend Persistence*, Technical Report, Munich RE, Munich, Germany, 1999.

[35] National Research Council, *National Disaster Losses: A Framework for Assessment*, National Academy Press, Washington, DC, 1999.

[36] V. Petrini, *Pericolosita' sismica e prime valutazioni di rischio in Toscana*, Report, CNR/IRRS, Milan, Italy, 1995.

[37] J. Pollner, *Catastrophe Risk Management: Using Alternative Risk Financing and Insurance Pooling Mechanisms, Finance, Private Sector and Infrastructure Sector Unit*, Working Paper 2560, Caribbean Country Department, Latin America and the Caribbean Region, World Bank, Washington, DC, 2000.

[38] A. Prekopa, *Stochastic Programming*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.

[39] E. L. Pugh, *A gradient technique of adaptive Monte Carlo*, SIAM Rev., 8 (1966), pp. 346–355.

[40] T. Rockafellar and S. Uryasev, *Optimization of conditional value-at-risk*, J. Risk, 2 (2000), pp. 21–41.

[41] J. M. Stone, *A theory of capacity and the insurance of catastrophe risks*, J. Risk Insurance, 40 (1973), pp. 231–244, 339–355.

[42] V. Rozenberg, T. Ermolieva, and M. Blizorukova, *Modeling Earthquakes via Computer Programs*, Interim Report 01-068, International Institute for Applied Systems Analysis, Laxenburg, Austria, 2001.

[43] G. Walker, *Current developments in catastrophe modelling*, in Financial Risks Management for Natural Catastrophes, N. Britton and J. Oliver, eds., Griffith University, Brisbane, Australia, 1997.

Chapter 23

Refinancing Mortgages in Switzerland

Karl Frauendorfer and Michael Schürle**

23.1 Introduction

Savings accounts, sight deposits, and mortgages make up a significant portion of a bank's assets and liabilities. According to statistics published by the Swiss National Bank, their average share in the balance sheet totals of all financial institutions in Switzerland amounts to 20% for savings and 53% for mortgages in 2000. Typically, a large percentage of these products consists of so-called nonmaturing accounts that can be characterized as follows. First, there is no contractual maturity; i.e., clients may withdraw their investments or repay their mortgages, respectively, at any point in time at no penalty. Second, the customer rate is not indexed to money or capital market rates but may be fixed by the bank as a matter of policy (in contrast to adjustable rate mortgages, as they are known in the United States). As a consequence, the volume of these positions may fluctuate heavily as clients react to changes in the customer rate, the relative attractiveness of alternative investment or financing opportunities due to rising or falling interest rates, etc.

23.1.1 Specific problems of nonmaturing accounts

In the case of variable-rate (nonfixed) mortgages, the change in volume is positively correlated with interest rates. When the latter are low, a sharp drop in demand can be observed since clients switch to fixed-rate mortgages to hedge themselves against a future interest increase (*prepayment risk*). On the contrary, the volume of savings accounts grows since their yields become relatively attractive compared to short-term securities, and even some institutional investors like pension funds prefer these deposits to direct investments in money

*Institute of Operations Research, University of St. Gallen, CH-9000 St. Gallen, Switzerland (karl.franendorfer @unisg.ch, michael.schuerle@unisg.ch).

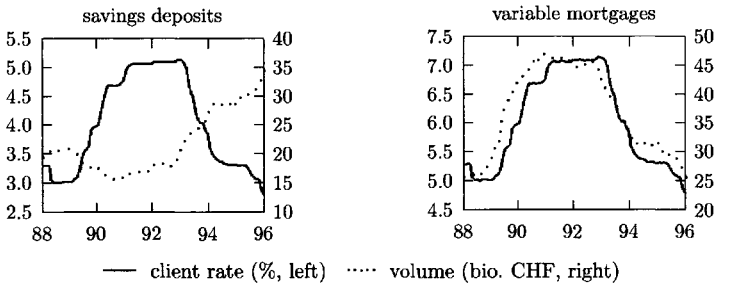


Figure 23.1. *Negative and positive correlation between client rate and volume.*

market instruments.

During a period of high interest rates, investors shift their assets from savings accounts to bonds with long maturities (*withdrawal risk*). This results in a negative correlation between yields and volume change. At the same time, homeowners' demand for variable-rate mortgages rises significantly. For example, while nonfixed mortgages represented approximately 90% of the entire mortgage volume of the largest Swiss cantonal bank in 1985, their percentage decreased with falling interest rates to less than 60% in 1989. Three years later, a new high of 80% was reached after an increase in market rates.

As the numbers above indicate, a bank is usually unable to refinance its mortgages directly by savings deposits since in general the latter have a much smaller percentage of the balance sheet total. Moreover, the volumes of both products fluctuate "out of phase" as a result of the different correlations with the level of interest rates (see Figure 23.1). Therefore, mortgages must be refinanced on the money and capital market at higher rates.

The challenge for the management is to find a mix of fixed-income instruments which not only minimizes the funding costs but also takes into account the prepayment risk that a significant portion of the volume under management is withdrawn since this would result in a surplus of liabilities over the assets (i.e., the portfolio of instruments with fixed maturities raised in the money and capital market versus variable mortgages). As an additional complication, there is a political cap on the mortgage rate in Switzerland, and numerous banks were not able to refinance their mortgages at a positive margin during the early 1990s.

23.1.2 Static versus dynamic approaches

The current practice in the management of nonmaturing accounts is the determination of a so-called replicating portfolio that mimics the behavior of the underlying position. To this end, one constructs a portfolio of fixed income securities whose return replicates the client rate of the relevant asset or liability position plus margin. Transaction costs remain low since liquid money market instruments and swaps are used that are held until maturity to avoid rebalancing ("buy and hold").

While maturing funds are always renewed at the same maturity, instruments are bought or sold at constant proportions whenever the volume of the position under management increases or decreases due to changes in customer demand. These weights are derived by minimizing the tracking error (i.e., the difference between the average portfolio rate plus

margin and the client rate) over a historic sample period under the constraint that the volume of the replicating portfolio matches that of the target account at all points in time. Therefore, prepayment and withdrawal risks are implicitly taken into account.

In this way, uncertain cash flows are transformed into (apparently) certain ones, allowing the bank to manage them like normal positions with fixed maturities. However, the considerable risk of an incorrect transformation remains. For example, different historic sample periods for the determination of a replicating portfolio may result in substantially different portfolio weights. While the minimization of the tracking error aims at the stabilization of the margin over time, it does *not* provide minimal funding costs or even guarantee a positive margin at all.

The approach is also static in the sense that it does not incorporate the possibility of *future* changes in random data such as interest rates and volume and their impact on the optimal portfolio composition. Therefore, the question arises of whether a *dynamic* policy with active reactions to changes in market rates and customer demand might increase the bank's profit or reduce its refinancing costs, respectively. In particular, it remains to be clarified whether the correlation between interest rates and volume can be exploited more appropriately to manage the different risks associated with nonmaturing account positions.

Multistage stochastic programs take all these aspects into account. Based on assumptions about the (joint) dynamics of relevant risk factors that are usually described by stochastic processes, representative scenarios for their future outcomes are generated. In the context of asset and liability management (ALM) problems, the latter quantify the impact of changes in the risk factors on the return of investment or refinancing strategies. Transactions may take place at discrete points in time over some finite planning horizon for the correction of an initial policy, given new observations of random data in each scenario. Furthermore, various constraints can be taken into account. For example, this allows the incorporation of liquidity restrictions in the market or limits for the risk exposure with respect to certain positions.

In general, stochastic optimization models result in large-scale programs since they have to include a large number of scenarios to reflect the entire universe of possible future outcomes of risk factors and cash flows. Since multistage programs suffer from an exponential growth in problem size with respect to the number of periods under consideration, the first models for financial planning that appeared in the early 1980s [25, 27] were restricted to a two-stage structure due to limitations of the computational resources available at that time.

Today, the dramatic improvement of powerful hardware as well as the development of efficient algorithms, in particular if they exploit the special structure and high sparsity inherent to stochastic programs, provide the basis for the solution of problems with even some million scenarios, like in the pension fund model of Gondzio and Kouwenberg [20]. Moreover, the inclusion of new theoretical models from the financial literature and related empirical evidence allows a more appropriate modeling of the complex dynamics of risk factors.

Among the first successful commercial multistage applications are the Towers Perrin–Tillinghast ALM system of Mulvey, Gould, and Morgan [29], the fixed-income portfolio management models of Zenios [32] and Beltratti, Consiglio, and Zenios [1], or the well-known Russell–Yasuda Kasai financial planning tool of Cariño et al. [3, 4, 5], to mention just

a few. Numerous related models have been derived from the latter like the InnoALM system [19] that exploits sophisticated econometric models and scenario dependent correlation matrices to specifically consider extreme market events. For an extensive survey on financial applications, see the book edited by Ziemba and Mulvey [33].

Motivated by the difficulties in the management of nonmaturing account positions and given their significant importance for most banks in Switzerland, we developed a stochastic optimization model for nonfixed mortgages that has been in use by a major Swiss bank since 1995 and was extended to savings accounts two years later [16]. Other versions have been implemented, e.g., for pension funds or cash management at an insurance company where one must deal with the seasonal behavior of premium payments, uncertain investment yields, and stochastic liabilities due to claims.

In the next section, we introduce a simplified formulation of the stochastic optimization model for refinancing variable mortgages. After a discussion of relevant characteristics of the underlying risk factors, i.e., interest rates and volume, section 23.3 describes two term structure models for different planning horizons and an autoregressive process for the volume dynamics. Barycentric approximation as a special scenario generation method that is particularly well suited for the given problem structure is outlined in section 23.4. Practical experience and results from a case study are reported in section 23.5. Section 23.6 summarizes the most relevant features of the model and gives an outlook to possible directions for its improvement.

23.2 Formulation of the optimization model

23.2.1 Framework for decision making under uncertainty

We assume that the evolution of random data is described by a discrete-time stochastic process $\omega := (\omega_t;\ t = 1, \ldots, T)$ defined in a probability space $(\Omega, \mathcal{F}, P)$. The sample space can be decomposed with respect to time as $\Omega := \Omega_1 \times \cdots \times \Omega_T$ with $\Omega_t \subset \mathbb{R}^{M_t}$. The filtration $\mathcal{F} := \{\mathcal{F}_t;\ t = 1, \ldots, T\}$ specifies the information structure and satisfies $\mathcal{F}_t \subset \mathcal{F}_{t+1}$. Each σ-field $\mathcal{F}_t := \sigma\{\omega^t\}$ is generated by observations $\omega^t := (\omega_1, \ldots, \omega_t)$ of the data process that are known at time t, and $P_t : \mathcal{F}_t \to [0, 1]$ denotes the corresponding conditional probability measure.

Restricting ourselves to the linear case, the underlying decision problem can be outlined as follows. At any point in time $t = 0, \ldots, T$, we have to decide on a portfolio composition x_t to meet an uncertain target h_{t+1} in the future (here, the volume of variable mortgages). The value of the portfolio after one period is given by $T_t \cdot x_t$, where the elements of the matrix T_t represent, e.g., prices of individual positions that may be affected by the outcomes of some risk factors $\omega_{t+1} \in \Omega_{t+1}$. Since it is likely that the portfolio fails to meet the target, a correction x_{t+1} may be required after a realization of ω_{t+1} has been observed to compensate for the discrepancy $h_{t+1} - T_t \cdot x_t$. Such a recourse action x_{t+1} is penalized by the fixed matrix W_{t+1}.

The objective is to minimize the cost $c_t' \cdot x_t$ in the current period t plus the expected cost $E_{t+1}\phi_{t+1}(x^t, \omega^{t+1}) = \int_{\Omega_{t+1}} \phi_{t+1}(x^t, \omega^{t+1}) dP_{t+1}$ for future transactions that depends on the sequence of observations $\omega^{t+1} := (\omega^t, \omega_{t+1})$ and earlier decisions $x^t := (x_0, \ldots, x_t)$.

This can be stated recursively for $t = T, \ldots, 1$ in terms of *value functions* as

$$\begin{aligned} \phi_t(x^{t-1}, \omega^t) &:= \min c_t'(\omega^t) \cdot x_t(\omega^t) + \int_{\Omega_{t+1}} \phi_{t+1}(x^t, \omega^{t+1}) dP_{t+1}(\omega_{t+1}|\omega^t) \\ &\text{s.t. } W_t \cdot x_t(\omega^t) = h_t(\omega^t) - T_{t-1}(\omega^t) \cdot x_{t-1}(\omega^{t-1}),\ x_t(\omega^t) \geq 0, \end{aligned} \tag{23.1}$$

with the boundary condition $\phi_{T+1}(\cdot) := 0$, whereas the problem for the first stage is

$$\begin{aligned} \phi_0 &:= \min c_0' \cdot x_0 + \int_{\Omega_1} \phi_1(x_0, \omega_1) dP_1(\omega_1) \\ &\text{s.t. } W_0 \cdot x_0 = h_0,\ x_0 \geq 0. \end{aligned} \tag{23.2}$$

A decision x_t may depend only on information ω^t available at time t and not on future outcomes of random data $\omega_{t+1}, \ldots, \omega_T$. This property is known as *nonanticipativity*. For simplicity of notation, we do not stress the dependency of x_t on ω^t in what follows but assume implicitly that $x_t \equiv \{x_t|\mathcal{F}_t\}$ is adapted to the filtration $\mathcal{F}_t$.

Due to linearity of the objective function and constraints, the multistage stochastic program given by (23.1) and (23.2) is convex. Its properties are widely discussed in the literature, e.g., see Part I of this volume or the textbook by Birge and Louveaux [2]. Here, we focus on the special case where the vectors $\omega_t \in \Omega_t \subset \mathbb{R}^{M_t}$ of random data can be decomposed into two components: $\eta_t \in \Theta_t \subset \mathbb{R}^{K_t}$ affects the coefficients in the objective; $\xi_t \in \Xi_t \subset \mathbb{R}^{L_t}$ influences the demand on the right-hand side of constraints with $M_t = K_t + L_t$. Then, the value function given by (23.1) is a saddle function for all $t = 1, \ldots, T$, and the corresponding optimization problem can be solved if the following conditions hold:

1. The support of $\omega_t = (\eta_t, \xi_t)$ is covered by compact and convex sets $\Omega_t = \Theta_t \times \Xi_t$.
2. $c_t(\eta^t)$ and $h_t(\xi^t)$ are linear affine in their respective random vectors $\eta^t := (\eta_1, \ldots, \eta_t)$, $\xi^t := (\xi_1, \ldots, \xi_t)$, and the matrices T_{t-1} are deterministic.

In other words, the value function (23.1) takes on the form

$$\begin{aligned} \phi_t(x^{t-1}, \omega^t) &:= \min c_t'(\eta^t) \cdot x_t + \int_{\Omega_{t+1}} \phi_{t+1}(x^t, \omega^{t+1}) dP_{t+1}(\omega_{t+1}|\omega^t) \\ &\text{s.t. } W_t \cdot x_t + T_{t-1} \cdot x_{t-1} = h_t(\xi^t),\ x_t \geq 0. \end{aligned} \tag{23.1'}$$

Since $\phi_{T+1}(\cdot) := 0$, this implies that the problem for the final stage T is convex in (x^{T-1}, ξ^T) and concave in η^T. When calculating the expectations $E_{t+1}\phi_{t+1}$ in the remaining stages $T-1, \ldots, 1$, the probability measures P_{t+1} depend on ω^t. As a consequence, the saddle property of the value function in T may not be "inherited" to the previous stages due to the integration with respect to $P_{t+1}(\cdot|\omega^t)$. However, if the corresponding distribution functions may be represented, e.g., in the form $Q_{t+1}\big(\omega_{t+1} + H_{t+1}(\omega^t)\big)$, where Q_{t+1} is independent of ω^t and H_{t+1} is a linear mapping, then it can be shown that the relevant expectation functionals $E_{t+1}\phi_{t+1}(x^t, \omega^{t+1})$ are continuous saddle functions [13]. This leads to the following requirement:

3. The distribution function of $P_t(\cdot|\omega^{t-1})$ depends linearly on the past.

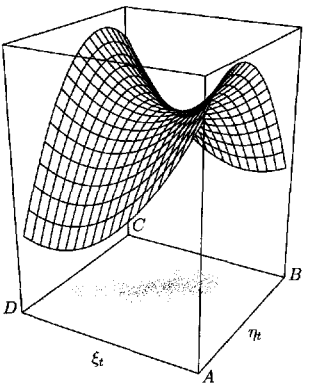


Figure 23.2. *Saddle property of the value function.*

In particular, this covers the case that the distribution of risk factors is independent of prior outcomes. If conditions 1–3 are fulfilled, the value functions $\phi_t(x^{t-1}, \eta^t, \xi^t)$ are convex in (x^{t-1}, ξ^t) and concave in η^t for $t = 1, \ldots, T$, which is called the *entire convex case*. (See Figure 23.2 for the case of $K_t = L_t = 1$.) These conditions are sufficient, and the saddle property may also be given for more general problem types.

23.2.2 Specification of the funding problem

Given the description of the mortgage funding problem in section 23.1.1, the formulation of the optimization model is straightforward: let $\mathcal{D} = \{1, 2, \ldots, D\}$ denote the set of maturity dates for fixed-income instruments held in the current portfolio from decisions in the past (D is the longest available maturity). Standard maturities traded in the money and capital market which can be used for refinancing are given by the set $\mathcal{D}^S \subseteq \mathcal{D}$. Due to liquidity restrictions in the Swiss interbank market which must be observed by large banks, transaction costs (bid-ask spreads) increase when a certain volume is exceeded. To this end, the amount to be refinanced in each maturity is split into several *tranches*. The maximum number of possible tranches for maturity d is given by $I^d, \mathcal{I}^d := \{1, \ldots, I^d\}$ is a corresponding index set, and $x^d_{i,t}$ denotes the transaction volume in tranche $i \in \mathcal{I}^d, d \in \mathcal{D}^S$.

The coefficient $\varphi^d_{i,t}$ quantifies the costs for raising \$1 in the ith tranche of maturity d at time t. It contains the net present value of all interest payments until maturity of the instrument, transaction costs, as well as penalty spreads for exceeding the tranche limits. Since the planning horizon must be truncated at time T while the bank's business is (hopefully) continued beyond that date, the present value of all cash flows which occur after the terminal stage, i.e., outstanding interest payments and repayment of the face value discounted at the yield curve in T, is also included to incorporate end effects.

For $t > 0$ these coefficients depend on future interest rates and, hence, are uncertain. It is assumed that the $\varphi^d_{i,t}$ for all $d \in \mathcal{D}^S$ at time t are functions of a random vector $\eta_t \in \mathbb{R}^{K_t}$. As discussed in what follows, the elements of η_t are the state variables of a term structure model which describes the evolution of the yield curve by a K_t-dimensional stochastic process. A formal specification of the functional relationship between the interest rate risk

factors η_t and $\varphi^d_{i,t}$ is omitted here because its notation becomes rather cumbersome.

Analogously, the mortgage volume υ_t depends on the random vector $\xi_t \in \mathbb{R}^{L_t}$. It may be correlated with η_t to reflect a dependency between interest rates and volume. With the objective to minimize the expected discounted refinancing costs over the planning horizon T, the corresponding multistage stochastic program is

$$\min \int_\Omega \left\{ \sum_{t=0}^{T} \sum_{d \in \mathcal{D}^S} \sum_{i \in \mathcal{I}^d} \varphi^d_{i,t}(\eta_t) \cdot x^d_{i,t} \right\} dP(\omega) \tag{23.3}$$

subject to

$$x^d_t - x^{d+1}_{t-1} = 0, \quad t = 0, \ldots, T \quad \forall d \notin \mathcal{D}^S, \text{ a.s.}, \tag{23.3.1}$$

$$x^d_t - x^{d+1}_{t-1} - \sum_{i \in \mathcal{I}^d} x^d_{i,t} = 0, \quad t = 0, \ldots, T \quad \forall d \in \mathcal{D}^S, \text{ a.s.}, \tag{23.3.2}$$

$$\sum_{d \in \mathcal{D}} x^d_t = \upsilon_t(\xi_t), \quad t = 0, \ldots, T, \text{ a.s.}, \tag{23.3.3}$$

$$\ell^{l,d}_{i,t} \le x^d_{i,t} \le \ell^{u,d}_{i,t}, \quad t = 0, \ldots, T \quad \forall i \in \mathcal{I}^d, \ \forall d \in \mathcal{D}^S, \text{ a.s.}, \tag{23.3.4}$$

$$x^d_{i,t} \ge 0; \ x^d_{i,t} \equiv \{x^d_{i,t} | \mathcal{F}_t\}, \quad t = 0, \ldots, T \quad \forall i \in \mathcal{I}^d, \ \forall d \in \mathcal{D}^S, \text{ a.s.}, \tag{23.3.5}$$

$$x^d_t \in \mathbb{R}; \ x^d_t \equiv \{x^d_t | \mathcal{F}_t\}, \quad t = 0, \ldots, T \quad \forall d \in \mathcal{D}, \text{ a.s.} \tag{23.3.6}$$

The budget constraint (23.3.1) ensures that the position x^d_t maturing after d periods equals the corresponding value in the previous period for nontraded maturity dates while (23.3.2) corrects it by the new transactions in t for traded ones. Herein, x^d_{-1} denotes the amount in the initial portfolio held in maturity d. Constraint (23.3.3) requires that the complete portfolio match the stochastic mortgage volume υ_t at all points in time.

Lower and upper limits $\ell^{l,d}_{i,t}$ and $\ell^{u,d}_{i,t}$ for tranches with different penalty spreads to reflect liquidity restrictions are given by (23.3.4). Obviously, the absolute amount that can be refinanced in maturity d can be controlled by a corresponding number of tranches i^d. All tranches with low indices are filled to their limits first in the optimal solution since the spreads are strictly increasing. Nonanticipativity constraints (23.3.5) and (23.3.6) require that decisions are adapted to the filtration $\mathcal{F}_t$. Moreover, the restrictions (23.3.1)–(23.3.6) at stage $t = 1, \ldots, T$ must hold almost surely, i.e., for all observations of the uncertain interest rate and volume risk factors $\omega_t := (\eta_t, \xi_t)$ with positive probability.

Since we distinguish between stochastic factors η_t that affect only coefficients in the objective and those ξ_t that occur solely in the constraints with deterministic left-hand sides, it can easily be seen that the mortgage funding problem above has the same structure as the special type of multistage stochastic programs given by (23.1′) and (23.2). This is a particularly useful property for the solution method introduced in section 23.4.

23.3 Modeling the dynamics of risk factors

Since the optimization problem (23.3) contains uncertain coefficients, we must specify stochastic processes that describe the dynamics of the relevant risk factors, i.e., interest rates of traded maturities and the volume of the mortgage position under management.

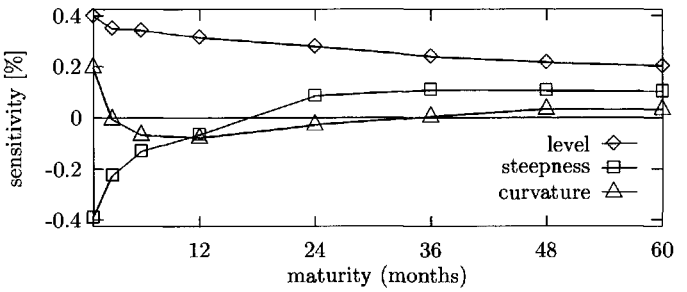


Figure 23.3. *Impact of principal components on changes in yields.*

The corresponding process parameters are then estimated by suitable statistical procedures. Alternative models should be compared by means of econometric tests which allow the rejection of inappropriate specifications. In this way, one can identify the model that provides the best explanation of the empirically observed data among different alternatives.

23.3.1 Term structure models for interest rate scenarios

Interest rates exhibit a number of characteristic features that should be taken into account to model their dynamics realistically. For example, principal component analysis reveals that three stochastic factors are sufficient to explain more than 95% of interest rate volatility. These factors can be associated with changes in the *level*, *steepness*, and *curvature* of the yield curve, e.g., see Figure 23.3 for Swiss Franc Euromarket rates [31]. Moreover, one can observe that interest rates fluctuate around a long-term mean during an economic cycle. This *mean reversion* property is incorporated into most term structure models that have been introduced in the financial literature over the last two decades; e.g., see James and Webber [23] for a comprehensive introduction.

The relevance of these different aspects depends on the underlying application and/or the desired planning horizon. For example, when a portfolio of short-term securities is managed for a horizon of up to one year like in cash management problems, it should be taken into account that the short end of the yield curve often shows complex and quickly varying shapes. This can be reflected well using three principal components as risk factors that are standard normally distributed and orthogonal by construction while the mean reversion property may be neglected. The sensitivities of relevant interest rates with respect to the principal components can easily be derived by means of most statistical software tools.

On the other hand, this approach implies a nonstationary distribution of interest rates. As a consequence, the model generates values that are outside of the usual range or may even become negative when the planning horizon is extended. A simple way to model mean reversion of a risk factor η_t is to introduce a drift term

$$\frac{d\eta_t}{dt} = \kappa(\theta - \eta_t), \qquad \kappa > 0. \tag{23.4}$$

If $\eta_t < \theta$ ($\eta_t > \theta$) at time t, then the left-hand side of this equation is positive (negative), which will cause η_t to increase (decrease) toward its mean reversion level θ at a speed controlled by parameter κ. Rewriting (23.4) and adding a volatility term where noise is

a function of a Wiener process with increment dz_t as is usual in modeling financial data yields the (continuous-time) process

$$d\eta_t = \kappa(\theta - \eta_t)dt + \sigma\eta_t^{\gamma} dz_t. \tag{23.5}$$

The instantaneous volatility $\sigma\eta_t^{\gamma}$ may depend on the current level of the factor if $\gamma > 0$ to reflect a possible heteroscedasticity of interest rates. Moreover, this prevents the process from becoming negative since in continuous time the volatility term becomes zero when $\eta_t = 0$ and only the positive drift remains.

Given K stochastic processes of type (23.5) for one or more risk factors in a classical term structure model, Ito's lemma can be used to derive a process for the price of a discount bound. By construction of a hedge portfolio consisting of $K + 1$ instruments with different maturities, one obtains a partial differential equation (PDE) for the term structure using a no-arbitrage argument. However, the latter can be solved analytically only for some special cases. Example are models of the affine type where all K factors (i) follow a process of the form (23.5) with identical exponent $\gamma \in \{0, 0.5\}$, (ii) are orthogonal, and (iii) sum up to the instantaneous rate r_t, i.e., the yield for an infinitely short holding period [17].

To overcome these restrictions, we implemented several exponential functions for an interpolation of the term structure instead of an arbitrage-free PDE. For the ease of exposition, only the rather simple form

$$R(s_t, l_t, d) = (s_t + \beta_1 \cdot d) \cdot e^{-\beta_2 \cdot d} + l_t \tag{23.6}$$

is depicted here, where $R(s_t, l_t, d)$ denotes the spot rate for maturity d. While the fix parameters β_1, β_2 control the shape of the yield curve, we model the long rate l_t and the spread $s_t := r_t - l_t$ by the stochastic processes

$$\begin{aligned} ds_t &= \kappa_s(\theta_s - s_t)dt + \sigma_s dz_{1,t}, \\ dl_t &= \kappa_l(\theta_l - l_t)dt + \sigma_l l_t^{\gamma} dz_{2,t}. \end{aligned} \tag{23.7}$$

The volatility of the long rate may depend on the current level of l_t when $\gamma > 0$ to incorporate a possible heteroscedasticity. Obviously, the process for the spread must allow for negative values to reflect normal ($s_t < 0$) as well as inverse term structures ($s_t > 0$). Although this does not preclude negative interest rates when the spread becomes sufficiently large, they are extremely unlikely for realistic parameter estimates.

The process specification (23.7) resembles in some sense the model of Schaefer and Schwartz [30] that uses the same state variables but assumes $\gamma = 0.5$ and no correlation between both Wiener processes to derive an analytical approximation for the term structure PDE. The advantage of our approach is that we have more flexibility in the specification of yield curve functions than in conventional term structure models. Furthermore, we may choose any value for the correlation $\rho = dz_{1,t} \cdot dz_{2,t}$ between the Wiener processes and the volatility exponent γ. Using discrete time approximations of (23.7)

$$\begin{aligned} s_{t+1} &= s_t + \kappa_s(\theta_s - s_t)\Delta t + \sigma_s\sqrt{\Delta t}\epsilon_{1,t}, \\ l_{t+1} &= l_t + \kappa_l(\theta_l - l_t)\Delta t + \sigma_l l_t^{\gamma}\sqrt{\Delta t}\epsilon_{2,t}, \end{aligned} \tag{23.8}$$

parameters can easily be estimated for both processes separately with maximum likelihood (under the assumption that the residuals $\epsilon_{i,t} \sim \mathcal{N}(0, 1)$, $i = 1, 2$, are serially independent and uncorrelated with s_t and l_t). Herein, the long rate l_t is approximated by observations of the 5-year CHF Euromarket rate as the longest available (liquid) maturity and the spread s_t by the difference between the 1-month rate and the latter.

In view of the approximation method introduced in what follows for the solution of multistage stochastic programs, we restrict ourselves to fixed values $\gamma \in \{0, 1\}$ when we estimate the parameters of the discrete processes (23.8). Under these specifications, the saddle property of value functions discussed in section 23.2.1 will be preserved. An analysis of different historic sample periods reveals that $\gamma = 1$ provides higher likelihood values in most cases which supports the assumption of heteroscedastic interest rates. After calibration of the processes, an estimate for ρ is obtained from the cross correlation between the residuals. Finally, parameters of the yield curve interpolation function (here, β_1, β_2) are determined that allow for the best fit of (23.6) to the observed rates of all maturities in the historic sample by quadratic minimization.

23.3.2 Specification of the volume process

At first sight, the specification and calibration of stochastic processes for the variable mortgage volume seems to be easier than for interest rates since it is directly observable and, in contrast to the term structure, does not depend on several latent factors. However, data for the estimation and assessment of alternative models are often difficult to obtain in practice. The Swiss National Bank publishes only an aggregate of variable and fixed mortgages in its monthly reports, and the percentage of nonfixed mortgages is available only on a yearly basis since 1996. Furthermore, fluctuations in mortgage demand often depend on the type of bank. For example, the clientele of large commercial banks in Switzerland consists mainly of an urban population that tends to react more actively to changes in the economy than customers of smaller cooperative banks in the countryside. As a consequence, any volume model has to be calibrated individually to the specific bank situation.

According to the problem description in section 23.1.1, it seems plausible that the demand for nonmaturing accounts depends on the level of interest rates. In the case of savings accounts, Schürle [31] found that a trend-stationary process with two factors of a term structure model as explanatory variables provides a good description for real deposit positions. However, when the implementation of the mortgage funding model started in 1993, we did not have a sufficiently large data set at our disposal to support such a hypothesis. Therefore, we simply model the volume by the autoregressive process

$$\widetilde{v}_t = a + \rho_v \widetilde{v}_{t-1} + \xi_t \tag{23.9}$$

to which a deterministic trend is added, i.e., $v_t = \widetilde{v}_t + bt$. The stochastic component (23.9) has a long-term mean of $a/(1 - \rho_v)$ if the process is stationary (i.e., $|\rho_v| < 1$). After correcting a sample time series by the trend bt, the parameters of the resulting standard AR(1) process can be easily estimated by means of most standard statistical software. A dependency on interest rates is taken into account by the correlation between the stochastic factor ξ_t and the residuals of the discrete-time interest rate processes (23.8).

23.4 Solution of multistage stochastic programs

The numerical difficulty in solving a problem of type (23.1) and (23.2) lies in the nested minimization and multidimensional integration of value functions. Since the latter are given only implicitly as the solution of stochastic programs with respect to the remaining stages, this integration cannot be performed analytically, and numerical techniques are required that can broadly be classified into simulation-based methods and bound-based approximations.

In the former case, random samples are drawn from the underlying probability distributions, e.g., to derive stochastic quasi gradients or for the application of stochastic decomposition algorithms. While the computational effort is independent of the dimension of random data, these approaches provide only *probabilistic* bounds for the discretization error. Loosely speaking, this is the error that results from replacing the entire universe of possible outcomes of random data by a relatively small set of scenarios.

In contrast to this, *bound-based approximation* methods partition the domain of random data into cells and determine representative points within them. The accuracy may be improved by adding new scenarios that result from partitioning the initial cells into smaller ones (*refinement*). For multistage problems, a careful control of the refinement process is necessary since the number of scenarios grows exponentially with the dimension size and the desired accuracy (*curse of dimensionality*). However, generalizations of the well-known inequalities of Jensen [24] and Edmundson/Madansky (see [28]) that exploit the saddle property of value functions discussed in section 23.2.1 provide *exact* lower and upper bounds to the original problem. (Bounds based on these inequalities and their refinements in the context of two-stage stochastic programming were introduced, e.g., in [21].) This allows a deliberate refinement for those scenarios and stages where the largest discretization error is observed to reduce the growth in problem size.

Since the evolution of interest rates and volume is modeled by low-dimensional processes, the underlying saddle property motivates the application of a bound-based scenario generation approach for the solution of the mortgage funding problem (23.3). To deal with the observed correlations between the risk factors, approximation schemes must take into account cross-moment information in a numerically efficient way. Such techniques have been developed, e.g., by Edirisinghe [6], Edirisinghe and Ziemba [8, 9, 10], and Frauendorfer [11, 13, 14]. See also [7] for a general survey on bound-based approximations.

23.4.1 Barycentric approximation

In what follows, we concentrate on the *barycentric approximation* technique that was originally introduced in [11] for two-stage stochastic programs and extend it to the multistage case. The basic idea is to replace the implicitly given value functions of type (23.1′) by two bilinear functions that can easily be integrated. Then, the best points where the original value function must be supported by its bilinear approximations to minimize the discretization error are the so-called generalized barycenters. These are determined with respect to cross simplices (or, briefly, $\times$-simplices), i.e., the Cartesian product of regular simplices, that cover the support of random data.

For the ease of exposition, we assume that the sets $\Theta_t(\omega^{t-1}) \subset \mathbb{R}^{K_t}$ and $\Xi_t(\omega^{t-1}) \subset \mathbb{R}^{L_t}$ that cover the support of η_t and ξ_t are regular simplices. Note that both may depend on prior observations ω^{t-1}, although this is not stressed in the notation for simplicity. Let

the vertices of Θ_t and Ξ_t be denoted by u_{ν_t}, $\nu_t = 0, \ldots, K_t$, and v_{μ_t}, $\mu_t = 0, \ldots, L_t$. The *barycentric weights* $\lambda_t(\eta_t) = \big(\lambda_{t,0}(\eta_t), \ldots, \lambda_{t,K_t}(\eta_t)\big)'$ of η_t with respect to Θ_t are those nonnegative weights that allow the representation of η_t as a linear combination of the vertices u_{ν_t} and sum up to one:

$$\begin{aligned} \lambda_{t,0} \quad + \quad \lambda_{t,1} \quad + \cdots + \quad \lambda_{t,K_t} \quad &= 1, \\ u_{t,0}\lambda_{t,0} + u_{t,1}\lambda_{t,1} + \cdots + u_{t,K_t}\lambda_{t,K_t} &= \eta_t. \end{aligned}$$

The barycentric weights $\tau_t(\xi_t) = \big(\tau_{t,0}(\xi_t), \ldots, \tau_{t,L_t}(\xi_t)\big)'$ of ξ_t with respect to Ξ_t are defined analogously. Hence $\lambda_t(\cdot)$, $\tau_t(\cdot)$ can be obtained as the solution of

$$U_t \cdot \lambda_t = \begin{pmatrix} 1 \\ \eta_t \end{pmatrix} \qquad \text{with } U_t = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ u_0 & u_1 & \cdots & u_{K_t} \end{pmatrix}, \tag{23.10}$$

$$V_t \cdot \tau_t = \begin{pmatrix} 1 \\ \xi_t \end{pmatrix} \qquad \text{with } V_t = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ v_0 & v_1 & \cdots & v_{L_t} \end{pmatrix}. \tag{23.11}$$

These weights may be used to derive the *generalized barycenters*

$$\xi_{\nu_t} = \big[\mathcal{M}_{\nu_t}(\{u_{\nu_t}\} \times \Xi_t)\big]^{-1} \cdot \sum_{\mu_t=0}^{L_t} v_{\mu_t} \int \lambda_{\nu_t}(\eta_t) \cdot \tau_{\mu_t}(\xi_t) dP_t(\eta_t, \xi_t | \omega^{t-1}), \tag{23.12}$$

$$\eta_{\mu_t} = \big[\mathcal{M}_{\mu_t}(\Theta_t \times \{v_{\mu_t}\})\big]^{-1} \cdot \sum_{\nu_t=0}^{K_t} u_{\nu_t} \int \lambda_{\nu_t}(\eta_t) \cdot \tau_{\mu_t}(\xi_t) dP_t(\eta_t, \xi_t | \omega^{t-1}) \tag{23.13}$$

with respect to the $\times$-simplex $\Theta_t \times \Xi_t$, where

$$\mathcal{M}_{\nu_t}(\{u_{\nu_t}\} \times \Xi_t) = \int \tau_{\mu_t}(\xi_t) dP_t(\eta_t, \xi_t | \omega^{t-1}), \qquad \nu_t = 0, \ldots, K_t, \tag{23.14}$$

$$\mathcal{M}_{\mu_t}(\Theta_t \times \{v_{\mu_t}\}) = \int \lambda_{\nu_t}(\eta_t) dP_t(\eta_t, \xi_t | \omega^{t-1}), \qquad \mu_t = 0, \ldots, L_t, \tag{23.15}$$

are the mass distributions induced by the probability measure P_t on the K_t-dimensional simplices $\Theta_t \times \{v_{\mu_t}\}$ and the L_t-dimensional simplices $\{u_{\nu_t}\} \times \Xi_t$, respectively. These mass distributions add up to one; i.e.,

$$\sum_{\nu_t=0}^{K_t} \mathcal{M}_{\nu_t}(\{u_{\nu_t}\} \times \Xi_t) = 1 \qquad \text{and} \qquad \sum_{\mu_t=0}^{L_t} \mathcal{M}_{\mu_t}(\Theta_t \times \{v_{\mu_t}\}) = 1.$$

Therefore, we may interpret (23.14) and (23.15) as probabilities assigned to the points (u_{ν_t}, ξ_{ν_t}) with probability $\mathcal{M}_{\nu_t}(\{u_{\nu_t}\} \times \Xi_t)$ for $\nu_t = 0, \ldots, K_t$ and $(\eta_{\mu_t}, v_{\mu_t})$ with probability $\mathcal{M}_{\mu_t}(\Theta_t \times \{v_{\mu_t}\})$ for $\mu_t = 0, \ldots, L_t$.

An illustration is given in Figure 23.4, where the samples represent the joint distribution of η and ξ for $K = L = 1$ (the time index is omitted for simplicity), indicating a negative correlation between the random data. In the one-dimensional case, the simplices covering the support of η and ξ are intervals and, thus, the resulting $\times$-simplex is a rectangle. For instance, in Figure 23.4(a) the edges AB and CD cover the support of η (here,

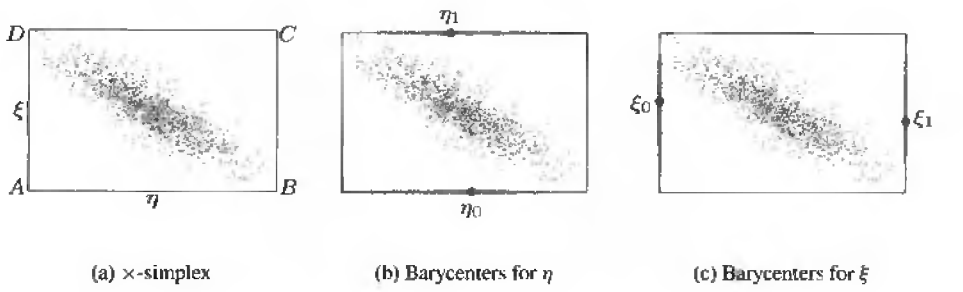


Figure 23.4. *Approximation of a correlated distribution* ($K = L = 1$).

the factor of a term structure model); i.e., A and D correspond to vertex u_0, while B and C are equivalent to u_1. Analogously, AD and BC represent the domain of (the volume risk factor) ξ and correspond to an interval with vertices v_0 and v_1.

Projecting the distribution mass onto AB and CD as in Figure 23.4(b), taking into account the distance from each sample point to the edges, provides the barycenters η_0 and η_1. For each simplex, they are determined as the center of gravity of the projected mass, and their probabilities are equivalent to the proportion of the latter to a total mass of one. Analogously, the barycenters ξ_0 and ξ_1 in Figure 23.4(c) result from a projection of the mass onto AD and BC, respectively.

An advantageous feature from a computational point of view is that the generalized barycenters in (23.12), (23.13) and their probabilities (23.14), (23.15) are completely derived from the first moments of η_t and ξ_t as well as the bilinear cross moments $E_t(\eta_{\nu_t} \cdot \xi_{\mu_t})$, $\nu_t = 0, \ldots, K_t$, $\mu_t = 0, \ldots, L_t$. Since the covariance of two random variables is determined by the first moments and the corresponding cross moments, the measures Q_t^u and Q_t^l incorporate implicitly the correlation between η_t and ξ_t as indicated by the different coordinates of the corresponding barycenters in Figure 23.4(b) and (c). However, cross moments (or covariances, respectively) between different elements of η_t are not taken into account (the same holds for the components of ξ_t). Therefore, no assumptions of independence between the random variables are required.

23.4.2 Lower and upper bounds for value functions

By application of (23.12)–(23.15) at each stage t, the original probability measure P_t is approximated by two discrete measures Q_t^l and Q_t^u with supports

$$\operatorname{supp} Q_t^l = \left\{(u_{\nu_t}, \xi_{\nu_t}) \middle| \nu_t = 0, \ldots, K_t\right\}, \tag{23.16}$$

$$\operatorname{supp} Q_t^u = \left\{(\eta_{\mu_t}, v_{\mu_t}) \middle| \mu_t = 0, \ldots, L_t\right\}. \tag{23.17}$$

The corresponding probabilities are given by $q_t^l(u_{\nu_t}, \xi_{\nu_t}) := \mathcal{M}_{\nu_t}(\{u_{\nu_t}\} \times \Xi_t))$ and $q_t^u(\eta_{\mu_t}, v_{\mu_t}) := \mathcal{M}_{\mu_t}(\Theta_t \times \{v_{\mu_t}\})$. Snbstituting P_t in (23.1′) by these approximate measures provides

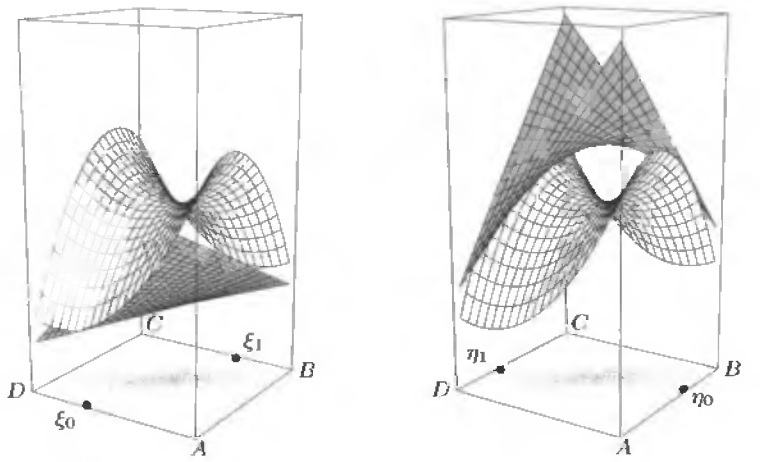


Figure 23.5. *Bilinear approximations of the value function.*

the new value functions

$$\begin{aligned}\psi_t(x^{t-1}, \omega^t) := \min c_t'(\eta^t) \cdot x_t + \int_{\Omega_{t+1}} \psi_{t+1}(x^t, \omega^{t+1}) dQ_{t+1}^l(\omega_{t+1}|\omega^t) \\ \text{s.t. } W_t \cdot x_t + T_{t-1} \cdot x_{t-1} = h_t(\xi^t),\ x_t \geq 0,\end{aligned} \tag{23.18}$$

and

$$\begin{aligned}\Psi_t(x^{t-1}, \omega^t) := \min c_t'(\eta^t) \cdot x_t + \int_{\Omega_{t+1}} \Psi_{t+1}(x^t, \omega^{t+1}) dQ_{t+1}^u(\omega_{t+1}|\omega^t) \\ \text{s.t. } W_t \cdot x_t + T_{t-1} \cdot x_{t-1} = h_t(\xi^t),\ x_t \geq 0,\end{aligned} \tag{23.19}$$

for $t = 1, \ldots, T$ with $\psi_{T+1}(\cdot) = \Psi_{T+1}(\cdot) := 0$. Both ψ_t and Ψ_t are bilinear functions since the integrand $\lambda_{\nu_t}(\eta_t) \cdot \tau_{\mu_t}(\xi_t)$ in (23.12) and (23.13) is bilinear in (η_t, ξ_t). It is shown in [13] that the following relation holds:

$$\psi_t(x^{t-1}, \omega^t) \leq \phi_t(x^{t-1}, \omega^t) \leq \Psi_t(x^{t-1}, \omega^t). \tag{23.20}$$

The meaning of this inequality is that the value function $\phi_t(\cdot)$ with respect to the original measure P_t (which is a saddle function) is supported from below and above by the two bilinear functions $\psi_t(\cdot)$ and $\Psi_t(\cdot)$ with respect to the approximate measures Q_t^l and Q_t^u. The barycenters ξ_{ν_t}, $\nu_t = 0, \ldots, K_t$, are the supporting points for the minorant, while the majorant is supported in η_{μ_t}, $\mu_t = 0, \ldots, L_t$. This situation is illustrated in Figure 23.5. (Again, the case $K = L = 1$ is considered and the time index omitted for simplicity.) Obviously, the two bilinear functions can easily be integrated since the calculation of the expectations reduces to the weighted sums

$$E_t \psi_t(x^{t-1}, \omega^t) = \sum_{\omega_t \in \text{supp } Q_t^l} \psi_t(x^{t-1}, \omega^t) \cdot q_t^l,$$

Figure 23.6. *Split of a* $\times$*-simplex and new positions of the barycenters* ξ_{v_t}.

$$E_t \Psi_t(x^{t-1}, \omega^t) = \sum_{\omega_t \in \text{supp } Q_t^u} \Psi_t(x^{t-1}, \omega^t) \cdot q_t^u,$$

which was the intention of the approximation. Therefore, the problems that result from the substitution of the original conditional measures P_t in (23.1′) and (23.2) by Q_t^l and Q_t^u can be treated as deterministic multistage programs.

Their solutions provide policies $x^l := (x_0^l, \ldots, x_T^l)$ and $x^u := (x_0^u, \ldots, x_T^u)$. While the decisions for $t > 0$ correspond to outcomes in the barycentric scenarios that may be seen as representative rebalancing actions, only the policy for $t = 0$ will be implemented and is of interest for the user. However, a situation may occur where the first-stage decisions are not unique, i.e., $x_0^l \neq x_0^u$. Then, more accurate bounds can be achieved when the support of random data at time t is partitioned into $\ell_t(\omega^{t-1})$ sub-$\times$-simplices with

$$\bigcup_{i=1}^{\ell_t} \Omega_t^i = \Omega_t \supset \text{supp}\, \omega_t, \tag{23.21}$$

$$\Omega_t^i \cap \Omega_t^j = \emptyset, \qquad i \neq j; i, j = 1, \ldots, \ell_t, \tag{23.22}$$

$$\Omega_t^i \text{ are regular } \times\text{-simplices for } i = 1, \ldots, \ell_t. \tag{23.23}$$

A collection $\mathcal{P}^{\ell_t}(\omega^{t-1}) := \{\Omega_t^1, \ldots, \Omega_t^{\ell_t}\}$ which satisfies (23.21)–(23.23) is called a *partition* of the support of ω_t, and an approximation can be obtained by application of the scheme (23.12)–(23.15) to each element Ω_t^i individually. Note that the probabilities assigned to the individual outcomes must be adjusted according to the percentage of the distribution mass that is covered by the corresponding $\times$-simplex. In case the accuracy of a bound is not sufficient, one may split one of the (sub-) $\times$-simplices in the initial partition $\mathcal{P}^{\ell_t}$ (refinement; see Figure 23.6). Then, the solution of the corresponding approximate problem based on the new partition $\mathcal{P}^{\ell_t+1}$ must be at least as good as the former bound. As $\ell_t \to \infty$ and the sub-$\times$-simplices become arbitrarily small with respect to diameter, the approximate value functions ψ_t and Ψ_t *epi-converge* to ϕ_t [13].

Nevertheless, dividing the elements of a partition without strategy may dramatically increase the number of scenarios and, hence, the computational complexity of the corresponding deterministic optimization problems. For example, one may refine the partition with the largest discretization error $\epsilon_t(\omega^t) := \Psi_t(u^{t-1}, \omega^t) - \psi_t(u^{t-1}, \omega^t)$ until the desired accuracy is achieved. If $\epsilon_t(\cdot) = 0$, then the approximation of ϕ_t is exact, and (further) refinements will not improve the solution. In this sense, the existence and utilization of exact bounds may be seen as one of the most important features of barycentric approximation for the solution of multistage stochastic programs. Details of refinement techniques in the context of barycentric approximation can be found in [15].

23.4.3 Cross-simplicial coverages and convergence

Obviously, the distributions induced by the stochastic processes we considered in sections 23.3.1 and 23.3.2 have unbounded support. For the determination of a $\times$-simplicial coverage, $P_t(\cdot|\eta^{t-1}, \xi^{t-1})$ must therefore be substituted by a normalized truncation so that $P_t(\Theta_t \times \Xi_t|\eta^{t-1}, \xi^{t-1}) \geq 1 - \varepsilon$ for some sufficiently small $\varepsilon > 0$.

In the case that the partition consists of a single $\times$-simplex only and all K risk factors are standard normally distributed and uncorrelated, such a coverage may be constructed as follows. A sphere with radius δ around the origin contains a percentage of $2\Phi(\delta) - 1$ of the total mass distribution (Φ denotes the c.d.f.). It can be covered by a regular simplex in $\mathbb{R}^K$ with $K + 1$ vertices. (Note that such a simplex is the polyhedron with the smallest possible number of independent vertices.) In the one-dimensional case, this simplex reduces to an interval $[-\delta, \delta]$ and for $K = 2$ to a triangle whose vertices may be chosen, e.g., as $u_0 = (-\sqrt{3}\delta, \delta)'$, $u_1 = (\sqrt{3}\delta, \delta)'$, and $u_2 = (0, -2\delta)'$. For example, with $\delta = 2$ the circle contains more than 95% of the probability mass, and since the latter is entirely covered by the simplex, any outcome within a range of two standard deviations will be considered in the approximation.

For arbitrary K-dimensional normal distributions with expectation μ and covariance matrix Σ, we make use of the fact that a standard normally distributed random vector $Z \in \mathbb{R}^K$ can be transformed into another one $Y \sim \mathcal{N}(\mu, \Sigma)$ using the lower triangular matrix of the Cholesky decomposition Γ of Σ, i.e., $\Sigma = \Gamma \cdot \Gamma'$. Given a simplex with vertices u_i^Z, $i = 0, \ldots, K$, around the truncated support of an uncorrelated standard normal distribution, the vertices of a simplex that covers the actual *correlated* distribution are obtained from the transformation $\mu + \Gamma \cdot u_i^Z$. This procedure is performed separately for the distributions of η_t and ξ_t at time t to determine the vertices of the (single) $\times$-simplex $\Theta_t \times \Xi_t$ in the initial partition.

According to results in [16], where we applied successive refinements to initial partitions consisting of only one $\times$-simplex, the convergence is rather slow when the (lower) bound based on the vertices of Θ_t and barycenters for ξ_t is refined while the other (upper) bound seems to be stable. The former effect can be explained geometrically by too-extreme coordinates of the vertices we selected to cover the support of η_t. As a consequence, the saddle function exhibits a relatively large degree of concavity with respect to the risk factors η_t and cannot be approximated well by bilinear functions.

Since the split of a simplex generates only one new point but the existing vertices remain in the partition, the influence of extreme outcomes in the initial discretization decreases slowly. Furthermore, the split of a $\times$-simplex in the partition for the first stage provides a better improvement than in case of the partition where the largest discretization error was observed. This is not surprising since the approximation of the distribution in $t = 0$ affects *all* scenarios and, hence, has the highest impact on the solution.

23.4.4 Discretization of interest rate processes

To achieve tighter bounds, possibly with fewer refinement steps, we modified the procedure for the determination of initial coverages in combination with the discrete-time version of the *two-factor mean reversion model* (23.8). While a triangle covers the truncated support of $\eta_t = (s_t, l_t)'$ with only three vertices, which keeps the scenario size moderate, a better ap-

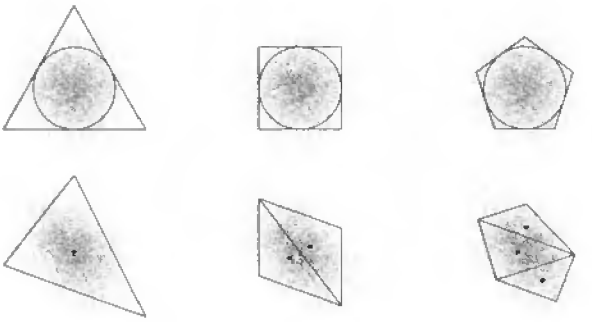


Figure 23.7. *Coverages for the uncorrelated and correlated case with barycenters.*

proximation of the circle around the mass of the (uncorrelated) standard normal distribution can obviously be obtained using polygons with $K_t' > 3$ vertices[1] (see Figure 23.7). More precisely, we determine pentagons as approximations of the (unit) circle that are partitioned into three triangles for the first stages of the stochastic program. Combining them with intervals that cover the support of ξ_t results in initial partitions of $\ell_t = 3$ cross simplices. After the transformation according to the expectation and covariance matrix of the actual distributions at time t, generalized barycenters and weights are determined for each of them individually according to (23.12)–(23.15).

Because this discretization yields nine outcomes for the lower bound (corresponding to the number of vertices in the partition), we use squares partitioned into two simplices ($\ell_t = 2$) and then single triangles ($\ell_t = 1$) at later stages where the approximation has less impact on the solution of the complete stochastic program in order to reduce the growth in problem size. The specific size ℓ_t of the initial partitions for all $t = 1, \ldots, T$ depends on the number of periods that must be achieved. The total number of scenarios at the tth stage is given by

$$s_t^l := \prod_{\tau=1}^{t}(K_\tau + 1) \cdot \ell_\tau = 3^t \cdot \prod_{\tau=1}^{t} \ell_\tau,$$

$$s_t^u := \prod_{\tau=1}^{t}(L_\tau + 1) \cdot \ell_\tau = 2^t \cdot \prod_{\tau=1}^{t} \ell_\tau$$

for the lower and upper bound, respectively. In our experience, we obtain a better accuracy for a similar problem size compared to sequential refinements of partitions where we start with a single $\times$-simplex for the discretization. For example, in the case of an eight-stage quarterly planning problem ($\mathcal{D}^S = \{3M, 6M, 1Y, 2Y, 3Y, 4Y, 5Y\}$) with $\ell_1 = 3, \ell_2 = 2, \ell_3 = \cdots = \ell_7 = 1$ and $\delta = 2$, we have 13,122 scenarios in the lower (larger) approximation. The objective function values are 5818.6 for the lower bound and

[1] General polyhedra to obtain tighter bounds are also exploited in Gassmann and Ziemba [18]. However, they develop an upper bound for the case of *single* polyhedrons with arbitrary number of vertices, while we divide the initial polyhedron that covers the support of η_t in regular subsimplices to apply the barycentric approximation scheme.

5888.4 for the upper bound,[2] equivalent to a relative difference of only 1.20%.

In the case of the *principal component model* for short-term planning, we make use of the fact that the K_t components of η_t are orthogonal (uncorrelated) by construction. This allows us to represent the Θ_t, $t = 1, \ldots, T$, themselves as $\times$-simplices, i.e., $\Theta_t = \Theta_t^1 \times \cdots \times \Theta_t^{K_t}$. Each $\Theta_t^i = [-\delta, \delta] \subset \mathbb{R}$ is an interval that covers the truncated support of the ith principal component η_t^i, $i = 1, \ldots, K_t$. The required modified formulas for the barycentric approximation scheme can be found in [12] for the two-stage case. For example, to obtain the generalized barycenters $\eta_{\mu_t} = (\eta_{\mu_t}^1, \ldots, \eta_{\mu_t}^{K_t})'$ and corresponding probabilities, we have to replace $\lambda_{\nu_t}(\eta_t)$ in (23.12)–(23.15) by $\prod_{i=1}^{K_t} \lambda_{\nu_t^i}(\eta_t^i)$ with $\nu_t = (\nu_t^1, \ldots, \nu_t^{K_t})$, $\nu_t^i = 0, \ldots, K_t$. Note that each principal component η_t^i may still be correlated with the volume risk factor ξ_t.

The $\times$-simplex Θ_t as the Cartesian product of K_t intervals is a multidimensional rectangle with 2^{K_t} vertices, and the number of scenarios at stage t is given by $s_t^l := \prod_{\tau=1}^{t} 2^{K_\tau}$ for the lower bound. The growth in problem size for the upper approximation with $s_t^u := \prod_{\tau=1}^{t} 2^{L_\tau}$ is less dramatic since we assumed $L_t = L = 1$. Thus, with $K_t = K = 3$ relevant factors, the size of the scenario set in the former case is multiplied by 8 with each additional stage and no refinements.

Since this increase may be too restrictive for the number of periods that can be taken into account, the growth in problem size may be reduced by considering all principal components only in the first stages, while the third and then the second factor are ignored at later points in time. This is motivated by the empirical observation that they contribute only 3% and 19%, respectively, to the dynamics of interest rates [31].

As a consequence, the scenarios incorporate complex tilt and humped movements at the beginning but may reflect only shifts of the yield curve toward the end of the planning horizon where the outcomes have less impact on the first-stage decision. In this way, we achieve a lower and upper bound of 12,830.9 and 12,873.3, i.e., a relative difference of 0.33% for a seven-stage monthly planning problem[3] with maturities $\mathcal{D}^S$ = {1M, 2M, 3M, 6M, 1Y, 2Y, 3Y, 4Y, 5Y} and dimension size $K_1 = K_2 = 3$, $K_3 = \cdots = K_6 = 2$, which results in 16,384 scenarios.

23.5 Application to the funding problem

Scenarios generated by means of any discretization method may be represented by a tree; see [14] for a formal description. Nodes of this *scenario tree* with depth t correspond to outcomes of random data at time t. Using the barycentric approximation scheme described above results in a total number of nodes $\sum_{t=0}^{T} s_t^l$ for the lower bounding problem, which is the larger one since we have at least as many interest rate as volume risk factors ($K_t \geq L_t \; \forall t$).

[2]Process parameters: $\kappa_s = 0.8687$, $\theta_s = -0.0061$, $\sigma_s = 0.0223$, $\kappa_l = 0.1940$, $\theta_l = 0.0507$, $\sigma_l = 0.1532$, $\rho = -0.3610$, $\phi_1 = -0.002689$, $\phi_2 = 0.4293$ for the term structure model ($\Delta t = 0.25$) and $a = 83.82$, $b = 0$, $\rho_\upsilon = 0.9835$, $\sigma_\xi = 166.0$ for the volume process.

[3]We estimated the factor sensitivities for CHF Euromarket rates as shown in Figure 23.3 with principal component analysis, and the correlations between the factor scores and the volume risk factor ξ_t are 0.34, −0.18, and −0.03. The different magnitudes of the objective values obtained with the mean reversion model above results from the fact that only a fraction of the mortgage volume could be included in the optimization there due to quarterly planning, i.e., the actual portfolio had to be split into four components since the period length was extended to three months there.

The funding model (23.3) has $D + \sum_{d \in \mathcal{D}^S} I^d$ variables at each stage.

For instance, with traded maturities $\mathcal{D}^S = \{3M, 6M, 1Y, 2Y, 3Y, 4Y, 5Y\}$ and 8 tranches for each of them, we have 76 variables which must be duplicated for all nodes. Thus, the corresponding deterministic program for the example above with 13,122 scenarios already consists of more than 1.5 million variables in 19,666 nodes. The problem generation and solution with standard optimization tools such as CPLEX requires up to a few hours on a medium-size workstation (Sun Ultra 10, 1 GB RAM). This may be seen as tolerable since the determination of refinancing policies is performed only once a month.

As a consequence of the exponential growth in problem size, we can deal with a limited number of stages only, although the planning horizon of the refinancing problem is actually infinite. Our experience from various case studies is that an increase in the number of stages (even at the cost of a reduced accuracy of the approximation) in general leads to a higher performance. This is because the model has a greater flexibility for corrections of the initial portfolio. Moreover, an extension of the horizon allows a better consideration of the impact of future changes in the risk factors (e.g., a sharp drop in volume that might lead to a surplus of liabilities over the mortgage position) for the first-stage decision.

However, in practice we are not able to solve problems with more than ten stages with the model formulation and solution techniques described before. If we wanted to include the whole spectrum of traded instruments in the interbank market from 1 month to 5 years and since the period length is given by the shortest maturity, this would correspond to a planning horizon of less than 1 year, which we consider as insufficient for the long-term model.

Therefore, the determination of refinancing decisions is carried out in two steps. First, we perform an optimization run with maturities between 1 and 5 years where scenarios are based on the mean reversion model. When the solution indicates that a short-term policy should be implemented, we use the principal component model to analyze whether the funding costs can be reduced by shorter maturities than 1 year. Alternatively, the liability portfolio may be split into a short- and a long-term component, which are optimized separately. The corresponding volumes depend, e.g., on the current portfolio composition, limits given by the bank's internal risk management system, or forecasts for interest rates and mortgage demand.

Before the stochastic optimization model was applied to real positions, a case study had been conducted to assess its performance for a period of high yields. The study was based on monthly money market rates and the volume of a real mortgage position provided by a Swiss bank. (See Figure 23.8; the lag between interest rates and volume results from the fact that banks adjust the relevant client rate with some delay to a new market situation.) The level of interest rates at which mortgages had to be refinanced on the market reached up to 10% while the client rate that the bank receives never climbed above 7% due to the political cap in Switzerland. Hence, refinancing the mortgage position with a replicating portfolio of 25% 6-month, 50% 1-year, and 25% 3-year instruments which serves as benchmark provides a (negative) average margin of -0.21% for the sample period. The static policy required also investing significant amounts at low yields because the share of portfolio positions with longer maturities was still too large when the mortgage demand dropped, resulting in a surplus of liabilities.

For simplicity, we did not follow the two-step procedure outlined before with separate runs of the long-term and short-term planning models. Instead, we used the former only with

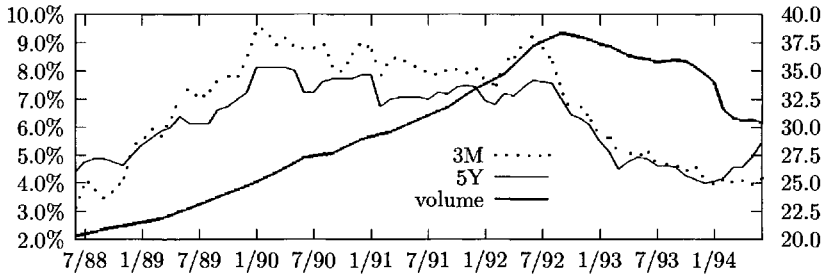


Figure 23.8. *Interest rates (left) and mortgage volume (right).*

maturities from 3 months to 5 years. With nine stages, this quarterly planning is equivalent to a time horizon of only 2 years. According to our experience, the performance of the model improves significantly for longer periods. To achieve this number of stages, we did not split the initial partitions.

Liquidity limits for transactions without penalties ranged from 400 million for the shortest to 100 million for the longest maturity. Transaction costs increased by one basis point (BP) with each additional tranche of 50 mio. except for 3 months, where tranches are twice as large. For the initial portfolio composition, it was assumed that the static 6M/1Y/3Y-mix had been implemented in the past; i.e., all positions had to be renewed within the first 3 years. Parameters for the stochastic processes were updated semiannually based on observations of the previous 5 years.

After a refinancing decision had been found, the portfolio positions were updated and a new optimization was started with the next set of interest rates and mortgage volume out of the sample period ("rollover planning"). The result after all 73 runs is summarized in Table 23.1. Compared to the replicating portfolio benchmark, the average refinancing costs for the renewed positions could be reduced from 6.50% to 6.13%. The margin as the difference between the client rate and the refinancing costs increased correspondingly by 37 BP to 0.16%.

Table 23.1. *Comparison of dynamic policies with the replicating portfolio.*

Method	Ref. costs [%]		Margin [%]	
	avg.	std. dev.	avg.	std. dev.
SP model	6.13	1.38	0.16	0.82
repl. portf.	6.50	1.80	−0.21	1.26

While this number implies only a small profit at first sight, one must take into account that the static approach was not able to provide a positive margin at all for the specific market situation in the case study, which results in significant losses for the bank. Moreover, the standard deviation was also reduced noticeably compared to the replicating portfolio although the latter was constructed to minimize the volatility of the margin.

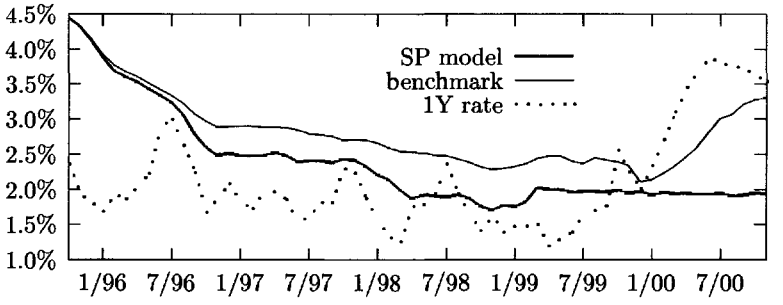


Figure 23.9. *Refinancing costs for dynamic and static portfolios since* 1995.

23.6 Conclusions

An extended version of the multistage stochastic programming model described in this paper has been in use by a major Swiss bank since 1995 for refinancing variable mortgages. The modifications are, e.g., additional constraints for the portfolio duration to achieve a target that is frequently defined by the bank's board of directors. While the two-factor mean reversion model provides a good description of both ends of the term structure due to the selection of state variables, the error of fit is slightly worse for medium-term maturities. Therefore, a third factor has been introduced that controls the curvature of the yield curve. Unfortunately, we cannot reveal all details of this commercial model in publications.

As evidence for the performance of the model since it has been applied in practice, the average funding rate compared to the static mix that was exploited by the bank before as benchmark is shown in Figure 23.9. With an average margin of approximately 70 BP that could be achieved with the replicating portfolio approach in the long run, an increase in the order of 37 BP represents a significant improvement of the bank's profits. Before the application of another version of the model to savings accounts started in 1997, another case study with data of a real position was conducted [16]. The results were compared to the performance of two replicating portfolios that had been used by the bank for the management of different deposit positions.

Figure 23.10 shows the evolution of the margin between the return of the invested funds and the client rate for the dynamic policies determined by the stochastic optimization model and the static benchmark policies. According to Table 23.2, the average margin over the sample period could be improved by 25 BP compared to the better replicating portfolio while volatility is significantly reduced. The increase in performance at lower risk can be explained by the possibility of rebalancing transactions in the dynamic approach. Similar observations have also been reported for other multistage stochastic programming models [5].

There are many possible ways to improve the model in the form presented here, e.g., with respect to the volume process specification (23.9). However, an empirical investigation of alternative models failed because of the lack of publicly available data. Moreover, new mortgage products recently were introduced in Switzerland that led to a noticeable change in the demand for variable mortgages, and it is difficult to correct the structural break induced by such a nonstochastic event in the data. We are also still investigating other term structure

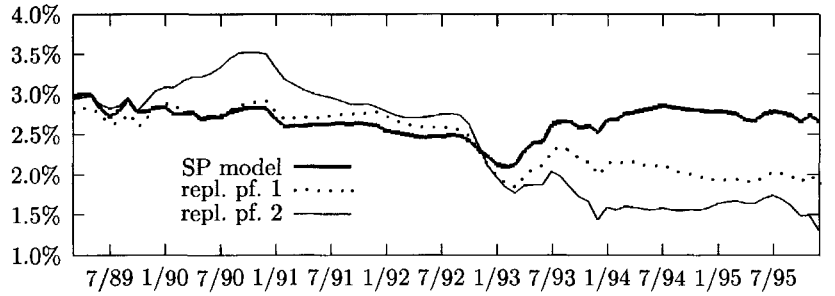


Figure 23.10. *Evolution of margin for dynamic versus static investment strategies.*

Table 23.2. *Performance of stochastic optimization model for savings accounts.*

Method	Weights repl. portf.			Margin [%]	
	1 yr.	2 yrs.	5 yrs.	mean	std. dev.
SP model	–	–	–	2.66	0.19
repl. portf. 1	0.0	0.5	0.5	2.41	0.36
repl. portf. 2	0.35	0.35	0.3	2.40	0.70

models with respect to their practical use for scenario generation.

Some authors derive interest rate scenarios from (binomial) lattice models that have been developed for pricing derivative securities. In a continuous-time framework, this is equivalent to a one-factor specification with time-dependent parameters. Since these models are calibrated only to the term structure observed at the present time, we are not confident as to whether such an approach can provide a suitable description of long-term interest rate behavior but should rather be seen as an interpolation technique for the initial yield curve.

As Hull and White [22] point out, "It is important to distinguish between the goal of developing a model that adequately describes term-structure movements and the goal of developing a model that adequately values most of the interest-rate-contingent claims that are encountered in practice. It is quite possible that a two- or three-state variable model is necessary to achieve the first goal."

According to our experience, models for a planning horizon of several years should be calibrated to a historical data set that covers at least an economic cycle. In can also be helpful to assess a term structure model by comparing scenarios generated in a simulation study with the characteristics of empirically observed data, as suggested by Frauendorfer and Schürle [17].

Interest rate scenarios based on the principal component and the two-factor mean reversion models in section 23.3.1 may not be free of arbitrage. This aspect should be taken into account in applications with simultaneous investing and refinancing. Otherwise, the model could try to exploit such spurious arbitrage opportunities in its decisions although they result only from a misspecification [26].

Using barycentric approximation, we are able to obtain tight bounds for the original

problem when the support of random data is partitioned into several $\times$-simplices. Despite the low discretization error, different first-stage decisions for the lower and upper approximate problems may still occur. This requires an improvement of the scenario selection, and the investigation of efficient refinement techniques based on suitable error measures is still ongoing. However, the example of the funding problem presented here and many other approaches described in this book show that multistage stochastic programming models have already passed the state of research and may be applied successfully to financial decision making under uncertainty.

Acknowledgment

We are grateful to W. T. Ziemba for his helpful comments on an earlier version of this paper that helped improve the exposition significantly.

Bibliography

[1] A. Beltratti, A. Consiglio, and S. A. Zenios, *Scenario modeling for the management of international bond portfolios*, Ann. Oper. Res., 85 (1999), pp. 227–247.

[2] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer, New York, 1997.

[3] D. R. Cariño, T. Kent, D. H. Myers, C. Stacy, M. Sylvanus, A. L. Turner, K. Watanabe, and W. T. Ziemba, *The Russell-Yasuda Kasai model: An asset/liability model for a Japanese insurance company using multistage stochastic programming*, Interfaces, 24 (1994), pp. 29–49.

[4] D. R. Cariño, D. H. Myers, and W. T. Ziemba, *Concepts, technical issues, and uses of the Russsell-Yasuda Kasai financial planning model*, Oper. Res., 46 (1998), pp. 450–462.

[5] D. R. Cariño and W. T. Ziemba, *Formulation of the Russell-Yasuda Kasai financial planning model*, Oper. Res., 46 (1998), pp. 433–449.

[6] N. C. P. Edirisinghe, *New second-order bounds on the expectation of saddle functions with applications to stochastic linear programming*, Oper. Res., 44 (1996), pp. 909–922.

[7] N. C. P. Edirisinghe, *Bound-based approximations in multistage stochastic programming: Nonanticipativity aggregation*, Ann. Oper. Res., 85 (1999), pp. 103–127.

[8] N. C. P. Edirisinghe and W. T. Ziemba, *Tight bounds for stochastic convex programs*, Oper. Res., 40 (1992), pp. 660–677.

[9] N. Edirisinghe and W. T. Ziemba, *Bounding the expectation of a saddle function with application to stochastic programming*, Math. Oper. Res., 19 (1994), pp. 314–340.

[10] N. EDIRISINGHE AND W. T. ZIEMBA, *Implementing bounds-based approximations in convex-concave two-stage stochastic programming*, Math. Program., 75 (1996), pp. 295–325.

[11] K. FRAUENDORFER, *Stochastic Two-Stage Programming*, Springer, New York, 1992.

[12] K. FRAUENDORFER, *The approximation of separable stochastic programs*, J. Comput. Appl. Math., 56 (1994), pp. 23–44.

[13] K. FRAUENDORFER, *Multistage stochastic programming: Error analysis for the convex case*, Math. Methods Oper. Res., 39 (1994), pp. 93–122.

[14] K. FRAUENDORFER, *Barycentric scenario trees in convex multistage stochastic programming*, Math. Program. Ser. B, 75 (1996), pp. 277–293.

[15] K. FRAUENDORFER AND C. MAROHN, *Refinement issues in stochastic multistage linear programming*, in Stochastic Programming Methods and Technical Applications (Proceedings of the 3rd GAMM/IFIP Workshop 1996), K. Marti and P. Kall, eds., Springer, New York, 1998, pp. 305–328.

[16] K. FRAUENDORFER AND M. SCHÜRLE, *Stochastic optimization in asset and liability management: A model for non-maturing accounts*, in Probabilistic Constrained Optimization: Methodology and Applications, S. Uryasev, ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 67–101.

[17] K. FRAUENDORFER AND M. SCHÜRLE, *Term structure models in multistage stochastic programming: Estimation and approximation*, Ann. Oper. Res., 100 (2000), pp. 189–209.

[18] H. GASSMANN AND W. T. ZIEMBA, *A tight upper bound for the expectation of a convex function of a multivariate random variable*, Math. Program. Study, 27 (1986), pp. 39–53.

[19] A. GEYER, W. HEROLD, K. KONTRINER, AND W. T. ZIEMBA, *The Innovest Austrian Pension Fund Financial Planning Model InnoALM*, Working Paper, University of British Columbia, Vancouver, 2001.

[20] J. GONDZIO AND R. KOUWENBERG, *High Performance Computing for Asset Liability Management*, Working Paper, University of Edinburgh, Edinburgh, 1999.

[21] C. HUANG, W. T. ZIEMBA, AND A. BEN-TAL, *Bounds on the expectation of a convex function of a random variable: With applications to stochastic programming*, Oper. Res., 25 (1977), pp. 315–325.

[22] J. HULL AND A. WHITE, *Pricing interest rate derivative securities*, Rev. Financ. Stud., 3 (1990), pp. 573–592.

[23] J. JAMES AND N. WEBBER, *Interest rate modelling*, in Financial Engineering, John Wiley, New York, 2000.

[24] J. JENSEN, *Sur les fonctions convexes et les inégalités entre les valeurs moyennes*, Acta Math., 30 (1906), pp. 175–193.

[25] J. KALLBERG, R. WHITE, AND W. T. ZIEMBA, *Short term financial planning under uncertainty*, Management Sci., 28 (1982), pp. 670–682.

[26] P. KLAASSEN, *Discretized reality and spurious profits in stochastic programming models for asset/liability management*, Eur. J. Oper. Res., 101 (1997), pp. 374–392.

[27] M. KUSY AND W. T. ZIEMBA, *A bank asset and liability management model*, Oper. Res., 34 (1986), pp. 356–376.

[28] A. MADANSKY, *Bounds on the expectation of a convex function of a multivariate random variable*, Ann. Math. Statist., 30 (1959), pp. 743–746.

[29] J. M. MULVEY, G. GOULD, AND C. MORGAN, *The asset and liability management system for Towers Perrin-Tillinghast*, Interfaces, 30 (2000), pp. 96–114.

[30] S. M. SCHAEFER AND E. S. SCHWARTZ, *A two-factor model of the term structure: Approximate analytical solution*, J. Financial Quantitative Anal., 19 (1984), pp. 413–424.

[31] M. SCHÜRLE, *Zinsmodelle in der stochastischen Optimierung*, Haupt, 1998.

[32] S. A. ZENIOS, *Asset/liability management under uncertainty for fixed-income securities*, Ann. Oper. Res., 59 (1995), pp. 77–97.

[33] W. T. ZIEMBA AND J. MULVEY, EDS., *Worldwide Asset and Liability Modeling*, Cambridge University Press, Cambridge, UK, 1998.

This page intentionally left blank

Chapter 24

Optimization Models for Structuring Index Funds

*Stavros A. Zenios**

24.1 Basics of market indices

An index is a single statistic that summarizes the relative changes of a set of variables, such as stock or bond prices. An index can be broad in scope, including variables based on value, growth rate, or geographical region, or it can take a narrow view of the market, focusing on a single economic sector or an industry.

As international institutions change location and invest funds outside their domestic market they need tools to guide investment analysis, asset allocation, and performance measurement in diverse markets. Investors quite often evaluate their portfolio choices with respect to overall market trends. Indices provide comprehensive measures of market trends and are useful benchmarks for portfolio performance. But they can also be used to guide portfolio selection using passive strategies whereby portfolios are structured to track a market index. There is ample empirical evidence that actively managed portfolios do not outperform the market, and those that do outperform do so inconsistently. This evidence explains the popularity of passive portfolio management strategies. In a study of the performance of 769 all-equity actively managed funds during the period 1983–1989 it was determined that the average fund return was from 200 to 500 basis points below the S&P 500 index. From among those funds that did well in one period, only one-fourth continued to do equally well in the next period. One-fourth of the best performers would find themselves among the worst performers in the following time period.

Market indices are valuable tools in financial decision making as they monitor performance of broad segments of the market. For instance, the NYSE composite index measures all common stocks listed on the New York Stock Exchange. It has four subgroup indices

*HERMES Center on Computational Finance and Economics, University of Cyprus, P. O. Box 20537, 1678 Nicosia, Cyprus, and Financial Institutions Center, The Wharton School, University of Pennsylvania, Philadelphia, PA 19104 (zenioss@ucy.ac.cy).

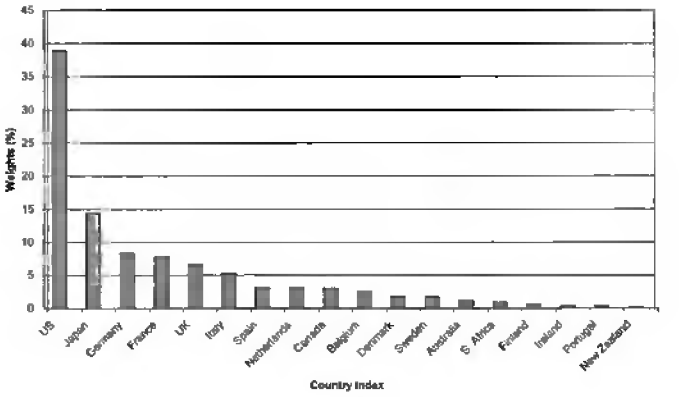


Figure 24.1. *Composition of the J. P. Morgan GBI broad in September,* 1997.

by sector: industrial, transportation, utility, and finance. The index tracks aggregate market trends of NYSE common stocks. However, with more than 3,000 stocks listed in the New York Stock Exchange this index cannot form the basis for a tradeable portfolio. In contrast, the S&P 500 measures the performance of 500 stocks selected by the index committee of Standard and Poor's. The size and the criteria for candidate stocks make this index suitable for trading. More than 1 trillion dollars in index funds are managed to replicate S&P indices.

Indices are not restricted to the stock markets. Bond indices are calculated for government bonds, corporate bonds, asset backed securities, and so on. For instance, J. P. Morgan publishes the government bond index (GBI) broad, which is a widely used benchmark for measuring performance and quantifying risk across international fixed income bond markets. J. P. Morgan indices measure the total, principal, and interest returns in 18 government bond markets and are reported in 85 different currencies. The countries represented in this index are shown in Figure 24.1. Merrill Lynch publishes a Euro dollar index of corporate bonds. The index consists of highly liquid securities so that liquidity risk is eliminated. Currency risk is also eliminated since the securities in the index are dollar denominated.

Indices may be *rules driven* or *discretionary*. In the former case detailed rules govern the choice of securities to form the index. These indices are transparent and predictable in their composition. We emphasize that predictability of composition does not imply predictability of returns. An analyst may expect, based on the publicized rules, that bonds of a certain maturity and coupon rate will be included in the index. But it is not known what the return of these securities will be.

The constitution of discretionary indices is determined by a committee. The views of this committee are known, but these are merely views and not detailed rules. For instance, the Standard and Poor's index committee makes its decisions based on trading analysis, liquidity, ownership, fundamental analysis, market capitalization, and sector representation. As an example of committee views we mention the committee's announcement of a "pronounced bias" against the addition of tracking stocks in their indices after 1999.

No matter how an index is created, to be a meaningful tool for decision support it must have certain characteristics. It is desirable that only traded issues available to investors constitute an index. This provides a realistic measure of market performance and eliminates

or reduces liquidity risk. Tradeable portfolios may be created from the securities in the index, except that the index set may be too large. Furthermore, the market or market segments included in the index should not have significant barriers to entry, and expenses for investing in these markets should be predictable and not excessive. The composition of the index should be relatively stable without unnecessary changes of the constituents, and investors should be able to replicate the returns reported by the index using market data. Finally, it is important to choose an index that closely approximates the universe of securities a manager actually invests in—such as large cap, small cap, growth, value, corporate bonds of a given industrial sector or a given rating, or the broad market.

For discretionary indices that are suitable for trading, any addition or deletion from the index has an effect on the market. Analysis of stock prices for the 188 companies that were added in the S&P 500 index during the period 1991–2000 showed that significant abnormal returns were realized when the index committee announced that these stocks would be added to the index. Abnormal returns were also realized for the securities added in the S&P MidCap 400 and the S&P SmallCap 600. The effects for stocks added to these two indices were less significant than the effects for stocks added to the S&P 500 stocks, as these indices are not followed as closely as the S&P 500 by index fund managers.

Once the constituents of the index are determined, their weighting must be specified. Securities in the index may be equally weighted, price weighted, or cap weighted. Under equal weighting, each security in the index is given the same weight, which is 1 divided by the total number of constituent securities. Price weighted indices assume that one share is held in each constituent. As a result the index is more heavily weighted toward the expensive bonds. The Dow Jones Industrial Average is price weighted, because when the index was conceived in 1890 the easiest calculation for assigning weights was to add the prices of the 12 securities in the index and divide by 12. Cap weighted indices assign weights to the securities in proportion to their market capitalization, which is the number of outstanding shares times the price. Figure 24.1 shows the weights in the J. P. Morgan international index as the percentage of each country's capitalization in the global government bond market.

The return of an index is a weighted combination of the returns of the constituent securities. Assume that there are K securities in the index with random return vector $\tilde{r} = (\tilde{r}_j)_{j=1}^K$ and normalized weights vector $w = (w_j)_{j=1}^K$ such that $\sum_{j=1}^K w_j = 1$. Note that we assume that the index is composed not of real securities but of generic securities that are representative of the risk factors of the index. Hence we use $j \in \mathcal{K} = \{1, 2, \ldots, K\}$ to denote the securities in the index. Equity indices are usually represented by real securities, chosen to be representative of some risk factor in the market that is being indexed. For instance, IBM stock in the S&P 500 may be representative of the risk factors of the computer industry. For fixed income indices it is usually the case that *generic* securities are constructed that have some characteristics that are typical of a broad market segment. The mortgage-backed securities indices consist of generic securities with a given weighted average coupon, given weighted average maturity, and issued by given issuers (e.g., Federal National Mortgage Association, Federal Housing Authority). There may be several traded securities with characteristics similar to the generic securities. The Salomon Brothers index of mortgage-backed securities consists of a couple of hundred generic securities, which are representative of the hundreds of thousands of mortgage securities available in the U.S. market.

The return of the index is

$$R_I(w;\tilde{r}) = \sum_{j=1}^{K} w_j \tilde{r}_j \tag{24.1}$$

and in a discrete scenario setting by

$$R_I(w;r^l) = \sum_{j=1}^{K} w_j r_j^l, \tag{24.2}$$

where l is an index from the scenario index set Ω. It is clear from these expressions that the choice of a weighting scheme will make a difference in the index returns. From the viewpoint of the manager of an index fund, however, the weights are given a priori.

How do we structure a portfolio whose growth rate will closely mimic an index? Such a portfolio is called an index fund, and optimization models for structuring index funds are developed next. In some cases funds are managed with the objective of outperforming an index while preserving the key risk characteristics of the index. These funds are called *enhanced index* or *index-plus* funds.

The discussion on indexation models focuses on fixed income index funds. Indexation of bonds is a much more complex problem than indexation of stocks. There are thousands of bonds with different maturities, coupon rates, issuer, or issue, and many of these are completely illiquid. Some indices, such as the indices for the mortgage-backed securities market, may consist of representative or generic bonds. Some of these bonds in actuality may not be issued by any agency, or they may not be actively traded, although bonds with similar characteristics are available. For portfolios of equities it is usually the case that the manager of a large fund can invest in all the stocks in the index in proportion to the weights they carry in the index. With this asset allocation the portfolio will perform exactly like the index.

Managers of index funds face significant challenges in matching an index. First is the very large number of instruments they must deal with. Second, the purchase or sale of securities by an index fund incurs transaction costs, while the calculation of index returns ignores these costs. Third, indices usually assume that coupon payments during a month are immediately reinvested in the index. (Such indices are called *fully invested.*) An index fund manager, on the other hand, has to identify appropriate asset purchases with the coupon payments and pay transaction costs for these investments. Finally, managers of large portfolios face liquidity risk in purchasing securities from the market represented by the index. Even if the index may not contain illiquid securities in its composition, the price of actively traded securities may carry a liquidity premium when the manager of a large fund needs to add or drop them from the portfolio. To address the complexities arising in structuring indexed portfolios we turn to mathematical models, and stochastic programming plays an important role.

24.2 Indexation models

There are two distinct modeling approaches for creating index funds: *structural* and *co-movements-based*. In a structural approach the index fund is created to have a risk factor structure similar to that of the index. An approach based on security comovements creates

a portfolio so that its response to the various risk factors is similar to that of the index. The former approach is more mechanistic; it is also called the *cell* approach or the *linear programming* approach. An approach based on comovements views the target index as a random liability. An integrative model is used to select securities with returns that mimic the target liability returns under several scenarios. Scenario optimization models can then be applied.

24.2.1 A structural model for index funds

We assume that K securities from the universe of investable securities or the set of risk factors $\mathcal{K}$ constitute the index, with normalized weights w_j, $j = 1, 2, \ldots, K$. We need to determine the holdings x_i of securities in the indexed portfolio from the universe $\mathcal{U}$. These holdings are in percentage of total assets. In a structural approach the universe of available securities is classified into cells according to the characteristics that affect bond returns. Cells may be created for different maturity ranges, sectors, coupon ranges, credit ratings, and features such as call, sinking fund, conversion, or other provisions. The weights of bonds in the index that belong to each cell are calculated from the index data as $\sum_{c \in \text{cell } k} w_c$. Since a security in the index may belong to multiple cells—for instance, to the cell of securities with medium maturity, and to the cell of securities of the telecommunications sector—these cell weights must be normalized to add up to 1. For simplicity we assume that there are as many cells as the securities in the index so the weight on the kth cell is w_k. An indicator function δ is

$$\delta_{ij} = \begin{cases} 1 \text{ if bond } i \text{ belongs to the } j\text{th cell,} \\ 0 \text{ otherwise.} \end{cases} \tag{24.3}$$

The following linear program creates a portfolio with a structure similar to the index.

Model 24.2.1 Linear program for indexed funds

$$\text{Maximize} \quad F(x) \tag{24.4}$$

$$\text{subject to} \quad \sum_{i=1}^{n} \delta_{ij} x_i = w_j \qquad \text{for all } j = 1, 2, \ldots, K, \tag{24.5}$$

$$\sum_{i=1}^{n} x_i = 1, \tag{24.6}$$

$$x \in X \subseteq \mathbb{R}_n^+. \tag{24.7}$$

The set X denotes the set of feasible solutions, which may be restricted by additional constraints such as diversification constraints, limits on portfolio turnover, the requirement that the duration of the index fund should be equal to the duration of the index, and so on. In the absence of constraints of this form, i.e., when $X = \mathbb{R}_n^+$, the optimal solution of the model will have $K + 1$ nonzero holdings at optimality, x_i^*, corresponding to the $K + 1$ equality constraints in the model. When K is large, the resulting index fund consists of a large number of small holdings, thus increasing management costs. A typical practical

approach for eliminating this problem is to impose equality constraints for cells with weight w_j greater than some user-specified threshold. Upper and lower bounds on the holdings in any security may also be imposed to limit very small positions that imply higher management costs and very large positions with significant exposure to security-specific risks and perhaps liquidity risk.

24.2.2 A model for index funds based on comovements

A model based on comovements views both the index return and the returns of securities in the set U as uncertain, conditioned on the scenario set Ω. The tracking error of the portfolio against the index is

$$R_\epsilon(x; w, \tilde{r}) = R_p(x; \tilde{r}) - R_I(w, \tilde{r}). \tag{24.8}$$

In the discrete scenario setting, and using the linear expression for portfolio and index returns, we have

$$R_\epsilon(x; w, r^l) = \sum_{i=1}^{n} r_i^l x_i - \sum_{j=1}^{K} w_j r_j^l. \tag{24.9}$$

We introduce variables y_+^l and y_-^l to measure, respectively, the positive and negative deviations of the portfolio return from the index return. The tracking error is

$$R_\epsilon(x; w, r^l) = y_+^l - y_-^l, \tag{24.10}$$

where

$$y_+^l = \max\left[0, \sum_{i=1}^{n} r_i^l x_i - R_I(w, r^l)\right], \tag{24.11}$$

$$y_-^l = \max\left[0, R_I(w, r^l) - \sum_{i=1}^{n} r_i^l x_i\right]. \tag{24.12}$$

y_+^l is nonzero in those scenarios when the portfolio outperforms the index, and y_-^l is nonzero when the portfolio underperforms the index. With these definitions of y_+^l and y_-^l we formulate the tracking model 24.2.2.

Model 24.2.2 Tracking model

$$\text{Maximize} \quad \sum_{i=1}^{n} \bar{r}_i x_i \tag{24.13}$$

$$\text{subject to} \quad \sum_{i=1}^{n} r_i^l x_i - R_I(w, r^l) \geq -\epsilon \qquad \text{for all } l \in \Omega, \tag{24.14}$$

$$\sum_{i=1}^{n} x_i = 1, \tag{24.15}$$

$$x \in X \subseteq \mathbb{R}_n^+. \tag{24.16}$$

Similarly, we can develop models to minimize the expected downside tracking error subject to a target return or to trade off upside potential against downside risk. None of these models is, strictly speaking, a tracking model, as they favor upside deviations. A tracking model that limits both upside and downside deviations is given as Model 24.2.3.

Model 24.2.3 Two-sided tracking model

$$\text{Maximize} \quad \sum_{i=1}^{n} \bar{r}_i x_i \tag{24.17}$$

$$\text{subject to} \quad -\epsilon \leq \sum_{i=1}^{n} r_i^l x_i - R_I(w, r^l) \leq \epsilon \qquad \text{for all } l \in \Omega, \tag{24.18}$$

$$\sum_{i=1}^{n} x_i = 1, \tag{24.19}$$

$$x \in \mathbb{R}_n^+. \tag{24.20}$$

24.3 Models for international index funds

The models in the previous section deal with the problem of choosing bonds from a given universe to create an index fund. Quite often indices are *composite* indices of other indices. There is then a need to determine the broad breakdown of the fund among the constituent indices before picking specific bonds. Such a case arises in the passive management of global portfolios using an indexation strategy. For example, the J. P. Morgan GBI broad tracks the trends in the government bond markets in 18 countries. Each country has its own index, and the GBI broad is a composite index of the trends of these 18 indices. The problem of choosing among broad indices over relatively long horizons is the *strategic asset allocation* problem. The *tactical asset allocation* problem deals with the more immediate task of investing so that a specific index is closely tracked for short time horizons.

When the index of indices refers to a single market, the portfolio selection model may be built directly on the universe of the bonds in all subindices. When dealing with an international index fund the problem is more complex. The portfolio model must consider exchange rate movements in addition to the trends of the subindices. In this section we develop models for tracking global indices. These models can also be used to track composite indices in a local market, taking into account the distinctive risk characteristics of each subindex. The exchange rates will be, in such cases, equal to one, but other risk factors are introduced to capture the uncertainty of the subindices. For instance, an index of corporate bonds may consist of subindices for different rating categories or different industrial sectors, and each subindex has unique credit risk characteristics. Exchange rate risk is not present—for investors denominated in the base currency—in this application, but credit risk becomes a dominant risk factor.

24.3.1 Creating a global index

Consider a broad international index of K markets. Each market presents its own risk characteristics, and we denote markets by a subscript j. An index, summarizing the changes

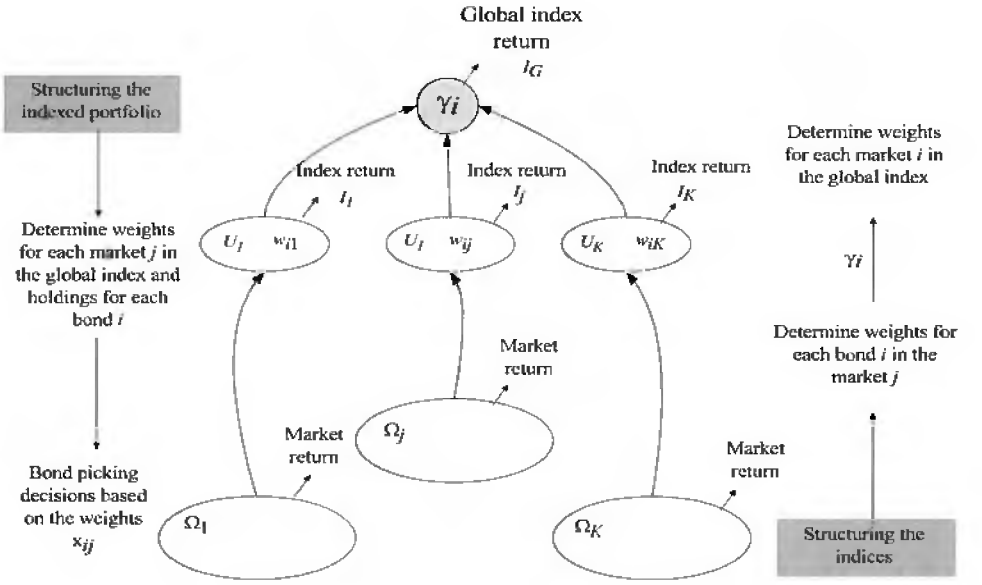


Figure 24.2. *The bottom-up process for structuring the market indices and the top-down approach for structuring an indexed portfolio.*

of each market $j = 1, 2, \ldots, K$, consists of securities in a representative sample $\mathcal{U}_j$ of size n. The sample is selected from the universe of available bonds Ω_j. (We assume for simplicity that the samples for all markets are of the same size.) For each security $i = 1, 2, \ldots, n$, in the representative set $\mathcal{U}_j$ the index specifies its weight w_{ij}. It is usually the case that these weights reflect the capitalization structure of the universe set Ω_j with bonds that have characteristics identical or similar to the ith bond. The global index assigns proportional weights, γ_j, to each country index based on the market value of each country index. Figure 24.1 shows the normalized weights γ_j in the 18 countries that compose the GBI broad.

In practice the sets Ω_j may consist of thousands or even hundreds of thousands of bonds differing by issuer, issue data, maturity date, coupon payments, etc. The representative sets $\mathcal{U}_j$ consist of a hundred or so representative securities. The process for constructing the global index from the country indices is illustrated in Figure 24.2.

The problem of an index fund manager is to determine the fraction of the portfolio value invested in each of the K markets and to pick specific bonds from each market Ω_j to add to the portfolio. These decisions can be made in two steps. First, make the strategic decision in setting the exposure of the portfolio to each market. Then suitable bonds are identified in each market to construct the country-specific portfolio. This portfolio has to track the country-specific index as well, and the tactical decisions can also be addressed using a tracking model. The time horizons of the two models might differ, however. The strategic decisions are usually made with a long time horizon—several months or even years. Bond picking decisions might have short horizons, such as weeks or months.

We denote the portfolio weights—as a proportion of total assets—allocated to the

index of each country $j = 1, 2, \ldots, K$ by z_j, and the proportional holdings in bond $i = 1, 2, \ldots, m$ in each country by x_{ij}. The strategic asset allocation model determines optimal weights z_j^*, and the bond picking model specifies optimal holdings such that $\sum_{i=1}^{n} x_{ij}^* = z_j^*$ for all $j = 1, 2, \ldots, K$. An integrated model determines jointly the optimal holdings x_{ij}^* and weights z_i^*. We develop first an integrated model for jointly determining the strategic and tactical decisions, and then we develop the optimization models for solving separately the problem of strategic asset allocation among indices and the tactical problem of bond picking in each market. The modeling process for structuring the indexed portfolios is also illustrated in Figure 24.2.

24.3.2 Integrated indexation models

Let r_{ij}^l denote the return of the ith bond in the jth currency in scenario l. This is the local return for investors denominated in the jth currency which is their domestic market. This return will usually differ from the return of the security when viewed by investors denominated in a currency other than j. To get the return for the later group of investors we scale the local return r_{ij}^l by the exchange rate appreciation or depreciation of the jth market currency against their base currency. The exchange rate appreciation is the ratio of the exchange rate E_j^l of currency j against the base currency in scenario l to the current exchange rate E_j^0, i.e., $e_j^l = E_j^l / E_j^0$. We can think of e_j^l as the total return of currency j.

The return of the international composite index in the base currency is

$$R_I(\gamma; , r^l) = \sum_{j=1}^{K} e_j^l \gamma_j R_j(w; r^l), \tag{24.21}$$

where $R_j(w; r^l) = \sum_{i=1}^{n} w_{ij} r_{ij}^l$ is the return of the index of the jth market as measured in the local currency and γ_j are the proportional weights of each country index in the global index. The return of the portfolio in the base currency is

$$R_p(x; r^l) = \sum_{j=1}^{K} e_j^l R_{pj}(x; r^l), \tag{24.22}$$

where

$$R_{pj}(x; r^l) = \sum_{i=1}^{n} r_{ij}^l x_{ij} \tag{24.23}$$

is the portfolio local return in the jth market.

We can now define the following tracking model, akin to Model 24.2.3. The model maximizes the expected portfolio return in the base currency and restricts the tracking error to be within $\pm\epsilon$.

Model 24.3.1 Integrated international indexation model

$$\text{Maximize} \quad \sum_{l \in \Omega} p^l \sum_{j=1}^{K} e_j^l R_{pj}(x; r^l) \tag{24.24}$$

$$\text{subject to} \quad -\epsilon \leq \sum_{j=1}^{K} e_j^l R_{pj}(x; r^l) - R_I(\gamma, r^l) \leq \epsilon$$

$$\text{for all } l \in \Omega, \tag{24.25}$$

$$\sum_{j=1}^{K} \sum_{i=1}^{n} x_{ij} = 1, \tag{24.26}$$

$$x \in X \subseteq \mathbb{R}_n^+. \tag{24.27}$$

If the solution of this model is denoted by x^*, we can estimate the exposure of the optimal portfolio to the jth currency by $z_j^* \doteq \sum_{i=1}^{n} x_{ij}^*$.

24.3.3 Nonintegrated models

We develop now two models that address separately the strategic asset allocation and the tactical asset allocation (i.e., the bond picking) problem. These models represent better the hierarchical operations of international portfolio managers that usually first determine their currency exposure, and then determine the portfolio holdings in each currency. We will see, however, that the integrated model produces results superior to those of the nonintegrated models.

The strategic asset allocation model

The return of a portfolio with normalized weights z_j takes the following scenario values, when measured with respect to the base currency:

$$R_p(z; R_{pj}(w, r^l)) = \sum_{j=1}^{K} e_j^l R_{pj}(w; r^l) z_j = \sum_{j=1}^{K} e_j^l \left(\sum_{i=1}^{n} w_{ij} r_{ij}^l \right) z_j.$$

The optimal normalized holdings in each market $(z_j^*)_{j=1}^K$ are determined by Model 24.3.2.

Model 24.3.2 Strategic model for international index funds

$$\text{Maximize} \quad \sum_{l \in \Omega} p^l \sum_{j=1}^{K} e_j^l R_{pj}(w; r^l) z_j \tag{24.28}$$

$$\text{subject to} \quad -\epsilon \leq \sum_{j=1}^{K} e_j^l R_j(w; r^l) z_j - R_I(\gamma; r^l) \leq \epsilon$$

$$\text{for all } l \in \Omega, \tag{24.29}$$

$$\sum_{j=1}^{K} z_j = 1, \tag{24.30}$$

$$z \geq 0. \tag{24.31}$$

The tactical bond picking model

Having obtained the optimal currency weights $(z_j^*)_{j=1}^K$ from Model 24.3.2, we can now use the following models to solve the bond picking problem for each constituent index. These models determine the optimal weights $(x_{ij}^*)_{i=1}^n$ for bond holdings in each currency $j = 1, 2, \ldots, K$, such that the optimal currency weights are preserved.

Model 24.3.3 Tactical model for international index funds
For each $j = 1, 2, \ldots, K$, *solve*

$$\text{Maximize} \quad \sum_{l \in \Omega} p^l R_{pj}(x; r^l) \tag{24.32}$$

$$\text{subject to} \quad -\epsilon \leq R_{pj}(x; r^l) - R_j(w; r^l) \leq \epsilon \quad \text{for all } l \in \Omega, \tag{24.33}$$

$$\sum_{i=1}^{n} x_{ij} = z_j^*, \tag{24.34}$$

$$x \in X \subseteq \mathbb{R}_n^+. \tag{24.35}$$

From Model 24.3.2, (24.30), we have that the weights in the K markets, z_j, add up to one, and it follows from (24.34) that $\sum_{j=1}^K \sum_{i=1}^n x_{ij}^* = 1$.

24.3.4 Operational model for international index funds

The models of this section can be formulated to account for the cost of transactions in rebalancing a portfolio and to allow for cash infusion or withdrawal, liquidity and diversification constraints, and other operational considerations. To incorporate these practical considerations the model variables are expressed not in percentages of total wealth, as was done above, but in face value. This choice of units is needed to model cash infusion or withdrawal. Constraints imposed due to trading liquidity considerations also require the modeling of face values. For instance, allocating the total wealth to a given market may not affect prices if the total portfolio value is small, but it may substantially affect prices for large portfolio values. (This was part of the problem facing long-term capital management when it had to unwind a very large position under extreme conditions.) We formulate here a model for building international index funds that incorporates operational constraints. The decision variables are redefined in terms of face values rather than as proportions of total assets, and some additional definitions are

x_{ij} face value invested in security i in the jth currency,

y_{ij} face value sold of security i in the jth currency,

z_{ij} face value of security i in the jth currency that remains as inventory in the indexed portfolio,

v^+ risk-free investment (i.e., cash) in the base currency.

Various constants and model parameters are also needed:

b_{0ij} face value of initial inventory of security i in the jth currency,

v_0 initial holdings in the risk-free asset (cash) in the base currency,

P^b_{0ij} current bid price of security i in the jth currency,

P^a_{0ij} current ask price of security i in the jth currency.

The difference between bid and ask prices reflects liquidity premia and transaction costs. In a highly liquid market these two prices differ only by the cost of the transaction. For illiquid securities, or for very large transactions, the gap between the bid and ask prices may widen.

With the above definitions we can now develop the model. The initial value of the portfolio is

$$V_0 = v_0 + \sum_{j=1}^{K} E^0_j \sum_{i=1}^{n} P^b_{0ij} b_{0ij}.$$

An *inventory balance* equation gives the face value of the inventory in bond i in the jth currency as a function of the investment and sale decisions

$$z_{ij} = b_{0ij} + x_{ij} - y_{ij} \quad \text{for all } i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, K. \tag{24.36}$$

Similarly a *cash flow balance* equation gives the amount invested in the risk-free asset (i.e., cash) in the base currency, as a function of the initial available cash v_0, any cash generated from security sales at the given bid prices, and any cash spent for investments at the given ask prices

$$v^+ = v_0 + \sum_{j=1}^{K} E^0_j \left(\sum_{i=1}^{n} P^b_{0ij} y_{ij} - \sum_{i=1}^{n} P^a_{0ij} x_{ij} \right). \tag{24.37}$$

With these definitions the value of the portfolio at the end of the holding period in scenario $l \in \Omega$ is

$$V^l_T = (1 + r^l_{fT}) v^+ + \sum_{j=1}^{K} E^l_j \sum_{i=1}^{n} (1 + r^l_{ij}) P^b_{0ij} z_{ij},$$

where r^l_{fT} is the risk-free rate of return of the base currency during the holding period T in scenario l.

The rate of return of the portfolio is

$$R^l_p(x; y, z, v^+; r^l) = \frac{V^l_T - V_0}{V_0}. \tag{24.38}$$

The following model determines an optimal index fund taking into account differences in the bid/ask prices and allowing for risk-free investments in cash as well.

Model 24.3.4 Operational model for index funds

$$\text{Maximize} \quad \sum_{l\in\Omega} p^l R_p^l(x, y, z, v^+; r^l) \tag{24.39}$$

$$\text{subject to} \quad z_{ij} - x_{ij} + y_{ij} = b_{0ij}$$

$$\text{for all } i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, K, \tag{24.40}$$

$$\sum_{j=1}^{K} E_j^0 \left(\sum_{i=1}^{n} P_{0ij}^a x_{ij} \right) + v^+ - \sum_{j=1}^{K} E_j^0 \left(\sum_{i=1}^{n} P_{0ij}^b y_{ij} \right) = v_0, \tag{24.41}$$

$$-\epsilon \le R_p^l(x, y, z, v^+; r^l) - R_I(\gamma; r^l) \le \epsilon$$

$$\text{for all } l \in \Omega, \tag{24.42}$$

$$x, y, z, v^+ \ge 0. \tag{24.43}$$

Equations (24.40) are the inventory balance constraints for all securities i in all currencies j. Equation (24.41) is the cash flow balance constraint. Inequalities (24.42) are the tracking constraints restricting the deviations of the indexed portfolio from the target index to be within $\pm\epsilon$. Bounds (24.43) restrict all variables to be nonnegative so that short sales are not allowed.

This model takes into account transaction costs in rebalancing an existing portfolio, as reflected in the spread between bid and ask prices. We can also add constraints to restrict holding in any one security to a fraction α of the total value of the portfolio

$$E_j^0 P_{0ij}^b z_{ij} \le \alpha \left(\sum_{j=1}^{K} E_j^0 \sum_{i=1}^{n} P_{0ij}^b z_{ij} \right) \text{ for all } i = 1, 2, \ldots, n, \quad j = 1, 2, \ldots, K.$$

Such constraints are useful for diversification purposes. They also serve in limiting liquidity risk by not allowing large positions in any single bond. For very large funds these positions may carry a liquidity premium.

24.4 Models for corporate bond index funds

The models of the previous section are also applicable, with some modifications, to problems of tracking corporate bond indices. Currency risk, of central concern to international portfolio managers, gives way to credit risk, which is a major source of risk for managers of credit-risky assets. We discuss here the key components of the models for tracking corporate bond indices.

Managing funds to track a corporate bond index is a challenging task for two reasons. First, the manager has to cope with the diverse sources of risk inherent in the corporate bond market. Second, the number of securities in the index is quite large (more than 1000 for the Merrill Lynch index we consider later), and it is prohibitively expensive for all but the largest funds to have holdings in all securities in the index.

The major risk factors of corporate bonds are fixed income market risk and credit risk. In particular, corporate bond prices are affected by the following events:

1. changes in the term structure of risk-free rates,
2. changes in the term structure of credit spreads,
3. changes in the rating of the bond,
4. the likelihood that a bond will go into default, and
5. the amount recovered if a bond goes into default.

Scenarios of holding period returns can be generated using Monte Carlo simulation of these risk factors and provide input data for optimization models to track corporate bond indices. The problem has similarities to the problem of managing international index funds, and the models of section 24.3.2 serve as the basis for the discussion. First, for investors denominated in the currency of the corporate bond index there is no exchange rate risk. For other investors there is only one exchange risk factor, namely, due to exchange rate fluctuations of the index currency against the investor's currency. For simplicity we assume the same currency for the index and the investor.

The corporate bond index is an index of subindices. In particular, each rating class (AAA, BBB, etc.) is a subindex of the market. In the context of Model 24.3.1 we use $j = 1, 2, \ldots, K$ to denote credit rating classes, and the exchange rate appreciation e_j^l is set identically equal to 1. The model is then applicable to the management of corporate bond portfolios.

It is common practice for corporate bond portfolio managers—as in the case of international portfolio managers—to separate the strategic from the tactical asset allocation decisions. Strategic asset allocation determines the broad allocation of assets among asset classes such as credit ratings, or industrial sectors, or maturity ranges, or any combination of these attributes. Tactical asset allocation will then pick specific bonds from each asset class so that the relevant subindex is tracked. Models 24.3.2 and 24.3.3 are applicable using, once more, $j = 1, 2, \ldots, K$ to denote the credit rating classes and setting properly the exchange rate appreciation.

24.5 Stochastic programming for index funds

We consider now the optimization of dynamic strategies for index tracking, using stochastic programming models. The model is developed on an event tree, assuming multiple trading dates. Multiple states of the index are plausible at each trading date. For each state we have available a set of bid and ask prices for all securities in the index—or returns of all subindices in a composite index—and the composition of the index. Thus state-dependent values of index returns can be calculated. Following the previous sections of this chapter, we develop a model to optimize the portfolios weights so that the portfolio return tracks the index return. However, in the multiperiod optimization framework of this section it is possible to rebalance the portfolio at each trading date, conditioned on the observed state. This model reflects more accurately the problem of indexed fund managers that rebalance their portfolio as more information arrives, with the objective of staying close to the index. We will also see that the stochastic programming models perform better, in ex post testing using out-of-sample data, for several real-world applications.

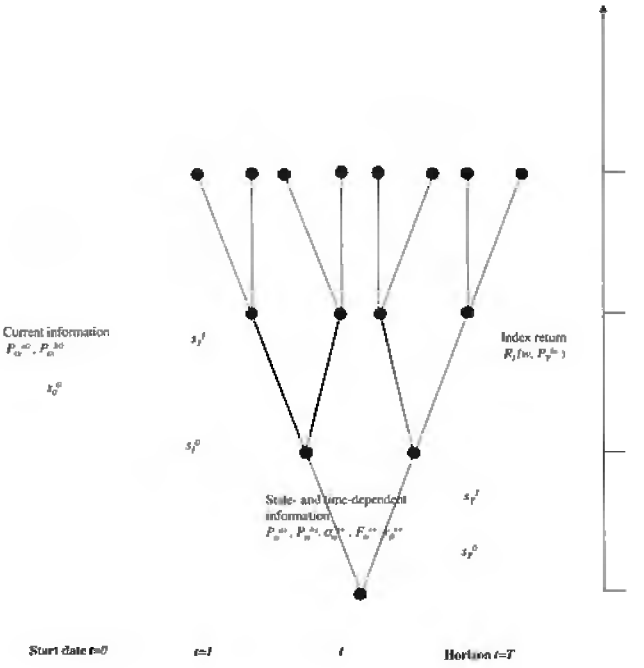


Figure 24.3. *Event tree with three trading dates and the random variables used in a stochastic programming model for index funds.*

24.5.1 Notation

To develop the dynamic portfolio optimization models we need to introduce variables for portfolio rebalancing at each trading date and for each state. The event tree in Figure 24.3 illustrates three trading dates with multiple states at each trading date after $t = 0$, i.e., today. The security and the index returns are indexed by state $s \in \Sigma_t$ at each trading date t. The difference from the indexation models of the previous sections when security and index returns were indexed by scenario $l \in \Omega$, using a linear scenario structure without utilizing any information arriving at intermediate time periods before the horizon. The time-dependent information about states from $t = 0$ to T is necessary for modeling portfolio rebalancing decisions. An event tree models the arrival of new information.

For each trading date and for each state we define variables to represent the buying and selling of securities, investments in the risk-free asset, and holdings of securities in the indexed portfolio. Investment decisions are in units of face value. The following variables are used, where time index t takes values over the trading dates $t = 0, 1, \ldots, T$ and state index s is from the set Σ_t:

x_{ti}^s face value invested in security i at period t in state s,

y_{ti}^s face value sold of security i at period t in state s,

z_{ti}^s face value of security i that remains as inventory in the indexed portfolio at period t

in state s,

v_t^{+s} cash invested in the risk-free rate at period t in state s.

We also need to model the dynamics of securities in the index on the event tree. In particular we need to know, at a given trading date t and state s, the bid and ask prices of securities in the index and the cash flows generated at the next trading date at successor state s^+. (The cash flows may be due to coupon payments, exercise of call options for callable bonds, default for corporate bonds, prepayments for mortgage securities, etc.) The following parameters are used:

$\alpha_{ti}^{s^+}$ *amortization factor* for security i from state s at period t to the successor state s^+ at period $t+1$;

$F_{ti}^{s^+}$ *cash flow* for security i, indicating cash generated by security i from t to $t+1$ per unit face value, due to scheduled dividend or coupon payments and exercise of any embedded options;

P_{ti}^{as} ask price of security i are period t in state s;

P_{ti}^{bs} bid price of security i are period t in state s;

$r_{ft}^{s^+}$: rate of return of the risk-free asset held from t to $t+1$. This value depends on the successor state s^+ reached at $t+1$ from state s at t.

The initial conditions for the portfolio at $t = 0$ are specified by the parameters

b_{0i} face value of initial holding of security i,

v_0 initial holdings in the risk-free asset.

24.5.2 Model formulation

The stochastic programming model for index tracking has two basic sets of constraints. One expresses cash flow accounting for the risk-free asset, i.e., cash, and the other is an inventory balance equation for each security in the indexed portfolio at all trading dates and for all states. We formulate the components of the model for $t = 0$ and for future trading dates $0 < t < T$. We have the first-stage problem at $t = 0$, and the t-stage problem models the recourse decisions.

First-stage constraints

At the first stage (i.e., at $t = 0$) all prices are known with certainty. We also know the portfolio composition, and the portfolio value is

$$V_0 = v_0 + \sum_{i=1}^{n} P_{0i}^{b0} b_{0i}. \tag{24.44}$$

For each asset class $i \in U$ in the portfolio we have an *inventory balance constraint*

$$z_{0i}^0 = b_{0i} + x_{0i}^0 - y_{0i}^0. \tag{24.45}$$

The *cash flow balance equation* specifies that the original endowment in the riskless asset, plus any proceeds from liquidating part of the existing portfolio, equal the amount invested in the purchase of new securities plus the amount invested in the riskless asset, i.e.,

$$\sum_{i=1}^{n} P_{0i}^{b0} y_{0i}^0 + v_0 = \sum_{i=1}^{n} P_{0i}^{a0} x_{0i}^0 + v_0^{+0}. \tag{24.46}$$

Time-staged constraints

Decisions made at future trading dates $t = 1, 2, \ldots, T$ are conditioned on the state $s \in \Sigma_t$ at every date. We have a set of constraints for each state at each time period. These decisions also depend on the investment decisions made at the previous trading date $t - 1$ at predecessor state s^-.

Asset inventory balance equations constrain the amount of each security sold or remaining in the portfolio to be equal to the outstanding amount of face value carried over from the previous trading date, plus any amount purchased at the current trading date. There is one constraint for each security $i \in U$ and for each state $s \in \Sigma_t$

$$z_{ti}^s = z_{(t-1)i}^{s^-} + x_{ti}^s - y_{ti}^s. \tag{24.47}$$

When dealing with instruments with embedded options—such as callable bonds that may be called, corporate bonds that may default, or mortgage securities that may prepay—we need to introduce amortization factors reflecting the exercise of the options. The asset inventory balance equation is

$$z_{ti}^s = \alpha_{(t-1)i}^s z_{(t-1)i}^{s^-} + x_{ti}^s - y_{ti}^s \text{ for all } i \in U,\ s \in \Sigma_t. \tag{24.48}$$

Cash flow balance requires that the amount invested in the purchase of new securities and in the risk-free asset is equal to the income generated by the existing portfolio during the holding period, plus any cash generated from sales and cash reinvested at the previous period at predecessor state s^-. There is one constraint for each state $s \in \Sigma_t$

$$\sum_{i=1}^{n} F_{(t-1)i}^s z_{(t-1)i}^{s^-} + \sum_{i=1}^{n} P_{ti}^{bs} y_{ti}^s + (1 + r_{f(t-1)}^s) v_{t-1}^{+s^-} = \sum_{i=1}^{n} P_{ti}^{as} x_{ti}^s + v_t^{+s}. \tag{24.49}$$

End-of-horizon constraints

At the end of the planning horizon we evaluate the value of the portfolio at each state $s \in \Sigma_T$ and restrict its deviations from the corresponding value of the index. The portfolio value will depend on the holdings in different asset classes, including cash, and the state $s \in \Sigma_T$. It is given by

$$V_T^s(z, v^+; P_T^{bs}) = v_T^{+s} + \sum_{i=1}^{n} P_{Ti}^{bs} z_{Ti}^s. \tag{24.50}$$

Similarly, if we invest an amount V_0 in the index, its value at the end of the horizon is

$$V_I^s(w; P_T^{bs}) = V_0(1 + R_I(w; r_T^s)), \tag{24.51}$$

where $r_T^s = (r_{Ti}^s)_{i=1}^n$, the rate of return of security i, is

$$r_{Ti}^s = \frac{P_{Ti}^{bs} - P_{0i}^{b0}}{P_{0i}^{b0}}. \tag{24.52}$$

The rate of return of the portfolio is

$$R_p^s(x, y, z, v^+; P_T^{bs}) = \frac{V_T^s - V_0}{V_0}, \tag{24.53}$$

and the following model maximizes the expected rate of return of the portfolio, while restricting its value to stay close to the index value.

Model 24.5.1 Stochastic programming for index funds

$$\text{Maximize} \quad \sum_{s \in \Sigma_T} p^s R_p^s(x, y, z, v^+; P_T^{bs}) \tag{24.54}$$

subject to

$$z_{0i}^0 = b_{0i} + x_{0i}^0 - y_{0i}^0$$
$$\text{for all } i \in \mathcal{U}, \tag{24.55}$$

$$\sum_{i=1}^n P_{0i}^{b0} y_{0i}^0 + v_0 = v_0^{+0} + \sum_{i=1}^n P_{0i}^{a0} x_{0i}^0, \tag{24.56}$$

$$z_{ti}^s = z_{(t-1)i}^{s^-} + x_{ti}^s - y_{ti}^s$$
$$\text{for all } t \in \mathcal{T}, s \in \Sigma_t, i \in U, \tag{24.57}$$

$$\sum_{i=1}^n F_{(t-1)i}^s z_{(t-1)i}^{s^-} + \sum_{i=1}^n P_{ti}^{bs} y_{ti}^s + (1 + r_{f(t-1)}^s) v_{t-1}^{+s^-}$$
$$= \sum_{i=1}^n P_{ti}^{as} x_{ti}^s + v_t^{+s}$$
$$\text{for all } t \in \mathcal{T}, s \in \Sigma_t, i \in U, \tag{24.58}$$

$$-\epsilon \le V_T^s(z, v^+; P_T^{bs}) - V_I^s(w; r_T^s) \le \epsilon$$
$$\text{for all } s \in \Sigma_T, \tag{24.59}$$

$$x, y, z, v^+ \ge 0. \tag{24.60}$$

The tracking constraint (24.59) is imposed only for the last time period T. The portfolio value will stay within $\pm\epsilon$ of the index value at the end of the horizon, but there is no assurance that it will stay close to the index at other trading dates. This model can be extended by imposing tracking constraints for other trading dates.

24.6 Applications of indexation models

The models of the previous sections are now applied to problems of managing indexed funds in diverse settings: (i) tracking an international government bond index, (ii) tracking a corporate bond index, (iii) creating enhanced index funds, (iv) tracking an index of mortgage-backed securities, and (v) tracking an index of callable bonds. The validation of the models is based on the ex post analysis of the performance of portfolios developed using the models. In general we expect that an "optimal" portfolio will perform well under the scenarios that were input in the modeling process. In this section, however, we study the performance of the indexed portfolios obtained by optimization models using simulated scenarios, under the realized market returns.

These applications serve many purposes. First, through the ex post testing in diverse settings it is demonstrated that the models of this chapter are effective tools in supporting portfolio managers of index funds.

Second, it is shown—in the application to international government bond indexation—that the integrative models generate portfolios that dominate the portfolios obtained using disintegrated approaches.

Third, it is shown—in the application to corporate bond indexation—that good tracking performance can be achieved by a strategic model that makes asset allocation decisions among subindices. However, extra value may be generated with tactical models that address bond picking decisions in an integrated fashion with the tactical asset allocation decision. It is also shown that, in the context of tracking GBIs, small corporate bond holdings can lead to superior risk return characteristics.

Fourth, it is shown—in the application to indexation of mortgage-backed securities and callable bonds—that stochastic programming models generate, ex post, performance superior to the single-period models.

Testing of the models proceeds along the following general lines. Using data available before the starting date of the backtesting period, say, first of month X, we calibrate appropriate Monte Carlo simulation models; generate scenarios of security returns, exchange rates, and index returns; and run the indexation model under investigation to select a tracking portfolio. We then move our clock a month forward to the first of month X+1, at which time we know the precise index return, security returns, and exchange rates. We can therefore determine the performance of our tracking portfolio and the tracking error. This completes one step of the backtest. We then use the data available up to month X+1 to recalibrate the Monte Carlo simulation models, generate new scenarios, and repeat the exercise. The process is repeated until the last month for which we have data. In most cases the experiments are repeated, in monthly steps, over a period of a few years.

24.6.1 Tracking an international GBI

Using the Salomon Brothers G7 index we create an equally weighted composite index of holdings in three major currencies, USD, DEM, and CHF. A single-sided version of the integrated tracking Model 24.3.1, whereby the right-hand inequality in constraint (24.25) is removed, is backtested in tracking this composite index over the period January, 1997, to July, 1998. Figure 24.4 shows the asset growth of 1000 USD invested in January, 1997, in the index and in portfolios generated using the models. The model tracks the volatile

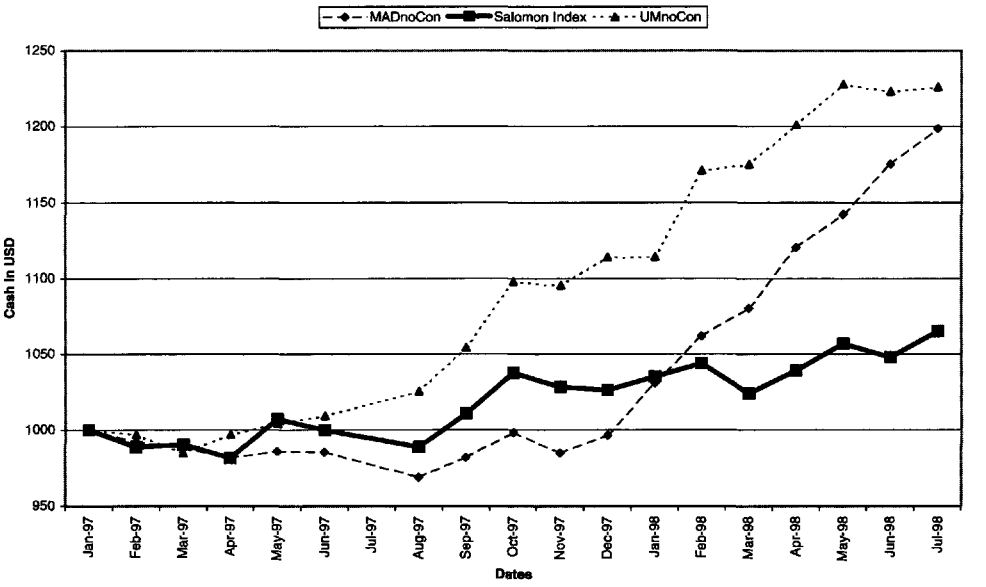


Figure 24.4. *Value of a US*$1000 *investment in a composite international GBI and in indexed portfolios.*

index well. The Sharpe ratios are −0.068 for the index and 0.369 for the indexed portfolios. Figure 24.4 shows the results of a single realization of the random index returns, namely, the historical realization during the period of the test. In some sense this is the only realization that matters. But how would the portfolio perform under alternative realizations of the random returns? Figure 24.5 illustrates the tracking error of a typical portfolio using out-of-sample scenarios. We observe from these simulated data that the tracking error is small and is limited to 1% (annualized) below the index under worst-case scenarios, while the lead over the index is 2.2% under the most favorable scenario.

To compare the integrated model tested here to the nonintegrated models of section 24.3.3, we develop now efficient frontiers—with a mean absolute deviation model—using both the integrated and the disintegrated models. For the disintegrated models we first solve the strategic asset allocation model and set the currency exposure to the one that yields the highest expected return. We then develop the efficient frontiers of portfolios in the three currencies such that the total exposure in each currency is equal to that determined by the strategic asset allocation model. With this hierarchical solution of the models we pick an indexed portfolio that has the highest expected return, which is consistent with the objective function of the integrated tracking model. Figure 24.6 shows the efficient frontiers obtained with both the integrated and the disintegrated models. Integration entails substantial benefits for the portfolio managers by reducing the volatility of the portfolio returns while also increasing somewhat the expected return.

A careful inspection of the asset weights, however, reveals that the integrated model generates poorly diversified portfolios. Left without any operational constraints the model would occasionally generate portfolios with holdings up to 99% in bonds in a single cur-

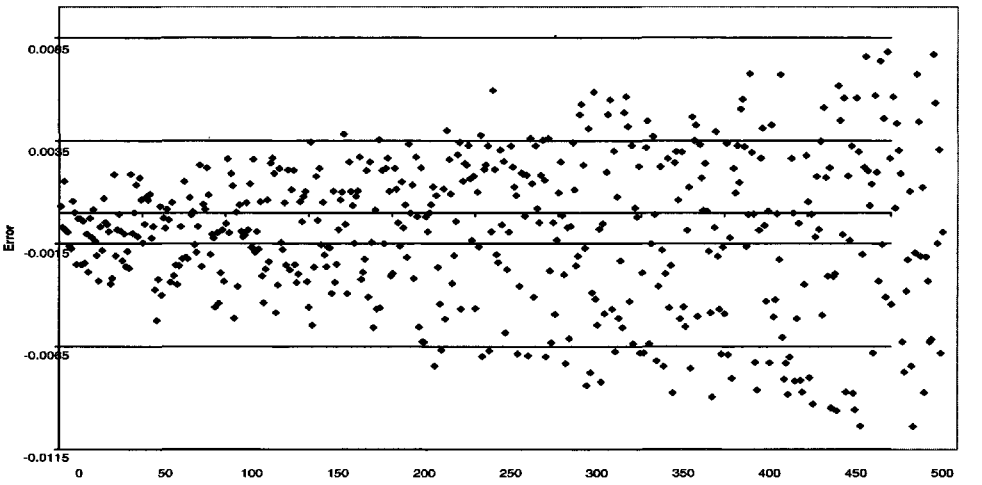


Figure 24.5. *Tracking error in annualized BP of a typical portfolio against the composite index for* 500 *simulated scenarios of returns and exchange rates.*

rency. When the comparison between the integrated and disintegrated models is repeated by imposing bounds on the total exposure in a single instrument, the efficiency gap of the two models is somewhat reduced. The integrated model with bounds still yields dominating portfolios, and an inspection of the weights shows that the portfolios of the integrated model are now well diversified, similar to the portfolios of the disintegrated models. In backtesting of the two models we also observe that the integrated model outperforms the disintegrated model; see Figure 24.6.

24.6.2 Tracking a corporate bond index

We now apply the models to develop indexed portfolios to track the Merrill Lynch Euro Dollar index. Similar to the case of international government bond indexation, we study the ex post performance of the model.

We start our experiments on January 31, 1999, and generate 3-month holding period return scenarios for all bonds with maturities up to 10 years and for each credit rating class using a simulation model calibrated to information available up to end of January only. The tracking model is then used to select a portfolio. We then move the clock 1 month forward, at which point (end of February, 1999) we know the bond returns and index performance and can therefore calculate the ex post performance of the tracking portfolio. Using now the information available up to February 28, 1999, we repeat the simulation, optimization, and performance analysis. This process is repeated until July 31, 2001. Transaction costs are considered for all trades during the backtesting period. We assume the same transaction cost for all bonds within a rating class: 5 BP for Aaa, 10 BP for Aa, 20 BP for A, and 40 BP for Baa.

We start by creating index funds using a strategic asset allocation model to allocate assets among the subindices that comprise the Merrill Lynch Euro dollar index. (These

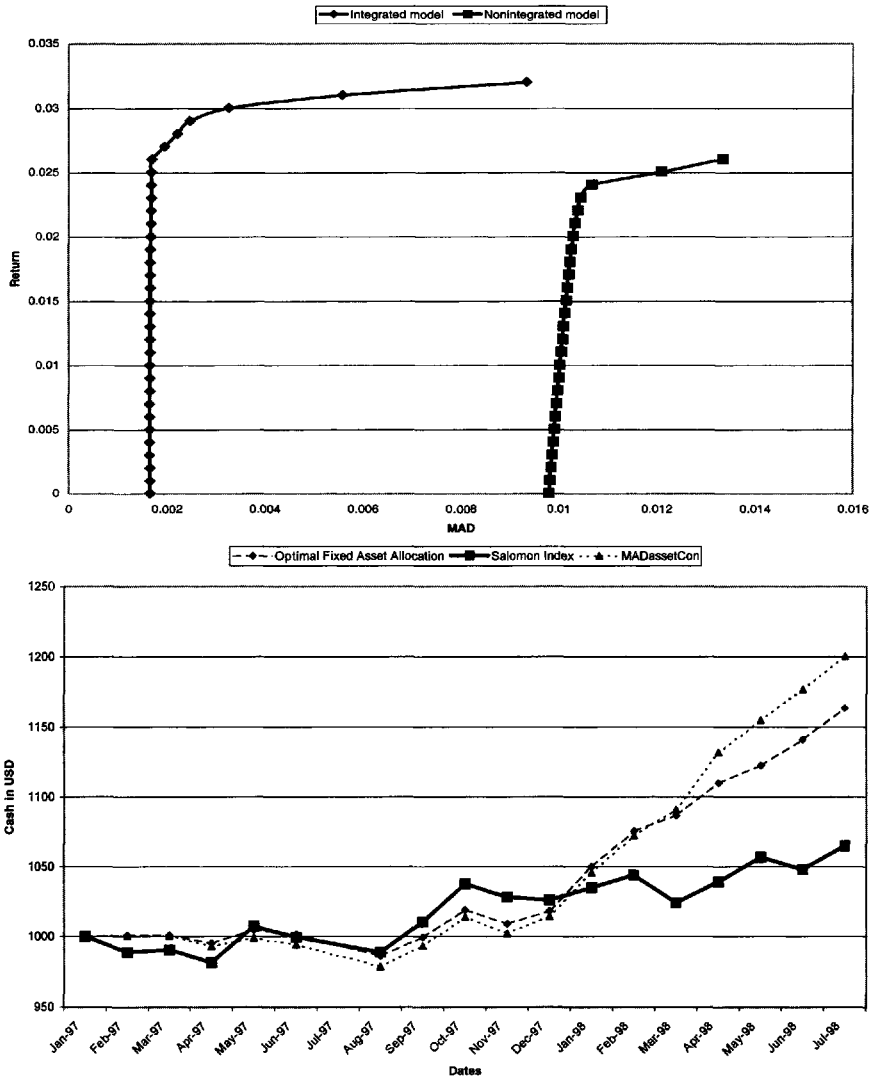


Figure 24.6. *The effects of integrating several financial decisions: The efficient frontier generated with the integrated and with the disintegrated models (top) and value of a US*\$100 *investment in portfolios generated with the integrated and the disintegrated models (bottom).*

subindices measure the performance of different asset classes of the corporate bond market such as by credit rating category, by industrial sector, by maturity, and for various combinations of these sectors.)

Figure 24.7 shows the value of a 100 USD investment in the index and the portfolio, and the tracking errors of the asset allocation when we define asset classes consisting of

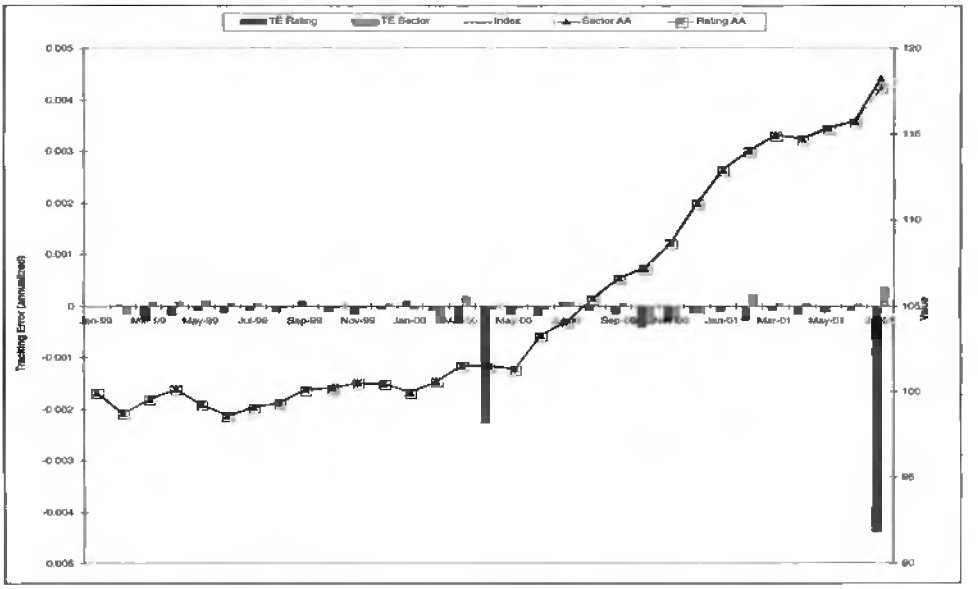


Figure 24.7. *Performance of index funds created using broad asset allocation models.*

securities with the same credit rating. We also show on the same figure the tracking errors when asset classes are defined by industrial sector rather than by rating. The portfolio tracks the index very closely; however, we do not seem to accumulate any extra value. The optimal portfolios have an almost identical structure to the index, and hence the portfolio growth follows closely the index growth.

We apply now the integrated indexation models to create indexed portfolios by picking securities directly from the universe of corporate bonds without any restrictions in tracking the subindices. Figure 24.8 shows the growth of 100 USD invested in the index on January 31, 1999, and in the optimal portfolios obtained with the model. The annualized tracking errors for each month are also shown.

Given the portfolio and index returns during this 30-month period, we calculate the historical Sharpe ratio for the tracking portfolio with respect to the index returns as 0.497; this is an encouraging statistic. The tracking errors are small on average and, as expected, the model underperforms by small amounts only in 5 months. Comparing with the results shown in Figure 24.7, we observe that indexed portfolios created using an integrative approach perform better than portfolios that deal with asset allocation and bond picking separately.

24.6.3 Enhanced index funds

We now consider the development of an enhanced index fund, to outperform an index while preserving the key risk characteristics of the index. In particular, we consider the example of staying within a positive margin of the GBI by assuming some exposure in the credit risk market. Of course, the enhancement in returns is coming by assuming some extra risks. Perhaps the most significant advance of the integrated models of this chapter is that they

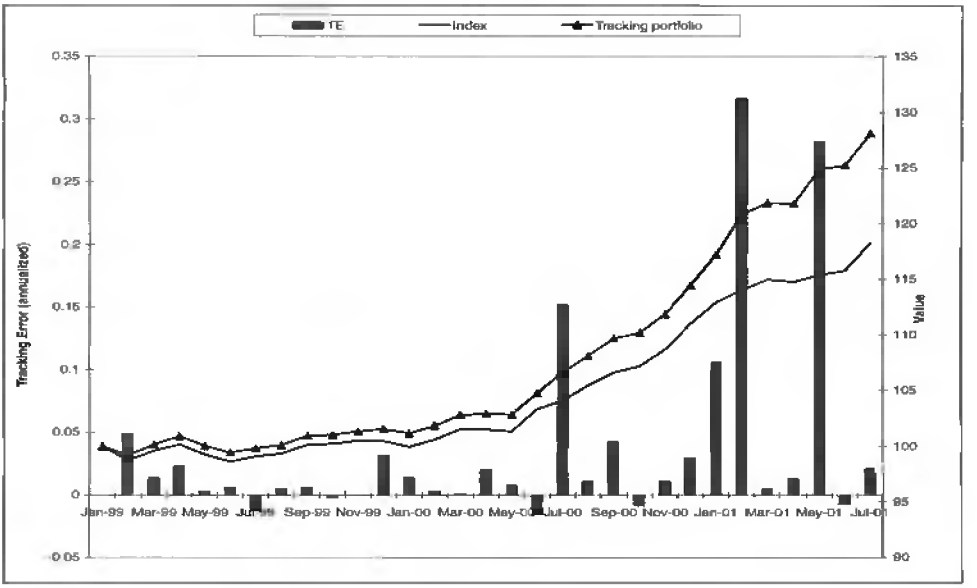


Figure 24.8. *Value of a US*\$100 *investment in the Merrill Lynch Euro Dollar index and in indexed portfolios.*

allow us to integrate market risk, credit spread, migration, and default risk so that we are now able to take a more holistic view of a manager's portfolio problem. This is demonstrated by creating indexed portfolios of treasury and corporate bonds to track a U.S. Treasury index. Integrating credit-risky bonds in government bond portfolios results in an enhanced index performance.

We focus on the Merrill Lynch U.S. Treasury index. The evolution of the U.S. Treasury index during our backtesting horizon is shown in Figure 24.9. As a first step we track this index by using only U.S. Treasury securities. We ignore all bonds with optional payoffs, such as callable or putable bonds. A bid-ask spread of 5 BP is assumed to capture transaction costs. Figure 24.9 shows the results of the experiment throughout the period. Tracking errors are very small. The historical Sharpe ratio is 0.04, and the model does not pick up extra value from the bond picking. This is expected, given the efficiency of the government bond market.

We now expand the universe of bonds for creating an enhanced indexed portfolio to include also the bonds from the Euro dollar index. The broader asset universe offers additional opportunities which may lead to improved performance, as shown in Figure 24.10 with the ex post analysis of the model.

We observe lower and less volatile tracking errors. Including corporate bonds improves the tracking performance, and the Sharpe ratio increases significantly from 0.04 to 0.31. In the optimized portfolio only a small fraction (approximately 5%) is invested in a few corporate bonds (approximately 10). Hence the increased downside risk of corporate bonds is taken into consideration by investing only a small fraction of the value in the risky asset class, and within this class, the model invests in a range of corporate bonds to minimize the exposure to a single security.

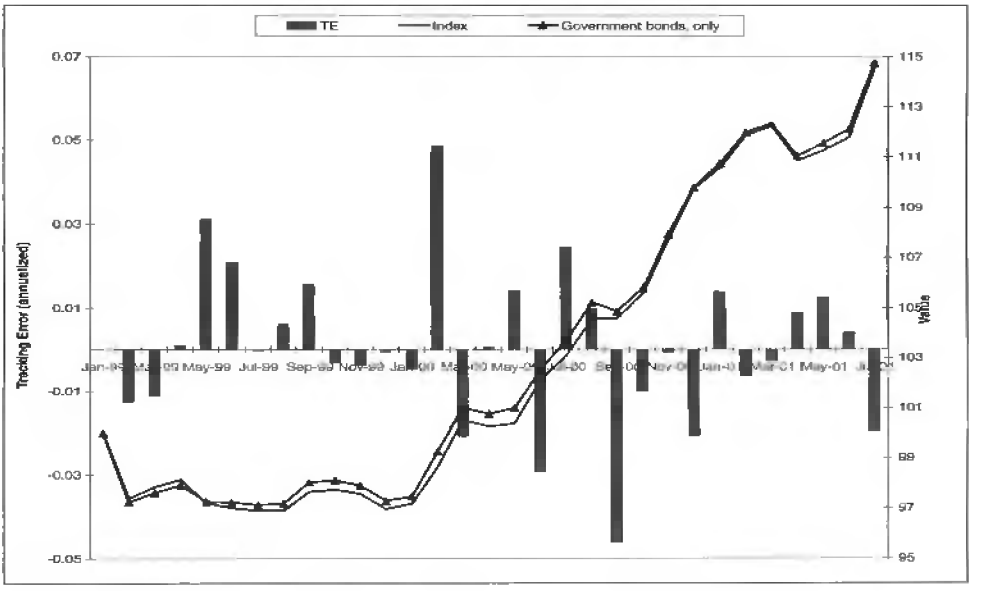


Figure 24.9. *Value of a US*$100 *investment in the Merrill Lynch U.S. Treasury index and in indexed portfolios constructed using treasury securities only.*

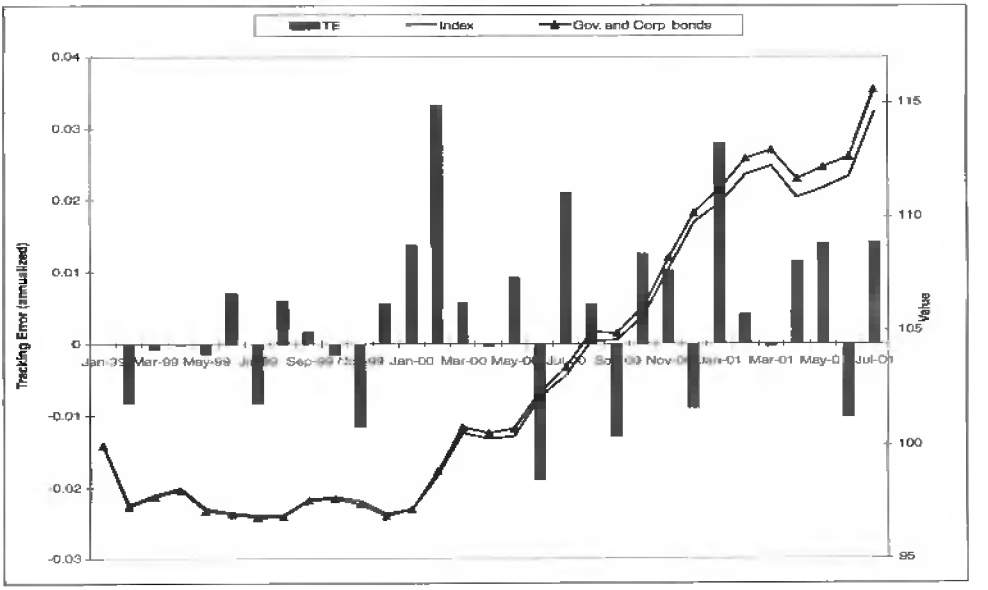


Figure 24.10. *Tracking performance of index-plus portfolios that track the Merrill Lynch U.S. Treasury index by investing in both treasury securities and corporate bonds.*

24.6.4 Stochastic programming models for index tracking

We now demonstrate, through ex post testing, that stochastic programming models for index funds perform better than the single-period models. The stochastic programming model is expected to perform better than the single-period models when applied to the same, simulated, scenarios, since the former model has one more degree of freedom, namely, the ability to revise the portfolio before the end of the planning horizon. The tests of this section address the following questions: "Do the stochastic programming models perform well ex post in tracking the marked indices?" and "Does a stochastic programming model for tracking an index outperform a single-period model?" The answer to both questions is affirmative as supported by the empirical results.

In the process we test the indexation models on two additional market indices—an index of mortgage-backed securities and an index of corporate callable bonds. Portfolios generated using the indexation models perform well for these two complex asset classes that contain options (call options for the callable bond market) or embedded options (the option for early repayment of the mortgage loan for the mortgage-backed securities market). The results provide further support to the findings of the previous sections on the effectiveness of the indexation models.

Index funds of mortgage-backed securities

Mortgage-backed securities are created when mortgages are pooled together and undivided interests or participation in the pool are sold. Understanding the features of mortgage-backed securities is essential in developing scenarios of security returns but plays no role in the indexation models tested here once the scenarios are somehow generated.

We develop indexed portfolios to track the Salomon Brothers mortgage index. This index captures the market trends of approximately 300,000 mortgage pools with a market value in excess of $1.5 trillion. This large universe of securities is classified into generic securities characterized by the coupon rate, issuing agency, issuing year, and remaining term. During the period of our backtesting, the Salomon Brothers index comprised between 118 and 144 generic securities. The indexation models create a portfolio consisting of a small number of generic securities, and then securities are bought from the available mortgage pool with characteristics that are identical—or very similar—to the generic securities recommended by the model.

We start our experiments on January 1, 1989, and generate 1-month holding period return scenarios for all securities in the index. Security pricing and scenario generation is done consistently with the risk-free term structures available on that date, and the model is calibrated to information available up to that day only. The tracking portfolio optimization model is then used to select a portfolio. For any securities that were added or dropped from the portfolio a transaction fee of 1/16th BP was charged. We then move the clock 1 month forward, at which point (February, 1989) we know the bond returns and index performance and can therefore calculate the ex post performance of the tracking portfolio. Using now the updated information available on February 1, 1989, we repeat the simulation, optimization, and performance analysis. This process is repeated until December, 1991.

Figure 24.11 shows the return of a 100 USD investment in the Salomon index during the 3-year period 1989–1991, together with the performance of indexed portfolios created using the single-period Model 24.2.2 and a single-sided version of the stochastic programming Model 24.5.1 whereby the right-hand inequality of (24.59) is relaxed. The tracking errors are small on average for both models, and the stochastic programming model performs better than the single-period model. We observe from this figure that the downside tracking error of the indexed portfolios generated using stochastic programming is less than the tracking error of the indexed portfolios generated by a single-period model.

The index realized an annualized return of 14.05% during the testing period, and the portfolios created by the single-period and the stochastic programming models realized returns of 14.18% and 15.10%, respectively. Given the returns of the two portfolios and the index during this 3-year period, we calculate the historical Sharpe ratio for the tracking portfolios with respect to the index returns as 0.149 (single-period model portfolios) and 1.071 (stochastic programming model portfolios). These are encouraging statistics for the portfolios generated by both models.

Index funds of callable bonds

Callable bonds are usually issued by corporations or agencies that reserve the right to call the bond before maturity. Holders of these securities face, in addition to fixed income market risk, the risk of exercise of the call option by the issuer and the risk of default.

We now compare a stochastic programming model and a single-period optimization model in tracking an index of callable bonds. The index was created by selecting a set of 230 securities out of a universe of 600 corporate bonds, excluding junk bonds of rating BBB or lower. Bonds in the universe were issued by financial institutions (18%), credit and banking institutions (17%), telecommunication companies (18%), utilities (17%), department stores (7%), food chains (7%), chemical companies (4%), clothing companies (5%), automobile producers (5%), and other industries (2%). The index of 230 securities was equally weighted.

The backtesting methodology is identical to the one we used in testing the mortgage indexation models. Our testing covers the period January, 1992, to February 1, 1993. Figure 24.12 shows the growth of a 100 USD investment in the index during the 14-month period of the testing, together with the performance of indexed portfolios created using the single-period Model 24.2.2 and a single-sided version of the stochastic programming Model 24.5.1. We see from this figure that the downside tracking error of the indexed portfolios generated using stochastic programming is less than the tracking error of the portfolios generated by a single-period model.

The index realized an annualized return of 11.63% during the testing period, and the portfolios created by the single-period and the stochastic programming models realized returns of 12.36% and 13.79%, respectively. The Sharpe ratio during the testing period for the tracking portfolios with respect to the index returns is 0.121 for the single-period model portfolios and 0.589 for the stochastic programming model portfolios. These results are consistent with the results obtained with the mortgage indexation models: the index funds created using a stochastic programming model perform better than index funds created with single-period models.

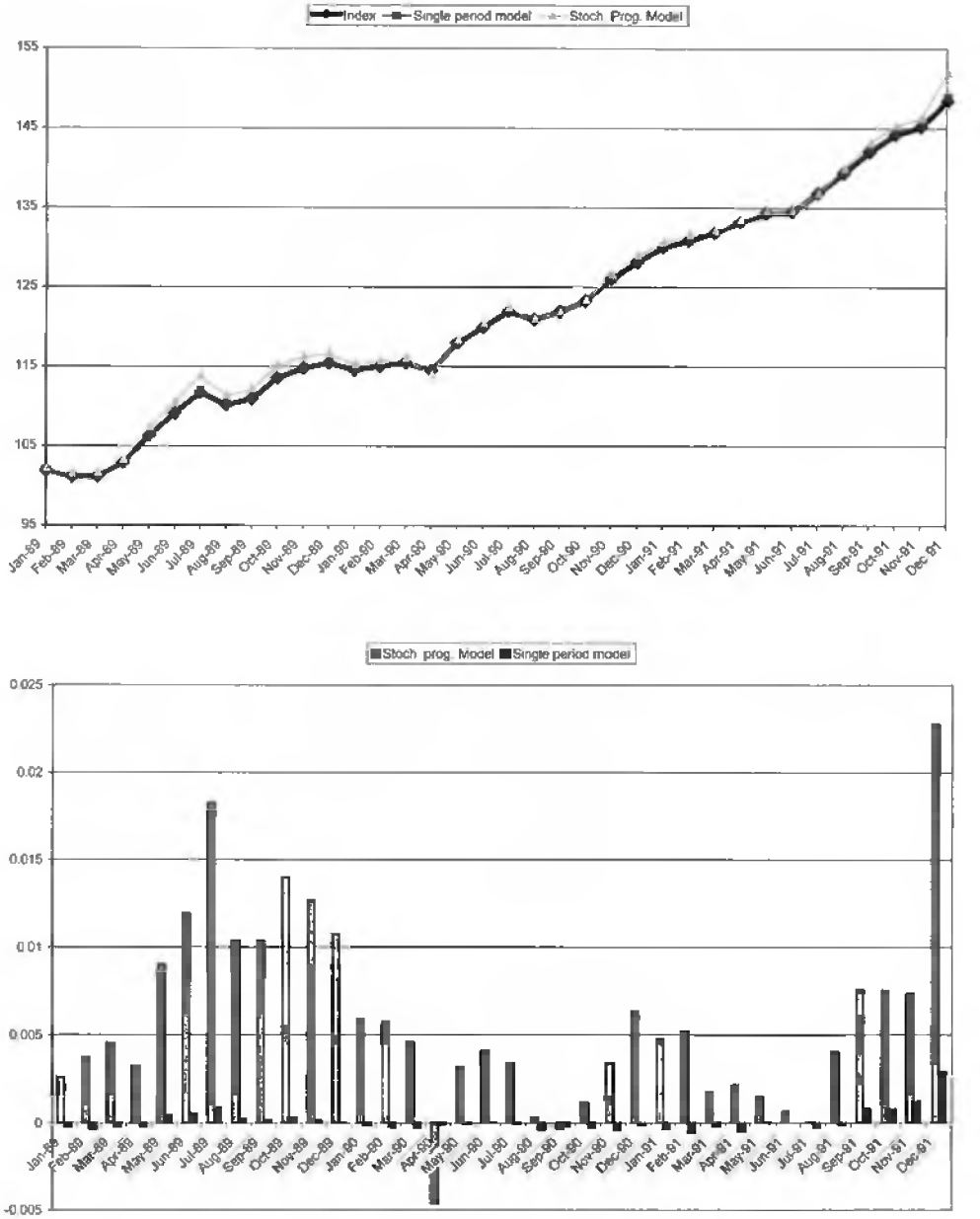


Figure 24.11. *Value of a US*\$100 *investment in the Salomon Brothers index of mortgage-backed securities and in indexed portfolios developed using a single-period and a stochastic programming model* (*top*), *and the tracking errors of the two indexed portfolios developed using a single-period and a stochastic programming model vis-à-vis the index* (*bottom*).

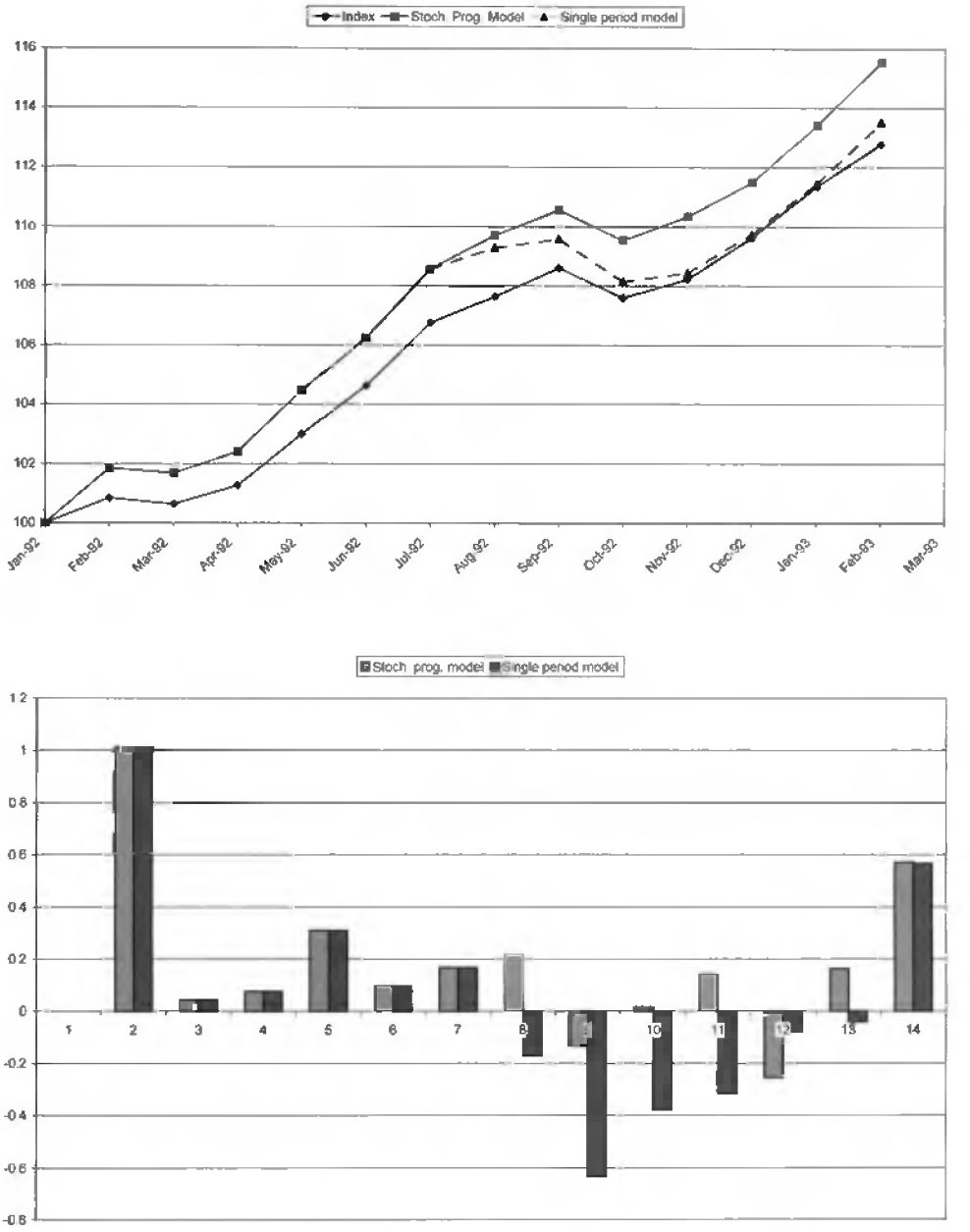


Figure 24.12. *Value of a US*\$100 *investment in the callable bond index and in indexed portfolios developed using a single-period and a stochastic programming model (top), and the tracking errors of the two indexed portfolios developed using a single-period and a stochastic programming model vis-à-vis the index (bottom).*

24.7 Notes and references

For an introduction to index funds, see Mossavar-Rahmani [11] and the references cited therein. Sources for material on market indices are the web pages of J. P. Morgan (http://www.jpmorgan.com/), Standard and Poor (http://www.standardpoor.com/), and the FTSE (http://www.ftse.com/).

An analysis of the performance of actively managed funds is due to Lakonishok, Shleifer, and Vishny [9]. The effect on prices when a stock is added in an index was studied by Bos [2].

A review of quantitative approaches for bond indexation is given by Seix and Akhoury [12]. Scenario optimization models were introduced by Dembo [4]. Miller, Krawitt, and Wands [10] and Worzel, Vassiadou-Zeniou, and Zenios [14] discuss the indexation of mortgage-backed securities. International bond portfolio indexation models were developed by Consiglio and Zenios [3], for callable bonds by Vassiadou-Zeniou and Zenios [13], and for corporate bonds by Jobst and Zenios [5, 7, 6]. For general background material on stochastic programming, see the books by Kall and Wallace [8] and Birge and Louveaux [1].

Several practical considerations that may be relevant for some managers of indexed funds are not discussed in this chapter. For instance, upper and lower bounds on the holdings in any security may also be imposed to limit very small positions that imply higher management costs and very large positions with significant exposure to security-specific risks and perhaps liquidity risk. Zenios [15] shows how such constraints may be incorporated.

Acknowledgment

This research was partially supported by EC grant ICA1-CT-2000-70015.

Bibliography

[1] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer, Heidelberg, 1997.

[2] R. J. Bos, *Event Study: Quantifying the Effect of Being Added to an S&P Index*, Technical Report, Standard & Poor's Quantitative Services, New York, 2000.

[3] A. Consiglio and S. A. Zenios, *Integrated simulation and optimization models for tracking international fixed income indices*, Math. Program. Ser. B, 89 (2001), pp. 311–339.

[4] R. Dembo, *Scenario optimization*, Ann. Oper. Res., 30 (1991), pp. 63–80.

[5] N. J. Jobst and S. A. Zenios, *Extending Credit Risk (Pricing) Models for Simulation and Valuation of Portfolios of Interest Rate and Credit Risk Sensitive Securities*, Working Paper 01–03, HERMES Center on Computational Finance and Economics, University of Cyprus, Nicosia, Cyprus, 2001.

[6] N. J. JOBST AND S. A. ZENIOS, *The tail that wags the dog: Integrating credit risk in asset portfolios*, J. Risk Finance, 3 (2001), pp. 31–43.

[7] N. J. JOBST AND S. A. ZENIOS, *Tracking Corporate Bond Indices in an Integrated Market and Credit Risk Environment*, Working Paper 01–04, HERMES Center on Computational Finance and Economics, University of Cyprus, Nicosia, Cyprus, 2001.

[8] P. KALL AND S. W. WALLACE, *Stochastic Programming*, John Wiley, New York, 1994.

[9] J. LAKONISHOK, A. SHLEIFER, AND R. V. VISHNY, *The structure and performance of the money management industry*, Brookings Papers on Economic Activity, 1992, pp. 339–391.

[10] L. MILLER, E. P. KRAWITT, AND M. P. WANDS, *Mortgage-backed securities indexation*, in The Handbook of Mortgage Backed Securities, F. J. Fabozzi, ed., Probus Publishing Company, Chicago, 1985, pp. 53–76.

[11] S. MOSSAVAR-RAHMANI, *Indexing fixed income assets*, in The Handbook of Fixed Income Securities, F. J. Fabozzi, ed., McGraw–Hill, 1997, pp. 913–924.

[12] C. SEIX AND R. AKHOURY, *Bond indexation: The optimal quantitative approach*, J. Portfolio Management, Spring (1986), pp. 50–53.

[13] C. VASSIADOU-ZENIOU AND S. A. ZENIOS, *Robust optimization models for managing callable bond portfolios*, Eur. J. Oper. Res., 91 (1996), pp. 264–273.

[14] K. J. WORZEL, C. VASSIADOU-ZENIOU, AND S. A. ZENIOS, *Integrated simulation and optimization models for tracking fixed-income indices*, Oper. Res., 42 (1996), pp. 223–233.

[15] S. A. ZENIOS, *Practical Financial Optimization: Decision Making for Financial Engineers*, manuscript, 2005.

This page intentionally left blank

Chapter 25

Decentralized Risk Management for Global Property and Casualty Insurance Companies

John M. Mulvey and Hafize Gaye Erkan**

25.1 Overview of DFA (centralized)

Dynamic financial analysis (DFA) provides a tool to analyze various business strategies and risk/return structures within enterprise-wide planning systems. A DFA aims at maximizing the shareholder value and tracking the free cash flow over time. Leading insurance and reinsurance companies have begun applying DFA to increase profitability, reduce enterprise risks, and identify the optimal capital structure of the firm. A DFA process should analyze the financial status of an insurance enterprise, namely, the ability of the firm's capital and earnings path to adequately support its future operations in light of stochastic external factors affecting the enterprise. A DFA model should combine the asset/liability structure of the enterprise and dynamic optimization of the strategies together with headquarters decisions. A DFA system consists of three major elements: a stochastic scenario generator (also see [14] for generating scenarios over a stochastic programming tree and see [4] and [9] for different scenario-generation methods), a multiperiod simulator, and an optimization module; see Figure 25.1.

Linking the assets and liabilities in a consistent fashion requires modeling the driving factors; e.g., see Figure 25.2. The factor models are well placed to support DFA. DFA is described further in [10].

Considering these external dynamic factors, a scenario tree is built up (Figure 25.3).

Today, there are two practical approaches for optimizing a multiperiod DFA system. The first involves stochastic programs. An alternative to stochastic programming involves developing a set of rules or policies to guide the company across the planning period at each decision node; this approach is called policy optimization. Each of the two

*Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544 (mulvey@princeton.edu, herkan@princeton.edu).

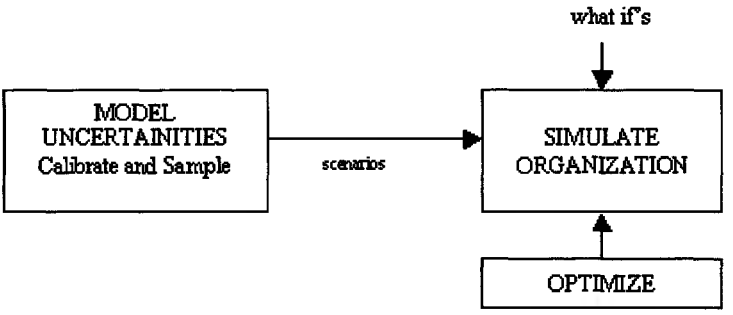


Figure 25.1. *Optimization module is one of the major components of the DFA system.*

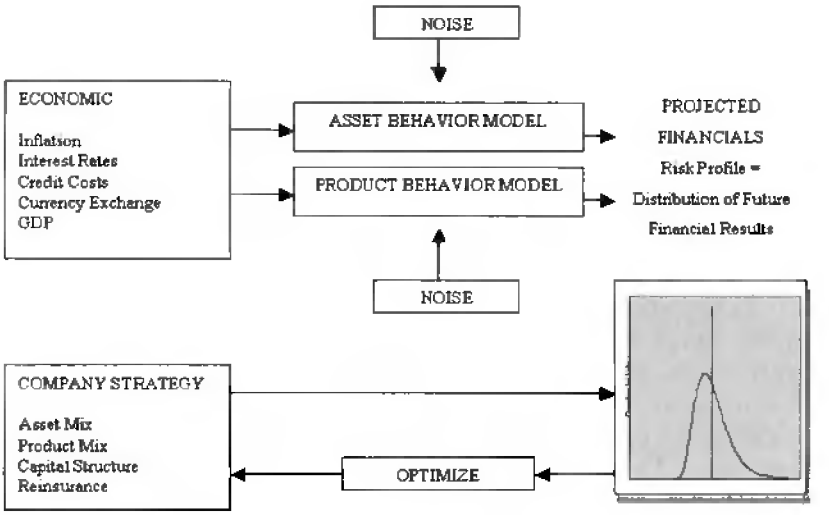


Figure 25.2. *DFA factor models.*

approaches—stochastic programming and policy optimization—has something to offer in the DFA context. Policy optimization is perhaps the easiest to implement. Of course, the scenario generator must be constructed and validated, along with a dependable policy rule such as the fixed-mix asset allocation. The resulting model becomes a Monte Carlo simulation in which the policy rule and the accompanying policy parameter setting are fixed. In addition, we can readily calculate the sampling errors and related statistical estimates. The recommended solutions can be evaluated using sensitivity analysis regarding the assumptions within the scenario generator.

On the other hand, stochastic programming has the potential for improving the recommendations, as compared with policy optimization. There are several provisos, however. First, the number of scenarios must be good enough to prevent the model from misestimating the full range of uncertainties. Second, the model must be solvable in a reasonable computer run time so that sensitivity analyses can be conducted. Third, the recommendations

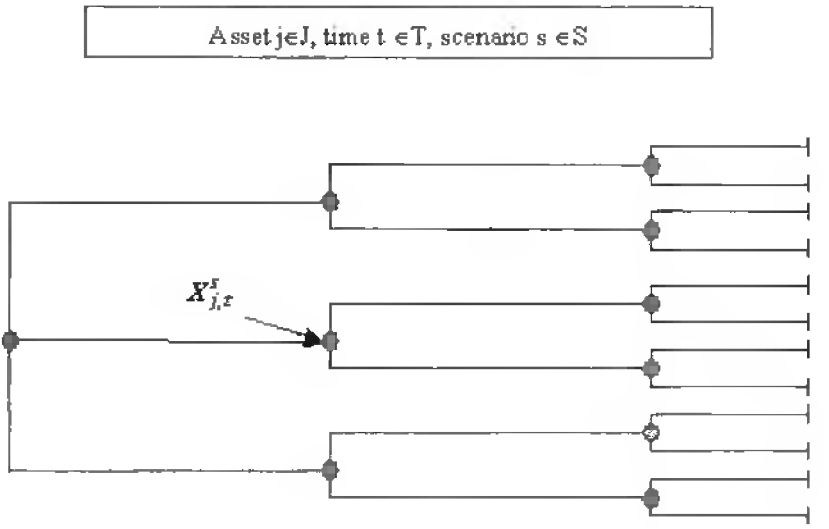


Figure 25.3. *Scenario tree.*

of the stochastic program should be understandable to the high-level executives who are ultimately responsible for the decisions. These issues present challenges to a modeler who wishes to optimize a DFA system. DFA models especially require a perfect selection of the actual scenario set S to adequately cover the range of risks encountered by insurance and other financial institutions (see [7] and [12] for several alternative sampling procedures).

25.2 Use of optimization for reducing risks and improving profits

In the light of [3], insurance companies can also be analyzed within the DFA context to reduce the enterprise risks and to improve the company's businesses. The insurance portfolio manager and underwriter require sophisticated analytical tools to assist decision making. The insurance portfolio manager needs to understand the effects of adding an additional account to the business line. In addition, there are many other issues the manager must address, such as the following: (1) Should an existing account be renewed and, if so, at what price? (2) Where are the best areas to expand the current portfolio? (3) How can two books of business be merged profitably?

The developed decision support system, called SmartWriter, answers these questions for one application area, the catastrophe property business. SmartWriter employs data from earthquake and hurricane modeling systems to show the effects of adding a new account or subtracting an existing account from the current portfolio. In addition, SmartWriter optimizes the portfolio composition to produce a portfolio meeting user-specified characteristics.

25.2.1 Evaluating new business

Suppose there is a portfolio of insurance liabilities. As an example, Berger, Mulvey, and Nish [3] consider a portfolio of commercial businesses insured against earthquakes in California by St. Paul, a large property and casualty insurance company. A potential new piece of business is presented to the portfolio manager, who must decide whether to write the account or reject it. Of course, some negotiating with the insurance broker who presents the account is possible, so the portfolio manager would also like to know the required premium to meet a profitability hurdle. Before analyzing the incremental business, there is a need to define a profitability measure for the existing portfolio. Two measures are return on allocated capital and expected utility (see [2] for an example). Their comparison is summarized in Table 25.1.

Table 25.1. *Comparison of allocated capital and expected utility objective function.*

	Advantages	Disadvantages
Allocated Capital	- Easy to explain - Returns have intuitive meaning	- Extra work to sort discrete distributions - Limited points on loss distribution
Expected Utility	- Handle entire loss distribution at once - Convex math program	- Hard to determine utility function - Results not intuitive

Sample decision

Table 25.2 presents a SmartWriter analysis of an account recently offered to USF&G's commercial property business.

The SmartWriter output is divided into three columns. The first column is the new account as a stand-alone business. The expected income for the account, after taking expenses and expected catastrophe losses from the premium, is $615,000. The new account requires $4,200,000 in capital based on the 1-in-100 year loss of $5,200,000. This yields a return of 14.6%, which is below our hurdle rate of 15%.

The second column contains data on the portfolio as it stands today, and the final column is the portfolio performance if the new account were added. The capital requirement for the combined portfolio is less than the sum of the new account and current portfolio capital. This indicates that the new account will diversify the business to some extent. Two

Table 25.2. *New account analysis. All numbers in* \$000 *except where indicated.*

	New Account	Current Portfolio	Combined
Premium	\$980	\$3,800	\$4,780
Expenses	\$294	\$1,140	\$1,434
Expected Catastrophe Loss	\$71	\$615	\$686
Expected Profit	**\$615**	**\$2,045**	**\$2,660**
Loss at $99^{th}\% = F^{-1}(0.99)$	\$5,200	\$14,300	\$18,100
Capital Required	\$4,200	\$11,600	\$14,700
Return on Capital: ROC	**14.6%**	17.6%	18.1%
Return on Marginal: ROMAC	**19.8%**		

additional items help quantify this diversification. The return on margin (ROMAC) for the new account is 19.8%, which means that the marginal return for adding the account divided by the marginal capital is significantly over the hurdle rate. The second item is the increase in the return on capital (ROC) for the portfolio from 17.6% to 18.1% if the account is added. For these reasons, the account was considered a good prospect, even though on a stand-alone basis it was slightly below the hurdle rate.

25.2.2 SmartWriter optimization module

For a portfolio of large commercial accounts, the optimizer could locate the five accounts most in need of repricing, or the subset of the current portfolio that maximizes return. For a homeowners portfolio, the book of business is managed less on a home-by-home basis and more on a ZIP code, county, or state level; the optimizer can focus on which counties to expand market penetration into and in which ZIP codes to reduce premium volume.

Variables and objective

Define the following sets:

$\{1, 2, \ldots, N\}$ set of accounts in the portfolio,

$\{1, 2, \ldots, S\}$ set of loss scenarios.

Define the following input parameters:

p_i premium for account i,

e_i noncatstrophe expense for account i,

l_{is} loss (in dollars) for account i in scenario s,

π_s probability of scenario s,

ρ discount factor.

Define the following decision variables:

$$x_i \ (i = 1, \ldots, N) = \text{amount of account } i \text{ in the portfolio.}$$

The objective is to maximize expected ROC

$$\text{Max} \sum_{s=1,S} \sum_{i=1,N} \pi_s \frac{(x_i(p_i - e_i - l_{is}))}{\left[\rho F^{-1}(0.99) - \sum_{i=1,N} (x_i(p_i - e_i))\right]}, \tag{25.1}$$

where $F^{-1}(0.99)$ is calculated from the revised loss distribution $x_i * l_{is}$.

Correlations are implicitly captured in the analysis. Since the entire loss distribution is calculated for the objective function, the correlation among accounts will affect the return on capital.

Constraints

The following constraints can be added to the model. An account can be either in the portfolio or out of the portfolio, and so we add a binary constraint

$$x_i \in \{0, 1\}.$$

If one or more properties must be retained, we add

$$x_i = 1.$$

The total premium for the portfolio cannot be reduced past a specified level, MinPrem:

$$\sum_{i=1,N} (x_i * p_i) \geq \text{MinPrem}. \tag{25.2}$$

The expected income on the portfolio cannot be reduced past a specified level, MinInc:

$$\sum_{i=1,N} (x_i * (p_i - e_i - l_{is})) > \text{MinInc}. \tag{25.3}$$

25.2.3 Example Results

Below is the SmartWriter output for a California earthquake portfolio with 173 accounts [3]. The results are from real company data, but the numbers have been disguised to protect client confidentiality. The optimizer recommended the removal of 16 accounts from the portfolio. Table 25.3 shows summary information before and after the optimization for the portfolio as a whole.

Table 25.3. *Portfolio before and after optimization. Numbers are in* 000.

	Portfolio Today	Optimized Portfolio
Number of accounts	173	157
Premium	$5,600	$5,200
Expenses	$1,700	$1,600
Expected Cat Loss	$500	$300
Expected Income	$3,400	$3,300
Loss at $99^{th}\% = F^{-1}(0.99)$	$28,600	$12,900
Capital Required	$23,200	$8,800
Return on Capital: ROC	14.7%	37.5%

On the whole, this was a profitable book of business, but there were a small number of poorly performing accounts. Not only did these accounts have a poor expected return, but also they had a severe effect in the tail of the distribution. Expected income decreased by only $100,000 (3%), but the loss at the 99th percentile decreased by more than $15 million. ROC jumped from 14.7% to 37.5%. We have seen this with other books of business as well: a small percentage of accounts represent a large portion of the tail of the loss distribution.

Ideally, the portfolio manager should reprice these accounts on renewal instead of terminating them. Although market conditions will determine the extent to which this is feasible, SmartWriter provides output on all the accounts targeted by the optimizer. Table 25.4 contains information for one of these accounts.

For this example, the premium needed to meet the stand-alone ROC hurdle of 15% is $150,000, much greater than the current premium of $20,000. Repricing is most likely not an option for this account, but for examples where the current ROC is closer to the hurdle rate, repricing can be viable. (SmartWriter analysis is described further in [3].)

Table 25.4. *Account targeted for removal or repricing by optimizer.*

	Account A
Premium	$20
Expenses	$6
Expected Cat Loss	$12
Expected Profit	$2
Loss at $99^{tho}\% = F^{-1}(0.99)$	$780
Capital Required	$740
Return on Capital: ROC	**0.3%**
Ret. on Marginal: ROMAC	**0.4%**
Premium needed to meet 15% ROC hurdle	**$150**
Premium needed to meet 15% ROMAC hurdle	**$145**

25.3 Description of centralized optimization model

Managing a global company via centralized DFA, however, requires coordination between the divisions (groups) and headquarters; see Figure 25.4.

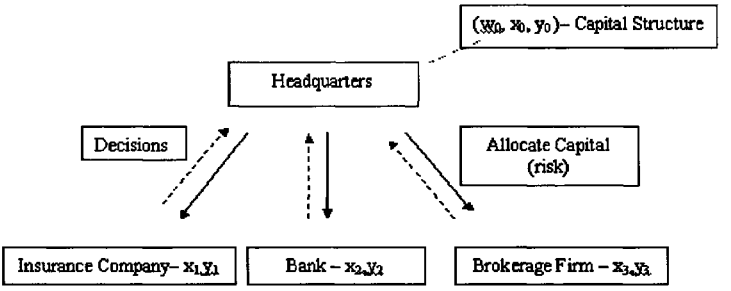


Figure 25.4. *Coordination among headquarters and divisions. Headquarters decides on the capital allocation and the asset-liability management of the company.*

To define the centralized DFA model, assume $K = \{1, \ldots, k\}$ divisions, a single time period, and multiple scenarios.

To define the full set of decisions (w, x, y) we introduce the following variables:

X asset,

Y liability related decisions,

W enterprise level decisions,

C firm economic capital (wealth),

P^s enterprise level profit/loss under scenario s.

The objective is to maximize the utility function

$$\text{Max } U[Z_1 \dots Zm], \tag{25.4}$$

where

$$\begin{aligned} Z_1 &= f_1(c, w, x, y), \\ Z_2 &= f_2(c, w, x, y), \\ Z_m &= f_m(c, w, x, y). \end{aligned}$$

$Z_1 \dots Z_m$ describe enterprise-wide statistics, e.g., expected surplus at the end of the planning period and/or probability of credit downgrade over the next 5 years. Other typical examples of objectives include the Tail VaR at the end of the first year or the planning period; see Figure 25.5.

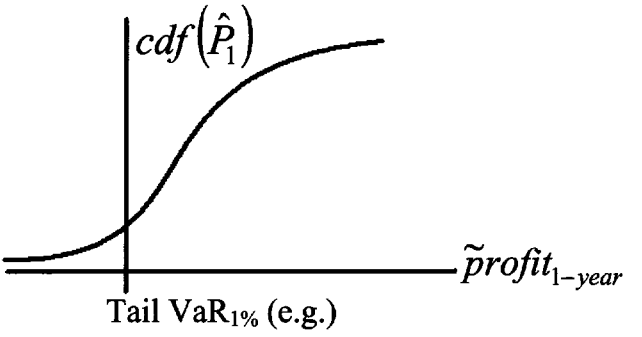


Figure 25.5. *Tail VaR as enterprise-wide statistics.*

The scheme in Figure 25.6 visualizes the centralized DFA model of a large-scale enterprise.

All major decisions are made within a single planning environment. In centralized DFA, the company is able to display its resources in a company-optimal fashion. The impact of any new activity or of any change in existing activities can be immediately evaluated with result to the enterprise. On the other hand, the centralized DFA is impractical for large-scale organizations because the division's optimal solution may be very different compared to the solution of the enterprise-wide optimization problem. To come to a compromise among the divisions and the headquarters we introduce the decentralized approach to the DFA.

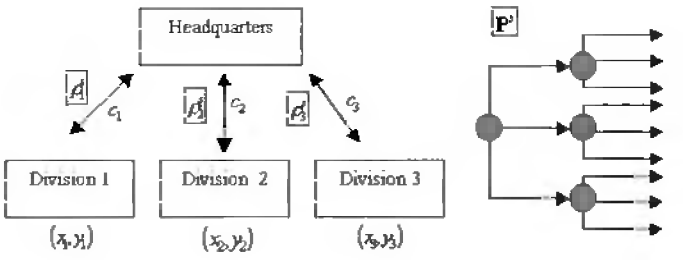


Figure 25.6. *Centralized DFA scheme.*

25.4 Formal decentralized optimization for DFA

The motivation behind the introduction of decentralized DFA is to design a practical system for analyzing and managing a large-scale financial organization. Many different scenarios are generated stochastically with the aim of giving information about the distribution of some important variables, like surplus or loss ratio. The steps towards a DFA model are

1. understanding risks within each division;
2. evaluating company-level risk;
3. enhancing enterprise value via multidimensional strategy:
   (a) reduce enterprise risks (reinsure),
   (b) modify capital structure,
   (c) direct divisions to change behavior;
4. optimizing total enterprise value (decentralized optimal DFA).

Let c_k be the capital allocated to the division k. With constraint $\sum c_k \leq C$, where C is the enterprise capital, the headquarters decides on the capital allocated to the division k and accordingly the division will be evaluated by means of a stochastic analysis s. The division has its own asset- and liability-related decisions. The simple approach to the decentralized DFA would be to estimate the expected profit/loss $E[\tilde{p}_k]$, std$[\tilde{p}_k]$ and correlation of $(\tilde{p}_k, P)$. The critical issues in decentralized DFA arise in evaluating the performance of each division. These issues can be adjusted with respect to the risk during the capital allocation or by means of risk adjusted return on capital (RAROC). Following the RAROC approach, the capital allocation should be based on the perceived risk of each division or on the quantile estimation of the profit/loss distribution associated to each division (VaR). Taking the cost of capital into account would minimize the shareholders' value and augment the riskiness of the division. A required return on the capital allocated above the riskless rate and risk-adjusted profit calculations would also implement the RAROC approach to the optimization. Froot and Stein [8] have developed a framework other than RAROC for analyzing the capital allocation and capital structure decisions facing financial institutions where they show how bank-level risk management considerations should factor into the pricing of the risks that cannot be easily hedged.

Historical estimates of mean and standard deviation may misrepresent future estimates. Moreover, the normal distribution may be ill suited to many types of uncertainties since insurance losses possess fat tails. This greatly underestimates the tail risk and hence the capital needs. And correlation may not be stable due to linkage with one or more underlying factors. Therefore we should project future scenarios based on a stochastic analysis and calculate the implied profit p_k^s for division k under scenario s and P^s enterprise profit under scenario s.

The algorithm in [6] was originally intended to be a computational technique for solving large linear programming problems that have a special structure. However, the steps of the Dantzig–Wolfe algorithm can be interpreted as an economic model for managing a distributed decision-making system; see [1]. The master decision maker must coordinate the solutions of the divisions to satisfy the corporate-wide constraint and maximize the entire objective function.

The decomposition in [5] is applied to problems with the following structure:

$$\text{Maximize } z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \tag{25.5}$$

subject to

$$\begin{gathered}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\
\vdots \\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \\
\text{- -} \\
e_{11}x_1 + e_{12}x_2 + \cdots + e_{1n}x_n = d_1 \\
\vdots \\
e_{q1}x_1 + e_{q2}x_2 + \cdots + e_{qn}x_n = d_q, \\
x_j \geq 0 \ (j = 1, 2, \ldots, n).
\end{gathered}$$

The first and the second sets of constraints are the complicating resource constraints and the subproblem constraints, respectively.

The constraints are divided into two groups. Usually the problem is much easier to solve if the complicating a_{ij} constraints are omitted, leaving only the easy subproblem constraints.

Consider any subproblem solution-proposal. Given $x_1, \ldots, x_n$ (a feasible solution to the subproblem constraints), we may compute the amount of resource r_i used in the ith complicating constraint and the profit p associated with the proposal

$$\begin{aligned}
r_i &= a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \quad (i = 1, \ldots, m), \\
p &= c_1x_1 + c_2x_2 + \cdots + c_nx_n.
\end{aligned}$$

When k proposals to the subproblem are known, the procedure acts to weight these proposals optimally. Decomposition in this context extends the interpretation to decentral-

ized decision making. It provides the mechanism by which prices can be used to coordinate the activities of several decision makers. We interpret the problem as utility maximization for a firm with two divisions. There are two levels of decision making—headquarters and division. Subsystem constraints reflect the divisions' allocation of their own resources that are not shared. The complicating constraints limit corporate resources, which are shared and used in any proposal from either division. The main disadvantage of centralized decision making by optimizing the firm as a single entity arises because of the expense of gathering detailed information about the divisions in a form usable by either corporate headquarters or other divisions. It is often best for each division and corporate headquarters to operate somewhat in isolation, having privacy and responsibility as much as possible. In the decentralized approach where the decomposition algorithms can be applied the information passed on is the *state-prices*, from headquarters to the divisions, and *proposals*, from the divisions to the corporate coordinator. Only the headquarters is aware of the full corporate constraints, and each division knows its own operating constraints. In decentralized decision making, the headquarters weights the subproblem proposals to maximize the overall utility. From the solution the state prices are set on the constraints and/or resources. The state price coordination among the two levels of decision making enables the headquarters to have an internal evaluation of the resources and to charge a cost of resource and/or capital to the divisions.

State-price coordination is summarized in Figure 25.7.

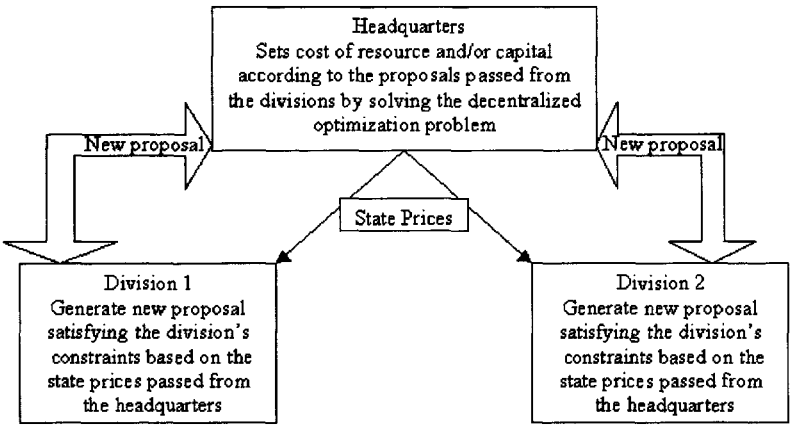


Figure 25.7. *State-price coordination among the headquarters and the divisions.*

25.5 Running the two-division example

We interpret the problem as a utility maximization of the certainty equivalent for an insurance firm with two divisions. There are 216 accounts to invest and 50,000 scenarios for uncertain losses associated with each account. The loss data for each account are generated by SmartWriter. There are two levels of decision making—headquarters and division. The centralized decision-making model allows only the headquarters to decide on the accounts

to be invested and on the leverage. This large-scale convex programming problem is solved and analyzed numerically by using the software LOQO [13]. The whole firm is modeled as a single entity as follows.

25.5.1 The centralized model

Indices

acc$\{1, 2, \ldots, N\}$ set of accounts in the portfolio, where N is the total number of accounts the firm can invest in (216);

scen$\{1, 2, \ldots, S\}$ set of loss scenarios, where S is the total number of scenarios with respect to the losses (50,000).

Parameters

$l_{[a][s]}$ loss matrix includes all the loss data regarding each account,

$p_{[a]} =$ Risk Premium$*60\%$ revenue generated from the investment in each account a in acc ($60,000),

c initial_starting_capital ($200,000),

r interest rate,

b borrowing rate,

f limit the amount borrowed by $f*$ initial_capital,

k asset to capital ratio,

π_s probability of scenario s,

f_1 and f_2 coefficients of the exponential part of the utility function.

Variables

x initial asset,

w amount leveraged,

$y_{[a]}$ fraction invested in account a,

$z_{[s]}$ capital at the end of the investment horizon in each scenario.

Utility maximization of CE (ending capital) forms the objective function in the convex programming formulation of the centralized model (25.6). Definitional constraints set up the initial capital and the CE in terms of borrowings and investments.

$$\text{Maximize} \sum_s \pi_s * [z_s - f_1 * \exp(-f_2 * z_s)]$$

subject to

$$x - w - \sum_a p_a * y_a = c,$$

$$z_s = (1+r) * x - (1+b) * w - \sum_a l_a^s * y_a \quad \forall s \in \text{scen}, \tag{25.6}$$

$$x \leq k * c,$$

$$0 \leq w \leq f * c,$$

$$0 \leq y_a \leq 1 \quad \forall a \in \text{acc}.$$

The objective function in (25.6) maximizes the utility from the certainty equivalent of the capital at the end of the investment horizon with respect to the losses in each scenario. The first constraint equates the initial asset to the sum of the amount borrowed and the return of the investments at the beginning of the time period. The second batch of constraints imply that the ending capital is calculated by subtracting the amount borrowed with the interest on it and the losses with respect to each scenario from the initial asset together with the return on it. We also have the constraint setting the asset-capital ratio to k. Other than that we limit the amount to borrow by a factor f times the initial capital we start with. The last constraint sets the bounds to the fraction we can invest in each account.

However, in the decentralized optimization, the initial capital of the firm will be allocated to each of the divisions and the capital allocation being a variable itself allows us to define the initial assets as two distinct variables for each division.

Moreover, the divisions will invest into the accounts mutually exclusively. Each division can invest only in a certain number of accounts. It is entirely the division's decision how much to invest in which account and how much to borrow. In that manner each division can decide its own strategies, and the capital allocation should not exceed the initial capital of the firm.

25.5.2 The decentralized model

Indices

- $\text{acc}_{[1]}\{1, 2, \ldots, A_1\}$, where A_1 is the total number of accounts the first division can invest in (216/2);
- $\text{acc}_{[2]}\{1, 2, \ldots, A_2\}$, where A_2 is the total number of accounts the second division can invest in (216/2);
- $\text{scen}\{1, 2, \ldots, S\}$, where S is the total number of scenarios with respect to the losses (50,000).

Parameters

$l_{[a][s]}$ loss matrix includes all the loss data regarding each account $a \in (\text{acc}_1 \cup \text{acc}_2)$ and $s \in \text{scen}$,

$p_{[a]}$ = Risk Premium $*$ 60% revenue generated from the investment in each account $a \in (\text{acc}_1 \cup \text{acc}_2)$ ($60,000),

c initial_starting_capital ($200,000),

r interest rate,

b borrowing rate,

f limit the amount borrowed by $f * c$,

k asset to capital ratio,

f_1 and f_2 coefficients of the exponential part of the utility function,

π_s probability of scenario s.

Variables

$x_{[1]}$ initial asset of the first division,

$x_{[2]}$ initial asset of the second division,

$c_{[1]}$ amount of initial capital allocated to the first division,

$c_{[2]}$ amount of initial capital allocated to the second division,

$w_{[1]}$ amount leveraged in the first division,

$w_{[2]}$ amount leveraged in the second division,

$y^1_{a_1}$ fraction invested in account $a_1 \in \text{acc}_1$,

$y^2_{a_2}$ fraction invested in account $a_2 \in \text{acc}_2$,

z^1_s first division's capital at the end of the investment horizon in each scenario,

z^2_s second division's capital at the end of the investment horizon in each scenario.

Model

In (25.7), in the objective function is the maximization of the utility of the certainty equivalent. Different from the centralized model, we define seperate variables for the initial asset, amount borrowed, ending capital with respect to the scenarios, and fractions invested in each account for each division. The first couple of constraints define the initial assets in terms of amount borrowed and revenues from accounts according to the fractions invested. The second couple of constraints calculate the ending capital by taking the account losses with respect to the scenarios into account. The amount borrowed cannot exceed k multiples of the capital allocated to that division. The capital allocation should adapt to the initial capital the company starts with. We have the upper and lower bounds for the amount borrowed in total and the fractions invested in each account. In the convex programming formulation

of the decentralized model each division decides on its own asset-liability management by watching out for the complicating resource constraints. The capital allocation is one of the major outputs of the decentralized model.

$$\text{Maximize} \sum_s \pi_s * [(z_s^1 + z_s^2) - f_1 * \exp(-f_2 * (z_s^1 + z_s^2))]$$

subject to

$$\begin{aligned}
&x_1 = w_1 + c_1 + \sum_{a_1 \in \text{Acc}_1} p_{a_1} * y_{a_1}^1, \\
&x_2 = w_2 + c_2 + \sum_{a_2 \in \text{Acc}_2} p_{a_2} * y_{a_2}^2, \\
&z_s^1 = (1+r) * x_1 - (1+b) * w_1 - \sum_{a_1 \in \text{Acc}_1} l_{a_1}^s * y_{a_1}^1 \quad \forall s \in \text{scen}, \\
&z_s^2 = (1+r) * x_2 - (1+b) * w_2 - \sum_{a_2 \in \text{Acc}_2} l_{a_2}^s * y_{a_2}^2 \quad \forall s \in \text{scen}, \qquad (25.7) \\
&x_1 \le k * c_1, \\
&x_2 \le k * c_2, \\
&c_1 + c_2 = c, \\
&0 \le w_1 + w_2 \le f * c, \\
&0 \le y_{a_1}^1 \le 1 \quad \forall a_1 \in \text{acc}_1, \\
&0 \le y_{a_2}^2 \le 1 \quad \forall a_2 \in \text{acc}_2, \\
&0 \le w_1, \\
&0 \le w_2.
\end{aligned}$$

25.5.3 Convergence

The decentralized optimization model should give an optimal solution similar to that of the centralized optimization model. Moreover, it has to answer all the questions, such as the asset-liability decisions separately for both of the divisions and the optimal capital allocation to the divisions. Let us prove that the decentralized model produces the same optimal result as the centralized version and answers all these strategical questions.

Theorem 25.1. *If the decentralized model has the optimal solution*

$$\hat{x}_1, \hat{x}_2, \hat{z}_s^1, \hat{z}_s^2, \hat{w}_1, \hat{w}_2, \hat{c}_1, \hat{c}_2, \hat{y}_a^1, \hat{y}_a^2, \qquad (25.8)$$

where

$$a \in \text{acc}(= \text{acc}_1 \cup \text{acc}_2) \qquad (25.9)$$

and

$$\begin{aligned}
&\hat{y}_a^1 = 0 \quad \text{if } a \in \text{acc}_2, \\
&\hat{y}_a^2 = 0 \quad \text{if } a \in \text{acc}_1,
\end{aligned} \qquad (25.10)$$

then

$$\begin{aligned} \hat{x} &= \hat{x}_1 + \hat{x}_2, \\ \hat{w} &= \hat{w}_1 + \hat{w}_2, \\ \hat{z}_s &= \hat{z}_s^1 + \hat{z}_s^2, \\ \hat{y}_a &= \hat{y}_a^1 + \hat{y}_a^2. \end{aligned} \tag{25.11}$$

Proof. Assume that the decentralized model has the optimal solution

$$\hat{x}_1, \hat{x}_2, \hat{z}_s^1, \hat{z}_s^2, \hat{w}_1, \hat{w}_2, \hat{c}_1, \hat{c}_2, \hat{y}_a^1, \hat{y}_a^2, \tag{25.12}$$

where we set

$$a \in \text{acc}(= \text{acc}_1 \cup \text{acc}_2) \tag{25.13}$$

and

$$\begin{aligned} \hat{y}_a^1 &= 0 \quad \text{if } a \in \text{acc}_2, \\ \hat{y}_a^2 &= 0 \quad \text{if } a \in \text{acc}_1. \end{aligned} \tag{25.14}$$

Since $\hat{x}_1, \hat{x}_2, \hat{z}_s^1, \hat{z}_s^2, \hat{w}_1, \hat{w}_2, \hat{c}_1, \hat{c}_2, \hat{y}_a^1, \hat{y}_a^2$ is optimum, it is also a feasible solution for the decentralized model. Namely,

$$\begin{aligned} \hat{x}_1 - \hat{w}_1 - \hat{c}_1 - \sum_{a \in \text{acc}} p_a * \hat{y}_a^1 &= 0, \\ \hat{x}_2 - \hat{w}_2 - \hat{c}_2 - \sum_{a \in \text{acc}} p_a * \hat{y}_a^2 &= 0, \\ \hat{z}_s^1 - (1+r) * \hat{x}_1 - (1+b) * \hat{w}_1 - \sum_{a \in \text{acc}} l_a^s * \hat{y}_a^1 &= 0 \quad \forall s \in \text{scen}, \\ \hat{z}_s^2 - (1+r) * \hat{x}_2 - (1+b) * \hat{w}_2 - \sum_{a \in \text{acc}} l_a^s * \hat{y}_a^2 &= 0 \quad \forall s \in \text{scen}, \\ \hat{x}_1 &\le k * \hat{c}_1, \\ \hat{x}_2 &\le k * \hat{c}_2, \\ \hat{c}_1 + \hat{c}_2 &= c, \\ 0 \le \hat{w}_1 + \hat{w}_2 &\le f * c, \\ 0 \le \hat{y}_a^1 &\le 1 \quad \forall a \in \text{acc}, \\ 0 \le \hat{y}_a^2 &\le 1 \quad \forall a \in \text{acc}. \end{aligned} \tag{25.15}$$

Define new variables from these variables of the decentralized model's optimum solution:

$$\begin{aligned} \hat{x} &= \hat{x}_1 + \hat{x}_2, \\ \hat{w} &= \hat{w}_1 + \hat{w}_2, \\ \hat{z}_s &= \hat{z}_s^1 + \hat{z}_s^2, \\ \hat{y}_a &= \hat{y}_a^1 + \hat{y}_a^2. \end{aligned} \tag{25.16}$$

Using these new defined variables, we reformulate the same equations as

$$\left.\begin{aligned}\hat{x}_1 - \hat{w}_1 - \hat{c}_1 - \sum_{a\in\text{acc}} p_a * \hat{y}_a^1 = 0,\\ \hat{x}_2 - \hat{w}_2 - \hat{c}_2 - \sum_{a\in\text{acc}} p_a * \hat{y}_a^2 = 0,\end{aligned}\right\}$$

$$(\hat{x}_1 + \hat{x}_2) - (\hat{w}_1 + \hat{w}_2) - (\hat{c}_1 + \hat{c}_2) - \sum_{a\in\text{acc}} p_a * (\hat{y}_a^1 + \hat{y}_a^2) = 0,$$

$$\left.\begin{aligned}\hat{z}_s^1 - (1+r) * \hat{x}_1 - (1+b) * \hat{w}_1 - \sum_{a\in\text{acc}} l_a^s * \hat{y}_a^1 = 0,\\ \hat{z}_s^2 - (1+r) * \hat{x}_2 - (1+b) * \hat{w}_2 - \sum_{a\in\text{acc}} l_a^s * \hat{y}_a^2 = 0,\end{aligned}\right\}$$

$$\begin{aligned}(\hat{z}_s^1 + \hat{z}_s^2) - (1+r) * (\hat{x}_1 + \hat{x}_2) - (1+b) * (\hat{w}_1 + \hat{w}_2)\\ - \sum_{a\in\text{acc}} l_a^s * (\hat{y}_a^1 + \hat{y}_a^2) = 0,\end{aligned} \tag{25.17}$$

$$\left.\begin{aligned}\hat{x}_1 \le k * \hat{c}_1\\ \hat{x}_2 \le k * \hat{c}_2\end{aligned}\right\} (\hat{x}_1 + \hat{x}_2) \le k * (\hat{c}_1 + \hat{c}_2),$$

$$\hat{c}_1 + \hat{c}_2 = c,$$

$$0 \le \hat{w}_1 + \hat{w}_2 \le f * c,$$

$$\left.\begin{aligned}0 \le \hat{y}_a^1 \le 1\\ 0 \le \hat{y}_a^2 \le 1\end{aligned}\right\} 0 \le (\hat{y}_a^1 + \hat{y}_a^2) \le 1,$$

since $\hat{y}_a^1 * \hat{y}_a^2$ is always zero because of the divisions' mutually exclusive investments.

Let's express these aggregate equations in terms of our new variables:

$$\begin{aligned}\hat{x} - \hat{w} - c - \sum_a p_a * \hat{y}_a = 0,\\ \hat{z}_s - (1+r) * \hat{x} - (1+b) * \hat{w} - \sum_{a\in\text{acc}} l_a^s * \hat{y}_a = 0 \quad \forall s \in \text{scen},\\ \hat{x} \le k * c,\\ 0 \le \hat{w} \le f * c,\\ 0 \le \hat{y}_a \le 1 \quad \forall a \in \text{acc}.\end{aligned} \tag{25.18}$$

This implies that the optimum solution for the decentralized model is feasible for the centralized model after the formulation as above.

Now we have to analyze the optimality of the new variables for the centralized version. Since $\hat{x}_1, \hat{x}_2, \hat{z}_s^1, \hat{z}_s^2, \hat{w}_1, \hat{w}_2, \hat{c}_1, \hat{c}_2, \hat{y}_a^1, \hat{y}_a^2$ is optimal, the following inequality holds:

$$\begin{aligned}\sum_s \pi_s * [(\hat{z}_s^1 + \hat{z}_s^2) - f_1 * \exp(-f_2 * (\hat{z}_s^1 + \hat{z}_s^2))]\\ \ge \sum_s \pi_s * [(z_s^1 + z_s^2) - f_1 * \exp(-f_2 * (z_s^1 + z_s^2))] \quad \forall z_s^1, z_s^2.\end{aligned} \tag{25.19}$$

This inequality above also implies

$$\sum_s \pi_s * [\hat{z}_s - f_1 * \exp(-f_2 * \hat{z}_s)] \geq \sum_s \pi_s * [z_s - f_1 * \exp(-f_2 * z_s)] \quad \forall z_s. \tag{25.20}$$

Therefore, the new defined variables in (25.16) form the feasible and optimal solution for the centralized model.

25.5.4 Numerical experiments

Addition of accounts

To see the effects of adding additional accounts into the portfolio, we solve the centralized model for different account numbers and compare the utility level and the wealth values (where we fix $f_1 = 1$ and $f_2 = 0.001$); see Table 25.5.

Table 25.5. *The effects of additional accounts. The marginal increase in the utility and wealth level is not that high after the number of accounts exceeds eight. After that level the newly added accounts start to substitute for some of the former ones.*

Number of Accounts	a2	a3	a4	a5	a6	a7	a8	a9	a10	a15
Utility Level	29.52	35.88	42.25	48.61	54.97	60.36	62.64	62.64	62.64	62.66
Percent Increase in Utility	0.216	0.177	0.150	0.131	0.096	0.038	0	0	0	0
Wealth (in $0000)	30.4866	36.8461	43.2061	49.5571	55.9170	61.2973	63.5820	63.5820	63.5820	63.5992
Percent Increase in Wealth	0.209	0.173	0.147	0.128	0.096	0.037	0	0	0	0
Borrowing	5.7906	5.8138	3.6200	3.2142	1.4540	0	0	0	0	0

The increase in the utility level is remarkable, especially until the number of accounts available in the portfolio increases to eight; see Figures 25.8 and 25.9. The reason is when we run the model for eight accounts, the profitability and the dominance of the accounts come into play. For instance, the eighth account dominates the first account because of its loss structure. That is why the marginal increase in the utility and wealth level is not that high after the number of accounts exceeds eight. After that level the newly added accounts start to substitute for some of the former ones (see Figure 25.6).

The addition of new accounts also affects the initial asset values of the company and its decomposition; see Figure 25.10.

We have almost the same pattern of marginal increase in the initial assets as in the utility and wealth level. However ,when we compare the decomposition of initial assets for a different number of accounts, it is much more clear that there are some accounts dominating the others. Moreover, as the variety of accounts gets wider, the amount borrowed decreases. When we compare the initial asset components with 10 accounts and 15 accounts, the fact is that accounts 12, 14, and 15 are dominating accounts 2, 3, and 5; see Figures 25.11 and 25.12.

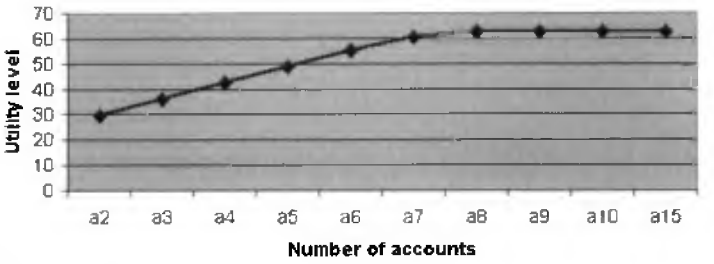


Figure 25.8. *How the addition of accounts affects the utility maximization* (1000 *scenarios;* $f1 = 1$, $f2 = 0.001$).

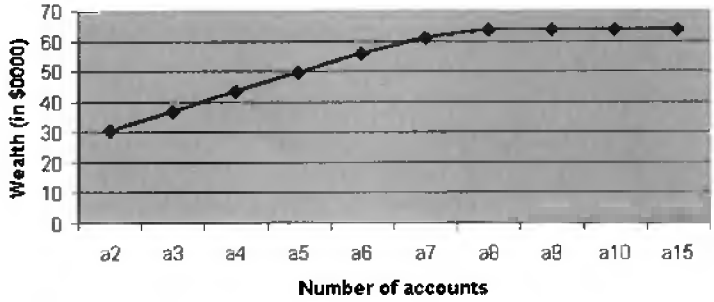


Figure 25.9. *Wealth versus number of accounts.*

Table 25.6. *Fraction invested in different accounts. After eight accounts are available, the latter accounts start to substitute for the former ones because of their more desired loss structure. At first, the first account is eliminated and the fraction invested in the second is decreased to* 67%.

Number of Accounts	a2	a3	a4	a5	a6	a7	a8	a9	a10	a15
Fraction invested in the 1st account	1	1	1	1	1	0.67	0	0	0	0
Fraction invested in the 2nd account	1	1	1	1	1	1	0.67	0.67	0.67	0
Fraction invested in the 3rd account		1	1	1	1	1	1	1	1	0.007
Fraction invested in the 4th account			1	1	1	1	1	1	1	1
Fraction invested in the 5th account				1	1	1	1	1	1	0
Fraction invested in the 6th account					1	1	1	1	1	1
Fraction invested in the 7th account						1	1	1	1	1
Fraction invested in the 8th account							1	1	1	1
Fraction invested in the 9th account								0	0	0
Fraction invested in the 10th account									0	0
Fraction invested in the 11th account										0
Fraction invested in the 12th account										1
Fraction invested in the 13th account										0
Fraction invested in the 14th account										1
Fraction invested in the 15th account										0.66

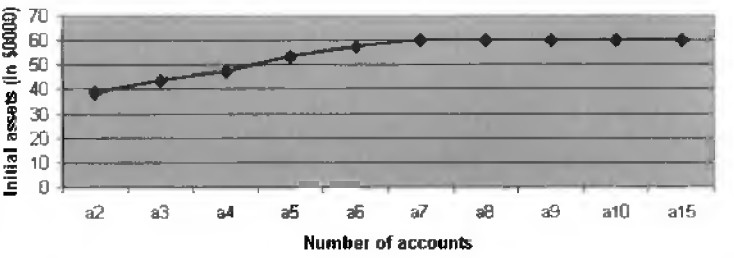


Figure 25.10. *Initial assets versus number of accounts.*

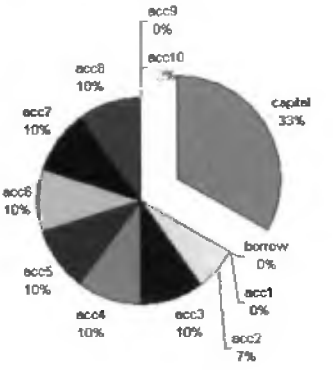


Figure 25.11. *Decomposition of initial assets with ten accounts.*

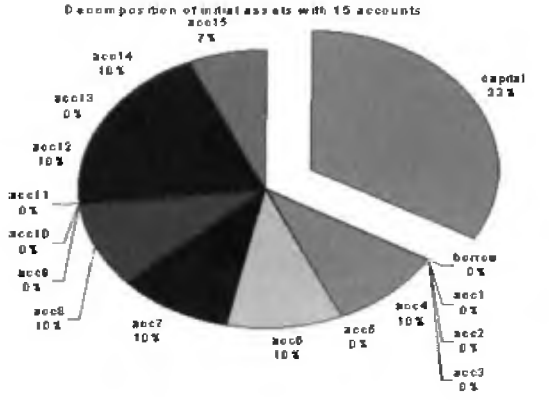


Figure 25.12. *Decomposition of initial assets for different numbers of available accounts. Accounts* 12, 14, *and* 15 *dominate accounts* 2, 3, *and* 5.

The VaR and the T-VaR values of the centralized model decrease with an increasing number of accounts; see Figure 25.13. The more accounts the company has to invest in, the better the company can hedge its risk against unprofitable scenarios. However, the decrease in the VaR values is not proportional to the increase in the number of accounts.

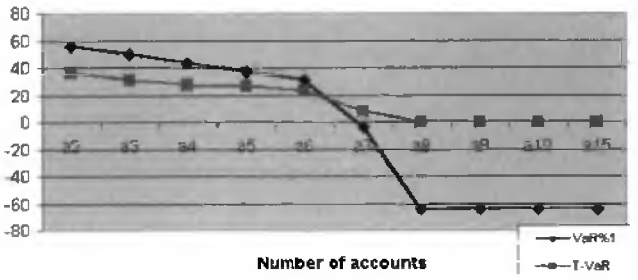


Figure 25.13. *VaR %1 and T-VaR for different numbers of accounts. The VaR and T-VaR decrease with increasing number of accounts.*

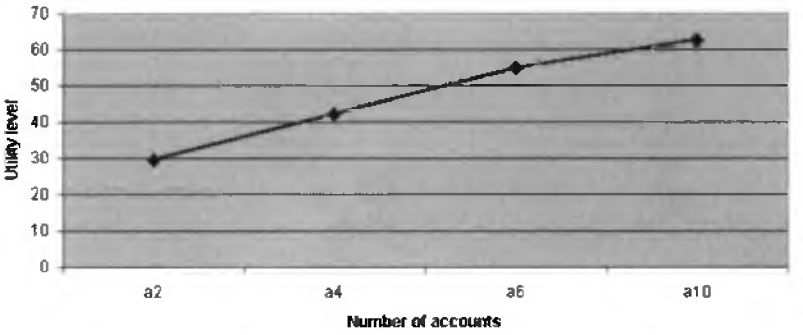


Figure 25.14. *Utility level versus number of accounts: decentralized optimization model.*

This is because some of the newly added accounts have a better probabilistic loss structure in the sense that they dominate some of the older accounts. Whenever this happens, the company invests in the new account by giving up some of the former accounts. This is especially the case when we increase the number of accounts from six to eight. There are no large fluctuations in the VaR values when we increase the number of accounts from 9 to 14 because during this increase the fractions by which the company invests into the accounts do not change radically. The decomposition of the portfolio remains very stable.

Now we focus on the decentralized model and analyze the effects of adding new accounts to the portfolio on the utility level, wealth, and risk statistics. First we analyze the changes in the utility level and wealth with respect to the account size; see Figures 25.14 and 25.15.

The wealth structure of the first and the second divisions is changing such that the total wealth of the company is increasing with a decreasing slope as more accounts are added to the portfolio. Meanwhile, borrowing becomes less attractive for the divisions as the variety of available accounts with different loss structures increases; see Figure 25.16.

The decomposition of the aggregated initial assets for the decentralized model with ten accounts looks exactly the same as in the centralized version (see Figure 25.17). This fact again assures that the decentralized optimization model is going to produce the same optimal results as the centralized model and, moreover, is going to provide to the company

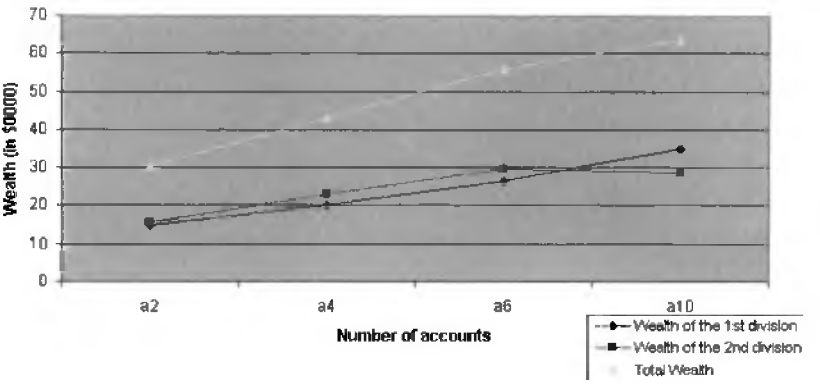


Figure 25.15. *Wealth versus number of accounts. The wealth structure of both divisions is changing so that the firm's wealth is increasing.*

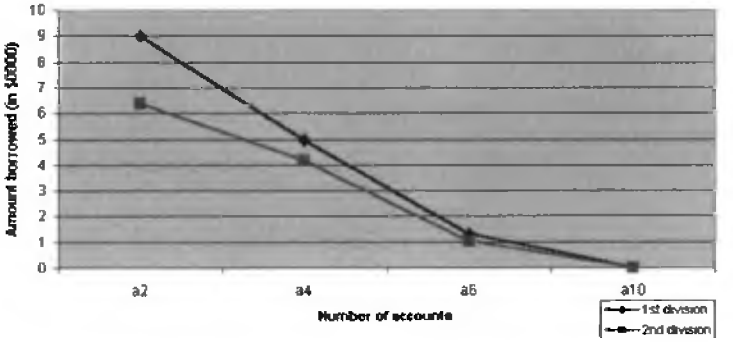


Figure 25.16. *Amount borrowed by the divisions. Borrowing becomes less attractive as the variety of available accounts increases.*

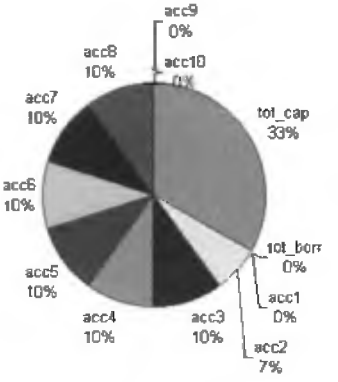


Figure 25.17. *Decomposition of total initial assets with ten accounts—same as the decomposition for the centralized model.*

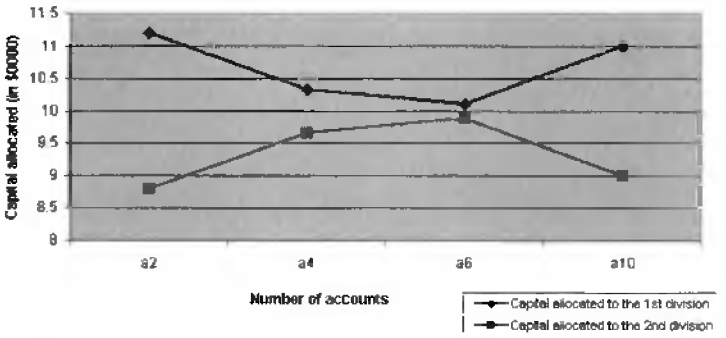


Figure 25.18. *Capital allocation to the divisions.*

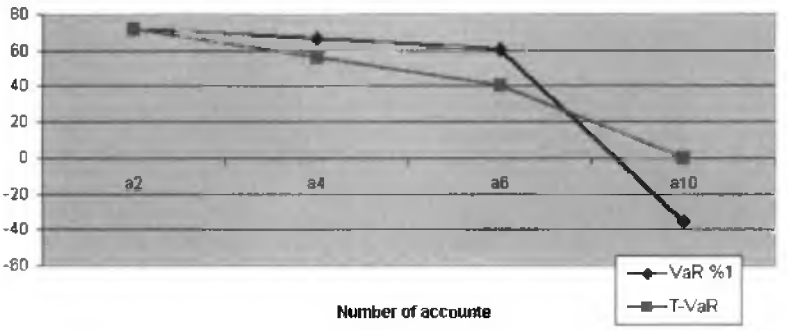


Figure 25.19. *VaR 1% and T-VaR values for the first division. The tail statistics are decreasing as more accounts are added.*

much more information, such as the optimum capital allocation, the mutually exclusive investment strategies of both divisions, and the optimum amount the divisions should borrow (see Figure 25.18).

The tail statistics of both the VaR and T-VaR values decrease as the number of accounts increases (see Figures 25.19 and 25.20). This pattern is similar to the VaR behavior at the centralized version.

Changing the parameters f_1 and f_2

We have also analyzed the decentralized model with respect to different f_2 values. We run the model first with $f_1 = 1$ and $f_2 = 0.001, 0.005, \ldots, 5, 30, 100$. As f_2 increases, we have a more risk-neutral utility function, which becomes closer to the linear function as f_2 gets larger. The pattern in Figures 25.21 and 25.22 is easy to interpret. Both divisions try to operate at such an optimum level that the aggregate company becomes increasingly profitable. The increase in VaR of any division is compensated by the decrease in VaR of the other division. In total, the VaR and T-VaR statistics for the whole company never increase as f_2 gets bigger.

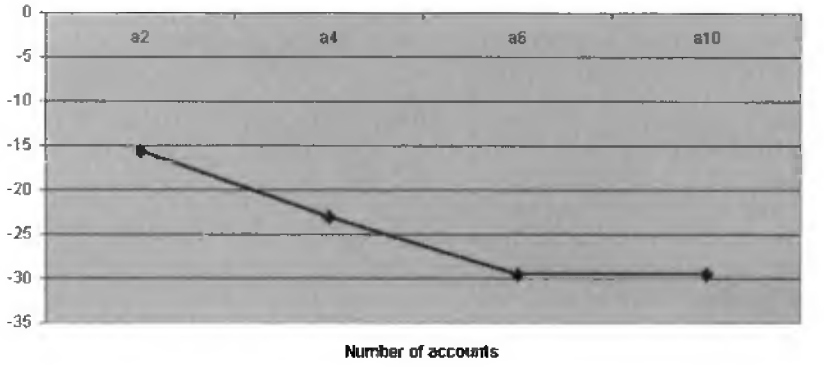


Figure 25.20. *VaR 1% values for the second division.*

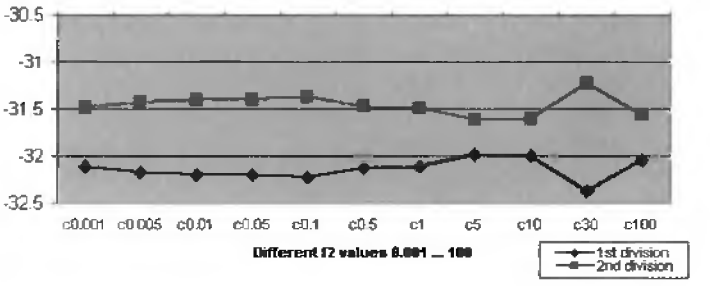


Figure 25.21. *VaR 1% values for the decentralized model with $f_1 = 1$. The increase in VaR of any division is compensated by the decrease in VaR of the other division.*

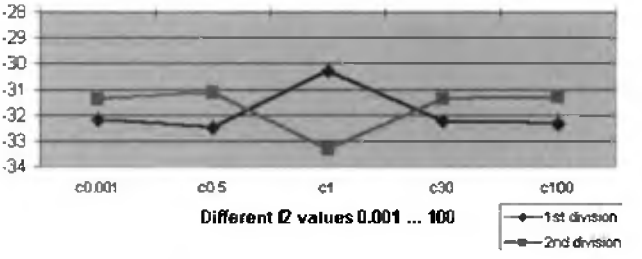


Figure 25.22. *VaR 1% values for the decentralized model with $f_1 = 30$. Both divisions operate at such an optimum level that the whole firm becomes increasingly profitable.*

When we run the decentralized model for different f_2 values by fixing $f_1 = 30$, we have almost the same type of trend in the VaR statistics. However, one difference is that at $f_1 = 30$ the pattern is just shifted leftward since at a higher level of f_1, even small changes of the f_2 parameter have a much faster effect on the statistics of both divisions.

Computational statistics

The centralized optimization model with 1000 scenarios, 216 accounts, $k = 3$, and $f_1 = f_2 = 1$ has the following time statistics:

```
Variables: nonneg    0, free   1001, bdd    217, total 1219
Constraints: eq    5002, ineq      0, ranged      0, total 1002
Nonzeros:    A   220219, Q         1
Total time in seconds: 436.92
CPU time in program: 478.0(seconds)
CPU time in system: 1.0(seconds)
Total CPU time: 8:00
Percentage of CPU time in program: 99%
```

The decentralized optimization model with 1000 scenarios, 216 accounts, $k = 3$, and $f_1 = f_2 = 1$ has the following time statistics:

```
Variables: nonneg 436, free   2005, bdd     0, total 2441
Constraints: eq    2006, ineq      0, ranged   217, total 2223
Nonzeros:    A   440881, Q         1
Total time in seconds: 728.47
CPU time in program: 783.0(seconds)
CPU time in system: 1.0(seconds)
Total CPU time: 13:06
Percentage of CPU time in program: 99%
```

The decentralized optimization model with 5000 scenarios, 216 accounts, $k = 5$, and $f_1 = f_2 = 1$ has the following time statistics:

```
Variables: nonneg 436, free  10005, bdd    0, total   10441
Constraints: eq    10006, ineq     0, ranged  217, total   10223
Nonzeros:    A   2200881, Q         1
CPU time in program: 7115.0(seconds)
CPU time in system: 2.0(seconds)
Total CPU time: 2:00:57
Percentage of CPU time in program: 98%
```

25.6 Conclusions and future work

As seen in the numerical experimentations and the theoretical convergence of the two models, we can conclude that the decentralized version enables the divisions to have much more flexibility and independence than the centralized version does. Moreover, in an aggregate sense the company has no trade-offs regarding the utility maximization.

The continuing convergence of the traditional capital and insurance markets should yield innovative approaches to managing emerging risks. Shareholders are increasingly holding boards of directors and senior executives to higher accountability standards. Aimed at giving shareholders more information and control and increasing the duty of care of directors, the Kon TraG bill was introduced into law in Germany in 1998. Growing corporate principles have been implemented in other countries, such as France (Marini Report, Levy–Lang Committee), Italy (Draghi Commission), Holland, and Canada. The Kon TraG bill

[11] emphasizes the necessity of an early warning system in an enterprise. According to the law, enterprises must have a risk management system to implement. The public limited companies are obliged to set up an enterprise risk management system and in the frame of the end-of-year inspection, the risk management system is to be judged by the chartered accountants.

We will focus next on extending this decentralized model to an enterprise risk management system that incorporates the multiperiod planning strategies and project hurdle rates and cost of capital issues.

Acknowledgment

This research was funded in part by NSF grant DMI-0323410.

Bibliography

[1] W. J. Baumol and T. Fabian, *Decomposition, pricing for decentralization and external economies*, Management Sci., 11 (1964), pp. 1–32.

[2] D. Bell, *Risk, return, and utility*, Management Sci., 41 (1995), pp. 23–30.

[3] A. J. Berger, J. M. Mulvey, and K. Nish, *A portfolio management system for catastrophe property liabilities*, Casualty Actuarial Society Forum, 1998, pp. 1–14.

[4] G. C. E. Boender, *A hybrid simulation/optimization scenario model for asset liability management*, Eur. J. Oper. Res., 99 (1997), pp. 126–135.

[5] S. P. Bradley, A. C. Hax, and T. L. Magnanti, *Applied Mathematical Programming*, Addison-Wesley, Reading, MA, 1977.

[6] G. B. Dantzig and P. Wolfe, *The decomposition algorithm for linear programs*, Econometrica, 29 (1961), pp. 767–778.

[7] M. A. H. Dempster, *The development of the Midas debt management system*, in Worldwide Asset and Liability Modeling, W. T. Ziemba and J. M. Mulvey, eds., Cambridge University Press, Cambridge, UK, 1998.

[8] K. A. Froot and J. C. Stein, *Risk management, capital budgeting, and capital structure policy for financial institutions: An integrated approach*, J. Financial Econ., 47 (1998), pp. 55–82.

[9] R. Kouwenberg, *Scenario generation and stochastic programming models for asset liability management*, Eur. J. Oper. Res., 134 (2001), pp. 279–292.

[10] J. M. Mulvey, B. Pauling, S. Britt, and F. Morin, *Dynamic financial analysis for multinational insurance companies*, in Handbook on Finance, W. T. Ziemba and S. Zenios, eds., North-Holland, Amsterdam, 2005, to appear.

[11] F. Romeike, *KonTraG: Gesetzlich verordnetes Risk management?*, RiskNews, 7 (2000), pp. 2–6.

[12] R. RUSH, J. M. MULVEY, J. MITCHELL, AND T. WILLEMAIN, *Stratified filtered sampling in stochastic optimization*, J. Appl. Math. Decision Sci., 4 (2000), pp. 17–38.

[13] R. B. VANDERBEI, *Optimization and Applications*, www.orfe.princeton.edu/~loqo.

[14] S. WALLACE, *Generating scenario trees for multistage problems*, Management Sci., 1 (2001), pp. 295–307.

Chapter 26

Wealth Goals Investing

Leonard C. MacLean, Yonggan Zhao,† and William T. Ziemba‡*

26.1 Introduction

In the standard portfolio selection problem, an investor determines an amount of capital to invest in riskless and risky investment opportunities at each point in time. The returns on the risky assets are dynamic random variables. If the distributions for future returns are known, then the accumulated capital at future times can be characterized for a particular investment strategy. For special models of asset return dynamics, such as geometric Brownian motion, the characterization of accumulated capital is analytic. In any case the path of capital can be simulated and the distribution of wealth can be described numerically. The objective in deciding on an investment strategy is to control the distribution of accumulated capital in the future. If preferences for wealth can be defined by a utility function, then investment strategies can be ordered by expected utility and a preferred strategy determined [13]. A component of preference in financial markets is aversion to risk, which requires that the utility function be concave in wealth. The degree of concavity has been used to define an index of risk aversion [11] and the premium placed on risk [1]. Typically the risk premium is a decreasing function of current wealth, and an important class of utilities exhibiting that property are *constant relative risk aversion (CRRA) functions*. Many studies of investment risk use this utility because it yields explicit solutions [10]. One property of the solution is that the optimal share of wealth invested in the various assets is independent of current wealth. This does not offer protection against financial collapse when unfavorable returns result. The need to protect against collapse has focused attention on downside risk measures

*School of Business Administration, Dalhousie University, Halifax, NS, B3H 3J5 Canada (lmaclean@mgmt.dal.ca).

†Nanyang Business School, Nanyang Technological University, 639798, Singapore (aygzhao@ntu.edu.sg).

‡Sauder School of Business, University of British Columbia, Vancouver, BC, V6T 1Z2 Canada (ziemba@interchange.ubc.ca), and Sloan School of Management, MIT, Cambridge, MA 02142.

[3]. MacLean and Ziemba [7] and MacLean, Ziemba, and Blazenko [8] use risk measures in identifying investment strategies which achieve capital growth with a required level of security, where security is defined as controlling downside risk. The standard risk measure is value at risk (VaR), which describes the worst loss that can happen under market conditions at a given confidence level [5]. Although VaR was developed as a descriptive measure, it is also used as a constraint condition in decision models for portfolio choice. Zhao and Ziemba [14] consider a portfolio insurance strategy with an additional VaR constraint in a dynamic setting. Basak and Shapiro [2] consider a VaR condition in a model with CRRA utility and asset price dynamics defined by geometric Brownian motion, which gives an explicit formulation for the optimal investment strategy. This strategy controls downside risk at the VaR horizon. An alternative approach to controlling the wealth accumulation process, and particularly the downside, is developed in [6]. In the spirit of process control, upper and lower control limits (UCL and LCL) are specified for the wealth process. If the process hits either limit, then it is considered out of control, the investment decision and model specifications are evaluated, and adjustments are made. That is, the returns distributions are revised with additional data, new control limits are set, and a new investment strategy is computed. The decision on investment is put in the context of capital growth with security.

An illustration of the process control approach to risk is presented in Figure 26.1.

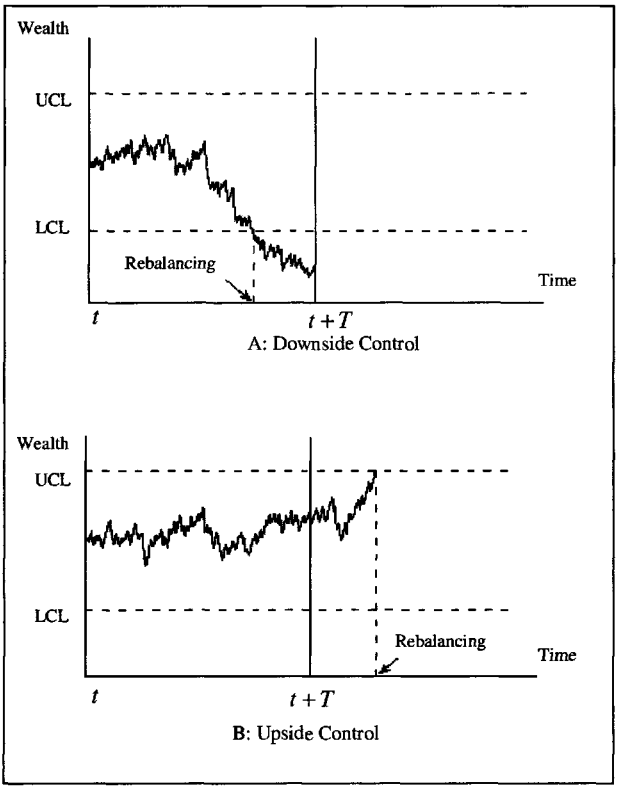


Figure 26.1. *Portfolio rebalancing with wealth goals.*

It can be seen that action is taken when a control limit is reached. That is, the rebalance time is random and reflects the fact that the current investment decision is not working as expected. One of the reasons for a mispriced investment decision is the use of forecasts for the future returns on assets. These forecasts are subject to estimation error, and such errors have a major impact on decisions and wealth accumulation [12]. Also shown in Figure 26.1 is a fixed planning horizon T, which is the basis for rebalancing in the VaR approach. The rationale for rebalancing is the signal that the dynamics of asset returns are not as forecasted, so the control limits are particularly helpful in this regard.

In this paper the methods for risk control are applied to the fundamental problem of asset allocation to stocks, bonds, and cash over time. The procedure used is a combination of VaR and wealth goals (control limits). Details on implementation of the return estimation, VaR decision, and control limits rebalancing are provided in section 26.2. The asset prices are modeled as geometric Brownian motion with random rates of return. Within that framework, there exist analytic solutions for Bayesian estimates of price parameters, optimal risk control investment strategies, and control limits. The optimal VaR is a blend of the risk-free asset and a benchmark portfolio, with the fraction depending on risk control parameters. The UCLs and LCLs for the trajectory of future wealth are based on the expected trajectory from the VaR decision. The application of risk control strategies to asset allocation is developed in section 26.3. Three assets are considered for investment: stocks, bonds, and cash. Underlying return distributions are established from data on historic prices. Then a Monte Carlo study is performed, where daily prices are simulated and capital is accumulated depending on the investment strategy followed. The VaR strategy with fixed time rebalancing and the VaR strategy with random time (control limits) rebalancing are implemented. Properties of the accumulated capital are compared for the alternative risk control approaches and for a range of settings on risk parameters. The contrast in accumulated capital emphasizes the distinction between periodic review (VaR) and a review based on wealth status (VaR + control limits).

26.2 Risk control investment strategies

The downside risk of investing in assets with uncertain return is the issue of concern, particularly the control of such risk while maintaining an expectation of strong positive returns. The framework for studying the problem is the standard lognormal asset pricing model. The stages in implementing a risk control system are (i) estimation of asset return, (ii) calculation of the risk return investment strategy, and (iii) identification of the time to rebalance. As indicated in Figure 26.2 this is a dynamic process, where estimates and strategies are updated based on additional information.

26.2.1 Estimation of asset returns

It is assumed that there are three assets for investment: stocks, bonds, and cash. The basis for investment decisions is current wealth, projections for future returns, and personal preferences for accumulated capital and risk. The asset prices are provided on a daily basis. Let the prices of assets be given as

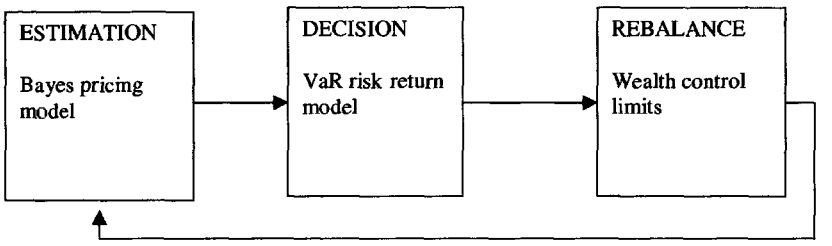


Figure 26.2. *Dynamic risk control.*

$P_0(t)$ = trading price of cash at time t,
$P_1(t)$ = trading price of stocks at time t,
$P_2(t)$ = trading price of bonds at time t.

It is assumed that the daily log prices follow an arithmetic random walk with the Gaussian increments

$$Z_i(t) = \ln P_i(t+1) - \ln P_i(t), \quad i = 0, 1, 2.$$

Then, $Z_i(t) = \pi_i + \delta_i \epsilon(t)$, where $\epsilon(t)$ are identically distributed standard Gaussian processes. For the means it is assumed that $\pi_0 = \ln(1+r)$, $\delta_0 = 0$, and (π_1, π_2) are normally distributed random variables that are dependent on a single latent state variable

$$\pi_i = \mu_i + \lambda_i F, \quad i = 1, 2,$$

where F is standard Gaussian.

If the covariance for risky asset log prices is $\Sigma = (\sigma_{ij}^2)$, then this structure implies the equation $\Sigma = \Lambda\Lambda^\top + \Delta$ with $\Lambda = (\lambda_1, \lambda_2)^\top$ and $\Delta = \text{diag}(\delta_1^2, \delta_2^2)$. That is, the covariance matrix can be factored. Furthermore, the components of the *fundamental equation* can be specified from the covariance Σ if the single state variable model is correct. The solution is

$$\Lambda = (\sqrt{\rho}\sigma_{11}, \sqrt{\rho}\sigma_{22})^\top,$$

$$\Delta = \begin{pmatrix} (1-|\rho|)\sigma_{11}^2 & 0 \\ 0 & (1-|\rho|)\sigma_{22}^2 \end{pmatrix},$$

where $\rho = \frac{\sigma_{12}}{\sigma_{11}\sigma_{22}}$ is the correlation between stocks and bonds. So the components of the state space model are identifiable from the mean and variance-covariance matrix. For an observation window, let the historical daily increments on log prices be $Z = Z_1, \ldots, Z_n$. With the prior $\pi = \mathbb{N}(\mu, \Lambda\Lambda^\top)$ and conditional distribution $Z|\pi \sim \mathbb{N}(\pi, \Delta)$, the posterior distribution for the mean increment given Z is

$$\pi|Z \sim \mathbb{N}(\tilde{\pi}, \Gamma),$$

Table 26.1. *Estimation of pricing parameters.*

Variable	Description
$\{Z_{1j}, \ldots, Z_{nj}\}$	Data on daily increments in rebalance cycle j
$(\bar{Z}_j, S_j)$	(Mean, standard deviation) of increments in cycle j
$\tilde{S}_j = 0.5\tilde{S}_{j-1} + 0.5S_j$	Smoothed covariance
(L_j, D_j)	Factor solution for $\tilde{S}_j$
$S_j^* = L_j L_j^\top + D_j$	Estimate for Σ
$\bar{\bar{Z}}_j = \frac{1}{n} e^\top \bar{Z}_j$	Grand mean estimate of μ
$\hat{\pi}_j = \bar{\bar{Z}}_j + (I - D_j S_j^{*-1})(\bar{Z}_j - \bar{\bar{Z}}_j)$	Estimated rate of return in cycle j

where

$$\tilde{\pi} = \mu + (I - \Delta\Sigma^{-1})(\overline{Z} - \mu),$$
$$\Gamma = (I - \Delta\Sigma^{-1})\Delta, \quad \text{and}$$
$$\overline{Z} = \frac{1}{n}\sum_{i=1}^{n} Z_i.$$

The vector $\tilde{\pi}$ at time t gives the Bayes estimate for the mean increment, given the price history to that date. The real significance of the above equations rests in the estimation of the defining parameters in price dynamics. From observed increments on log prices since the last rebalance point, the covariance S_j at the jth rebalancing is computed. S_j is an estimate of the covariance value Σ. The matrix S_j can be volatile, so it is combined with the previous smoothed estimate to get $\tilde{S}_j = 0.5\tilde{S}_{j-1} + 0.5S_j$, $j = 1, 2, \ldots$. (It is possible that Σ changes over time and the smoothed $\tilde{S}_j$ incorporates such dynamics.) The smoothed covariance is factored providing estimates of Σ, Λ, Δ. Then the observed mean and prior mean are combined to give the updated estimate $\tilde{\pi}$ of the conditional mean increment. This process is repeated at each rebalance point. The formulas for updating the estimates for parameters in the pricing model are summarized in Table 26.1.

26.2.2 VaR strategy

The dynamic stochastic prices on stocks and bonds provide the information required for decisions on the amount of current capital to invest in each opportunity. Let $x_i(t)$ be the fraction of wealth invested in asset i at time t for $i = 0, 1, 2$. With the budget constraint $\sum_{i=0}^{2} x_i(t) = 1$, the fraction invested in stocks and bonds is unconstrained since the fraction invested in cash, with borrowing/lending at the rate $1 + r$, can be chosen to balance the constraint. The strategy $X = (x_1, x_2)$ will be written as $q\tilde{X}$, where $\tilde{X}$ is a portfolio of risky assets and q is a fraction of wealth invested in that portfolio.

Table 26.2. *Calculation of VaR solution variables.*

Variables	Description
$\phi = (\phi_1, \phi_2)$	Random expected growth rate of risky assets
$\Delta = \text{diag}(\delta_1^2, \delta_2^2)$	Specific variance of risky assets
r	Rate of return on riskless asset
T	Planning horizon (rebalance interval)
w_t	Current wealth
$\beta = 1 - \frac{\underline{w}}{w_t}$	Fallback rate for VaR
α	Risk level: probability of fallback
$\tilde{X} = (\phi - re)\Delta^{-1}$	Kelly portfolio
Φ^{-1}	Inverse of Gaussian distribution
q^{VaR}	VaR fraction
Formula	$q^{VaR} = \frac{(B+\sqrt{B^2-4AC})}{2A}$, where $A = \frac{1}{2}\tilde{X}\Delta\tilde{X}^\top\sqrt{T}$, $B = \tilde{X}\Delta\tilde{X}^\top\sqrt{T} + \Phi^{-1}(\alpha)\sqrt{\tilde{X}\Delta\tilde{X}^\top}$, $C = (\beta - r)\sqrt{T}$.

Consider an investor who revises investment decisions at regular intervals of length T. At the current rebalance point, a forecast for the price parameters over time period T is developed and an investment strategy is calculated. The strategy is chosen to maximize the expected utility of accumulated capital at the horizon T, while keeping the risk of substantial loss at a low level. Let current wealth be $W(t)$ and the wealth after time T be $W(t+T)$. If the VaR is $\underline{W}$ and the risk level is α, then the VaR problem is

$$\begin{aligned} &\text{Maximize} && E\,u(W(t+T)), \\ &\text{Subject to} && \Pr[W(t+T) \geq \underline{W}] \geq 1 - \alpha. \end{aligned}$$

If the utility of wealth has constant relative risk aversion, i.e., $u(w) = \frac{1}{\gamma}w^\gamma$, $\gamma \leq 1$, then the VaR problem has a *fractional* Kelly solution [2, 6]. That is, the solution $q\tilde{X}$ is such that $\tilde{X}$ is the solution to the unconstrained expected utility problem, and the fraction $q \leq 1$ captures the risk constraint. The solution for the case $\gamma = 0$, i.e., log utility, is presented in Table 26.2. With log utility the objective is the expected growth rate, so $\tilde{X}$ is the optimal growth rate or *Kelly portfolio*.

The closed form of the solution in Table 26.2 follows from the lognormal distribution for $W(t+T)$, where

$$W(t+T) = w_t \exp\left\{T\tfrac{1}{2}q\tilde{X}\Delta Z + (q\tilde{X}(\phi - re)^\top + r - \tfrac{1}{2}q^2\tilde{X}\Delta\tilde{X}^\top)T\right\},$$

$$\phi_i = \pi_i + \tfrac{1}{2}\delta_i^2, \qquad i = 1, 2.$$

The form of the VaR solution is revealing about the way risk is controlled. The solution

is a blend of a benchmark portfolio of risky assets which optimizes expected utility and a riskless asset, with the blend fraction controlling the risk.

26.2.3 Wealth control limits

An important feature of the VaR approach is the VaR horizon T, where the αth quantile of the wealth distribution is controlled. The strategy to control wealth is based on estimates of the parameters which drive the asset prices. The strategy and the wealth trajectory are very sensitive to estimation errors for those parameters [4, 12]. A natural way to deal with the uncertain direction of a trajectory of the stochastic dynamic wealth process is to set process control limits and to *adjust* (update estimates for returns and resolve the growth-security problem) when a limit is reached. The limits can be selected so that they are consistent with the specification for the growth-security problem for wealth at time T. To develop the control limits consider

$T_w(X(t), w_t)$ = first passage time to wealth w, starting from wealth w_t at time t and following strategy $X(t)$.

Then the UCL will be set from the expected return at the horizon, the wealth level expected if the estimates for model parameters are correct. It is expected that the UCL is reached exactly as the time horizon arrives. The LCL provides *downside risk control*. To match the security provided by the VaR problem, the LCL is selected so that the wealth process with optimal strategy will reach the LCL before the UCL at most $100\alpha\%$ of the time. The UCL is

$$\bar{w} = E[W(T)|X(t) = X_{VaR}(t)],$$

and the LCL is

$$\underline{w} = \sup\{w|\Pr[T_w(X_{VaR}(t)) < T_{\bar{w}}(X_{VaR}(t))] \leq \alpha_L\}.$$

The closed form of the $X_{VaR}(t)$ solution facilitates the computation of the control limits. With $X_{VaR}(t) = q^{VaR}\tilde{X}(t)$, let

$$\mu(\tilde{X}) = \sum \tilde{x}_i(\phi_i - r) + r, \quad \sigma^2(\tilde{X}) = \sum \tilde{x}_i^2\delta_i^2, \quad \theta(\tilde{X}) = \frac{2\mu(\tilde{X}) - \sigma^2(\tilde{X})}{\sigma^2(\tilde{X})}.$$

Then control limits applicable to that decision are

- UCL: the upper limit is the expected wealth at the horizon

$$\bar{w} = E[W(T)|\tilde{X}_{VaR}(t)] = w_t \exp\left\{\sum_{i=1}^{K} q\tilde{x}_i(\hat{\phi}_i - r) + r\right\} T.$$

- LCL: the lower control limit is computed for known $\bar{w}$ by

$$\underline{w} = w_t \left(\frac{\alpha_L}{1 - (1 - \alpha_L)\left(\frac{w_t}{\bar{w}}\right)^{\theta(X)}}\right)^{\frac{1}{\theta(\tilde{X})}}.$$

Table 26.3. *Calculation of control limits.*

Variable	Description
$\mu(\tilde{X}) = \sum \tilde{x}_i(\phi_i - r) + r$	Mean rate of return on Kelly portfolio
$\sigma^2(\tilde{X}) = \sum \tilde{x}_i^2 \delta_i^2$	Variance of return on Kelly portfolio
$\theta(\tilde{X}) = \frac{2\mu(\tilde{X}) - \sigma^2(\tilde{X})}{\sigma^2(\tilde{X})}$	Sharpe ratio for Kelly portfolio
q^{VaR}	VaR investment fraction
$\bar{w} = w_t \exp\{(\sum q^{VaR} \tilde{x}_i(\phi_i - r) + r)T\}$	UCL
$\underline{w} = w_t \left(\frac{\alpha_L}{1-(1-\alpha_L)(w_t/\bar{w})^{\theta(\tilde{X})}} \right)^{\frac{1}{\theta(\tilde{X})}}$	LCL

The formulas for the wealth limits are in Table 26.3.

To summarize the approach, the growth-security problem is solved at time t for an optimal strategy based on forecast returns and the planning horizon of T. Then control limits are computed, which are consistent with the growth-security specifications and serve as stopping boundaries for the wealth process. The intention is that the portfolio rebalancing would take place only when a boundary is reached and the trajectory is not proceeding as anticipated.

26.3 Application to fundamental problem

The risk control procedures of the previous section are now applied to the allocation of investment capital to stocks, bonds, and cash over time. Baseline values for the parameters in the dynamic pricing model will be established using data from past returns. Then daily prices will be generated and investment strategies computed using the formulations in Tables 26.1, 26.2, and 26.3. A number of studies deal with the VaR strategy [2], but little is known about the effect of wealth limits [9]. Even in the case of VaR, most research assumes that the price parameters are known, and the impact of risk settings (α, β) on strategies compatible with those parameters are studied. In practice a single trajectory to date is known, and the chance of incompatibility between the pricing process and a strategy based on estimates is high. In this study, strategies based on observed trajectories are considered. In particular the advantage of using wealth status rather than time as the basis for rebalancing the portfolio will be developed. The information basis for investment will be daily trading prices for stocks, bonds, and cash. These prices will be generated from the random rates of return model in section 26.2. The baseline values for the true price process are presented in Table 26.4. They are determined from statistics on returns from the S&P 500, Solomon Brothers bond index, and U.S. Treasury bills.

Figure 26.3 gives sample price trajectories on stocks and bonds from the model. The recent past may be misleading in determining an investment strategy for the near future.

The approach to analyzing investment decisions is to take the VaR model as the standard and to calculate upper and lower wealth limits using the VaR strategy. At rebalance

Table 26.4. *Daily rates of return.*

	Stocks	Bonds	Cash
Mean	0.00050	0.00031	0.00019
Variance	0.00062	0.00035	0
Covariance	0.000046		

Figure 26.3. *Daily prices for stocks and bonds.*

time τ, first the strategy is computed with risk specifications (T, β, α). Then the upper wealth limit is defined as $\overline{w} = E\,[W^{VaR}(\tau + T)]$ = expected wealth at the VaR horizon with the VaR strategy. Finally, given values w_t, $\gamma = 1 - \alpha$, $\overline{w}$, the value $\underline{w}$ is computed using the formulas in Table 26.3.

Table 26.5 gives examples of strategies for the initial phase ($w_0 = 1$, $T = 10$) and a range of values for the VaR parameters (α, β). The benchmark strategy is the same for each combination, so the differences result from changes in q^{VaR}. Significant changes in strategy result from variations in risk specifications.

For each VaR strategy, the upper limit and the lower limit are calculated to determine the next rebalancing time. Table 26.6 provides the control limits corresponding to the scenarios in Table 26.5. The values in these tables will be updated as the portfolio is rebalanced.

The computational experiment which tests the risk control methodology consists of generating daily prices for stocks and bonds for 1 year (260 trading days), using the log-normal model with parameter values in Table 26.3. The results that follow will consider (i) VaR strategies with fixed time rebalancing set every 10 days, and (ii) VaR strategies with random time rebalancing determined by the control limits, where the upper limit is the

Table 26.5. *Initial VaR strategies (stocks, bonds).*

	Fallback $(1-\beta)$				
	0.95	0.96	0.97	0.98	0.99
0.01	(1.34, 0.89)	(1.08, 0.72)	(0.82, 0.55)	(0.56, 0.37)	(0.30, 0.20)
0.02	(1.52, 1.01)	(1.22, 0.81)	(0.93, 0.62)	(0.64, 0.42)	(0.35, 0.23)
0.03	(1.65, 1.10)	(1.33, 0.89)	(1.01, 0.68)	(0.69, 0.46)	(0.38, 0.25)
0.04	(1.78, 1.18)	(1.43, 0.95)	(1.09, 0.72)	(0.75, 0.50)	(0.40, 0.27)
0.05	(1.89, 1.26)	(1.52, 1.01)	(1.16, 0.77)	(0.79, 0.53)	(0.43, 0.29)

Table 26.6. *Control limits* $(\overline{w}, \underline{w})$ *for VaR strategy.*

	Fallback $(1-\beta)$				
	0.95	0.96	0.97	0.98	0.99
0.01	(1.005, 0.998)	(1.005, .998)	(1.005, .998)	(1.004, .999)	(1.003, .997)
0.02	(1.005, 0.999)	(1.005, 0.999)	(1.005, .999)	(1.005, .999)	(1.003, .999)
0.03	(1.005, 1.000)	(1.005, 1.000)	(1.005, 1.000)	(1.005, 1.000)	(1.003, 1.000)
0.04	(1.005, 1.000)	(1.005, 1.000)	(1.005, 1.000)	(1.005, 1.000)	(1.004, 1.000)
0.05	(1.005, 1.000)	(1.005, 1.000)	(1.005, 1.000)	(1.005, 1.000)	(1.004, 1.000)

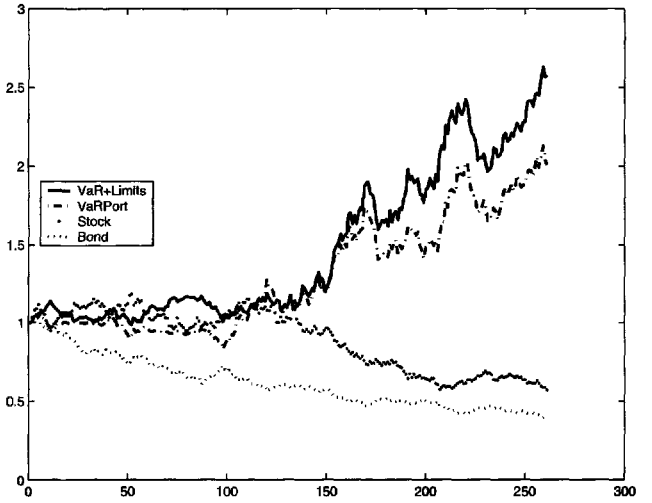


Figure 26.4. *Performances against stocks and bonds.*

expected VaR wealth after 10 days.

An illustration of a wealth trajectory from a sequence of VaR strategies without limits (fixed time rebalancing) and VaR strategies with limits (random time rebalancing) is given in Figure 26.4. Also shown are wealth paths from all stocks and all bonds strategies. In this

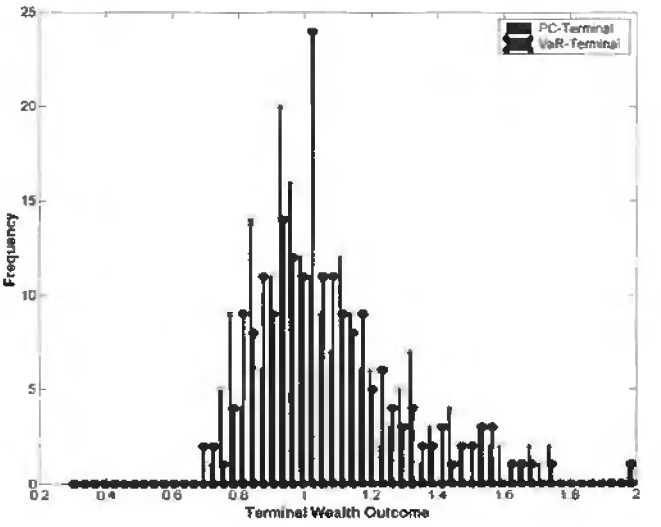


Figure 26.5. *Scenario distribution of terminal wealth.*

scenario, the portfolio rebalanced at random times outperforms the others.

The terminal wealth distribution from 200 scenarios using VaR strategy without limits (VaR) and VaR strategy with limits (VaR + limits) is provided in Figure 26.5. The VaR + limits wealth distribution has a heavier right tail, indicating superior returns.

The expected returns from the 200 scenarios for the full range of VaR parameters are given in Table 26.7. The average terminal wealth when control limits are included is higher in most of the scenarios considered. The advantage grows as the value at risk (fallback) decreases. This improved performance is attributable to rebalancing at the right time, that is, when the forecast for returns on assets is clearly in error.

The standard deviation of terminal wealth is given in Table 26.8. These values are reasonably large, and the results for random rebalancing are a bit higher. Since wealth is lognormally distributed, the longer right tail increases the standard deviation. This effect is seen in the wealth distribution in Figure 26.5.

26.4 Discussion

The allocation of capital to risky investment opportunities is a complex task, where information and preferences are combined to determine an investment portfolio in an uncertain financial market. In this paper a wealth limits approach to the control of risk in financial markets is studied in an application involving stocks, bonds, and cash. The purpose of the wealth limits is to signal the need to rebalance the portfolio. The rebalance times are the random hitting times of the stochastic wealth process. The random rebalance time approach to risk is contrasted with the conventional fixed rebalance time approach in a VaR methodology. The results in the numerical experiments conducted here indicate that strategies supplemented with wealth goals perform well in comparison to VaR strategies based on

Table 26.7. *Expected terminal wealth.*

VaR strategy without control limits (rebalance interval = 10 days)

	Fallback $(1-\beta)$				
	0.95	0.96	0.97	0.98	0.99
0.01	1.1292	1.1053	1.0881	1.0764	1.0629
0.02	1.1509	1.1189	1.0953	1.0797	1.0649
0.03	1.1703	1.1310	1.1019	1.0826	1.0666
0.04	1.1895	1.1432	1.1084	1.0854	1.0681
0.05	1.2091	1.1557	1.1154	1.0884	1.0696

VaR strategy with control limits (random rebalancing)

	Fallback $(1-\beta)$				
	0.95	0.96	0.97	0.98	0.99
0.01	1.2156	1.1737	1.1395	1.1050	1.0905
0.02	1.2227	1.1700	1.1315	1.1024	1.0799
0.03	1.2573	1.2059	1.1547	1.1187	1.0730
0.04	1.2802	1.2045	1.1708	1.1227	1.0714
0.05	1.2853	1.2085	1.1696	1.1270	1.0726

Table 26.8. *Standard deviation of wealth.*

VaR strategy without control limits (rebalance interval = 10 days)

	Fallback $(1-\beta)$				
	0.95	0.96	0.97	0.98	0.99
0.01	.7439	.5702	.4165	.4757	.1468
0.02	.8874	.6704	.4837	.3216	.1686
0.03	1.0127	.7567	.5403	.3551	.1862
0.04	1.1335	.8390	.5932	.3858	.2022
0.05	1.2565	.9214	.6450	.4157	.2174

VaR strategy with control limits (random rebalancing)

	Fallback $(1-\beta)$				
	0.95	0.96	0.97	0.98	0.99
0.01	.8524	.6407	.4562	.3047	.1515
0.02	1.0507	.7793	.5504	.3605	.1813
0.03	1.2260	.8934	.6189	.3981	.2050
0.04	1.4192	1.0069	.6994	.4374	.2237
0.05	1.6186	1.1381	.7756	.4757	.2397

fixed rebalance times. The VaR strategy with wealth limits has higher expected return with comparable downside risk. A major ingredient of an investment strategy is the forecast for return on assets. Errors in those forecasts imply some misdirection in the strategy. The

significance of the control limits is that serious misdirection will be identified early rather than at a periodic review time. Intervention when data indicate that forecasts are too much in error will result in a timely adjustment in estimates for returns parameters. Conversely, if forecasts are good and the wealth process unfolds as expected, then adjustment (rebalancing) is unnecessary. The choice of control limits in the implementation was designed to provide downside risk equivalent to that of the VaR strategy. Obviously this is arbitrary and likely suboptimal. An important open question concerns the best selection of control limits and the corresponding strategy for specified risk requirements.

Acknowledgment

This research was partially supported by the Natural Sciences and Engineering Research Council of Canada.

Bibliography

[1] K. J. Arrow, *The role of securities in the optimal allocation of risk-bearing*, Rev. Econ. Stud., 31 (1964), pp. 503–546.

[2] S. Basak and A. Shapiro, *Value-at-risk based risk management: Optimal policies and asset prices*, Rev. Financial Stud., 14 (2001), pp. 371–405.

[3] C. Breitmeyer, H. Hakenes, A. Pfingsten, and C. Reshtien, *Learning from Poverty Measurement: An Axiomatic Approach to Measure Downside Risk*, Working Paper, University of Muenster, Muenster, Germany, 1999.

[4] V. Chopra and W. Ziemba, *The effect of errors in means, variances, and covariances on optimal portfolio choice*, J. Portfolio Management, 19 (1993), pp. 6–11.

[5] P. Jorion, *Value-at-Risk: The New Benchmark for Controlling Market Risk*, Irwin, Chicago, 1997.

[6] L. MacLean, Y. Zhao, and W. Ziemba, *Risk control of dynamic investment models*, J. Banking and Finance, forthcoming.

[7] L. MacLean and W. Ziemba, *Growth versus security tradeoffs in dynamic investment analysis*, Ann. Oper. Res., 85 (1999), pp. 193–225.

[8] L. MacLean, W. Ziemba, and G. Blazenko, Management Sci., 38 (1992), pp. 1562–1585.

[9] L. MacLean, W. Ziemba, and Y. Li, *Time to Wealth Goals in Capital Accumulation*, Stochastic Programming E-Print Series, 2002.

[10] R. Merton, *Continuous-Time Finance*, Blackwell Publishers, Cambridge, MA, 1992.

[11] J. W. Pratt, *Risk aversion in the small and in the large*, Econometrica, 32 (1964), pp. 122–136.

[12] L. C. ROGERS, *The relaxed investor and parameter uncertainty*, Finance and Stochastics, 5 (2001), pp. 131–154.

[13] L. J. SAVAGE, *The Foundations of Statistics*, John Wiley, New York, 1954.

[14] Y. ZHAO AND W. T. ZIEMBA, *A dynamic asset allocation model with downside risk control*, J. Risk, 3 (2000), pp. 91–113.

Chapter 27

Scenario-Based Risk Management Tools

Helmut Mausser and Dan Rosen**

27.1 Introduction

Financial institutions worldwide have devoted much effort to developing systems that integrate information across their organizations to measure, monitor, and manage risk on an enterprise-wide basis. Beyond simply measuring risk, an effective risk management function first must help the firm to understand the sources of its exposures and to identify the major risk contributors. Second, it should indicate how changes in external (i.e., market) risk factors or in the portfolio itself (i.e., potential trades) affect the firm's risk. Finally, it must provide the means to optimally trade off risk and reward, both within and across various business lines.

Risk management requires tools that construct a comprehensive picture of all types of risk faced by the firm and that permit the effective utilization of the wealth of financial products available in the markets to obtain the desired risk and reward profiles. A risk manager's toolkit includes risk analytics that decompose the overall portfolio risk, explain the effects of new trades, and identify potential hedges for individual instruments, as well as mathematical programming models that allow portfolios to be optimally restructured.

The most widely used risk management tools extend the insights originally developed in [29] and [43] in modern portfolio theory. Thus, they assume that the underlying risk factor changes follow a joint normal (or, more generally, elliptic) distribution and that the asset values depend linearly on these risk factors. Essentially, this takes the implicit view that the variance (or standard deviation) of the portfolio's losses is an appropriate measure of risk. Despite their onerous assumptions, these parametric tools constitute a solid conceptual basis for a risk management toolkit, and they have been applied to both market (e.g., [28])

*Algorithmics, Inc., 185 Spadina Avenue, Toronto, ON, M5T 2C6 Canada (hmausser@algorithmics.com, drosen@algorithmics.com).

and credit risk (e.g., [25]).

Recognizing the limitations of simple formulaic approaches, methodologies that simulate the portfolio over a set of risk factor scenarios are now increasingly used for measuring market, credit, liquidity, and operational risks, as well as for managing assets and liabilities (see, for example, [3, 10, 23, 39, 50]). Simulation offers several advantages over simpler parametric approaches in that it readily accommodates

- nonlinear instruments, such as options;
- complex risk factor processes that give rise to nonnormal distributions;
- multiple time horizons;
- discrete markets, reflecting the fact that trading is costly and liquidity is limited in practice;
- the natural integration of various types of risk (e.g., market, credit, operational, liquidity) within a single framework;
- a broad range of risk measures.

Simulation comes with the cost of increased computational complexity, since the portfolio must be priced in all scenarios in order to compute an empirical loss distribution. However, it is widely (and incorrectly) believed that scenario-based risk management tools are impractical because they entail a significant additional computational burden [14, 37]. In fact, it is possible to obtain risk analytics, comparable to those of parametric approaches, without any additional simulation beyond that required for risk measurement purposes. This is one of the foundations of the mark-to-future (MtF) methodology described in [10].

Many applications of simulation and scenario-based tools to risk management problems are documented in the literature. Mausser and Rosen [31, 33, 36] construct trade risk profiles and triangular risk decompositions, and compute risk analytics for quantile-based risk measures such as value at risk (VaR) and expected shortfall (ES). A rich set of scenario-based optimization models has been developed for financial and risk management applications involving both single- and multistage decisions (see, for example, [48, 7, 50, 8]). The expected downside risk measure known as regret [9, 11, 12] is both broadly relevant and computationally efficient, given that it can be formulated as a linear program. Examples of this type of risk measure in financial applications are given in [26] and [49]. Rockafellar and Uryasev [40] show that ES also can be optimized using linear programming. Regret and ES are used to reduce the credit risk of a bond portfolio in [34, 35, 4].

In this paper, we construct trade risk profiles and compute the marginal risk, best hedge, and risk contribution for the positions in a portfolio. These analytics allow risk managers to reduce the overall portfolio risk by trading individual instruments. We also demonstrate that when scenario-based tools are applied to quantile-based risk measures, particularly VaR, the quality of the risk analytics depends on the selected quantile estimator. Greater potential risk reductions are afforded by optimization models, which consider multiple trades concurrently. Our discussion of such models will be limited to the problem of minimizing risk subject to a set of trading and return constraints over a single time period. In particular, optimizing VaR requires solving an integer program, whose size makes its exact solution

impractical. We show that minimizing a more tractable measure, such as ES or regret, is an effective heuristic approach for obtaining portfolios with a low VaR.

This paper is organized as follows. The next section introduces the notion of risk measurement and defines VaR and ES. We then provide general descriptions of the risk management tools that will be considered in the paper. The risk measures and tools are first presented for the case in which losses are normally distributed with mean zero, and then scenario-based versions are developed for simulation approaches. We illustrate the tools on a set of three sample portfolios, respectively consisting of a collection of foreign exchange (FX) forward contracts, an equity option portfolio, and a set of credit-risky bonds issued by emerging markets. The final section offers several concluding remarks.

27.2 Risk measurement

We suppose that a portfolio $\mathbf{x}$ comprises a set of positions, where the position size x_i is the number of units (e.g., shares, contracts) of instrument i, for $i = 1, 2, \ldots, N$. More generally, $\mathbf{x}$ might be chosen to represent a set of dollar values or relative weightings for both individual and aggregated positions (i.e., subportfolios). Suppose that the current unit value of instrument i is v_i^0 and its (uncertain) unit value at some specified future time is v_i, and let $\Delta v_i = v_i^0 - v_i$ denote the unit loss of instrument i. The random variables $L_i(x_i) \equiv \Delta v_i \cdot x_i$ and $L(\mathbf{x}) \equiv \sum_{i=1}^{N} L_i(x_i)$ define the loss in value of position i and of the portfolio, respectively, over the time horizon.

Given a set V of real-valued random variables, a risk measure is a mapping $\rho : V \to \mathbb{R}$. For our purposes, consider $X \in V$ to represent the loss in the value of some asset or portfolio over a specified period (i.e., $X \equiv L_i(x_i)$ for an individual position or $X \equiv L(\mathbf{x})$ for a portfolio).

Historically, risk has often been measured in terms of the standard deviation, or volatility, $\sigma(X)$. Recently, the quantile-based measure VaR has become widely adopted by financial institutions and regulators for allocating risk capital. Specifically, for a given time horizon, the $100\alpha\%$ VaR, denoted $\mathrm{VaR}_\alpha(X)$, is the size of loss that will be exceeded with probability $(1-\alpha)$. Thus, if losses follow the cumulative distribution function (cdf) $F(X)$, then $\mathrm{VaR}_\alpha(X) = F^{-1}(\alpha)$.

Unfortunately, neither of the aforementioned risk measures is coherent in the sense defined by [5]. The standard deviation violates monotonicity—adding a position that cannot lose value to a portfolio can nevertheless increase the portfolio's volatility. While VaR is monotonic, it is not subadditive—the VaR of a portfolio consisting of two positions may exceed the sum of the individual position VaRs. This runs counter to the tenet of diversification as a means of reducing risk.

The insurance industry has long used a risk measure, ES, that is coherent [1, 38]. The $100\alpha\%$ ES, denoted $\mathrm{ES}_\alpha(X)$, is the average of the $100(1-\alpha)\%$ largest losses and is calculated as follows:

$$\begin{aligned}\mathrm{ES}_\alpha(X) = (1-\alpha)^{-1}(&\mathbb{E}[X \cdot 1_{\{X \geq \mathrm{VaR}_\alpha(X)\}}] \\ &- \mathrm{VaR}_\alpha(X) \cdot (\alpha - \mathbb{P}[X < \mathrm{VaR}_\alpha(X)])),\end{aligned} \tag{27.1}$$

where the indicator function $1_{\{\xi\}}$ equals 1 if the condition ξ is satisfied and zero otherwise.

Equation (27.1) includes a correction term to ensure that the total probability of the averaged losses is exactly $(1-\alpha)$ in the case of discrete loss distributions (see [2]). ES and slight variations thereof have been reported in the literature under various names, including "tail conditional expectation" [5] and "conditional VaR" [40]. Cariño et al. [7] define shortfall more generally as a piecewise linear, convex, downside risk measure.

27.3 Risk management tools

The fact that instruments respond differently to underlying risk factors allows managers to affect the risk of a portfolio by modifying its constituent positions. We now present a set of tools, ranging from simple risk analytics that consider each instrument independently to stochastic optimization models that operate at the portfolio level, for helping risk managers in this regard. While the discussion considers risk over a single time horizon, the concepts are also applicable when risk is aggregated across multiple time steps. We denote the risk of a portfolio consisting of positions $\mathbf{x}$ by $\rho(\mathbf{x})$

Trade risk profile The trade risk profile (TRP) for instrument i plots the portfolio risk against the position size x_i while all other position sizes remain constant. It provides risk managers with a quick, visual summary of several key risk characteristics for each instrument or subportfolio within the portfolio. The convexity of the TRP depends on the risk measure; if the portfolio loss is linear in the position sizes, then the TRP is convex for standard deviation and ES [41] but not necessarily for VaR [17].

Best hedge position The best hedge position for a given instrument or subportfolio i is the position size that minimizes the portfolio risk (all other positions remaining fixed). Thus, it corresponds to the minimum of the TRP.

Marginal risk Marginal risk measures the impacts of infinitesimal changes in instrument positions on the portfolio risk. It is defined as the partial derivative of the portfolio risk with respect to the position size of a given instrument or subportfolio. Thus, it is the slope of the tangent to the TRP at the current position size.

As shown by [17], the marginal VaR for instrument i is its expected loss per unit conditional on the portfolio loss being equal to the VaR:

$$\frac{\partial \mathrm{VaR}_\alpha(\mathbf{x})}{\partial x_i} = \mathbb{E}[\Delta v_i | L(\mathbf{x}) = \mathrm{VaR}_\alpha(\mathbf{x})]. \tag{27.2}$$

Similarly, the marginal ES is (see [42, 46])

$$\frac{\partial \mathrm{ES}_\alpha(\mathbf{x})}{\partial x_i} = (1-\alpha)^{-1}\{(\mathbb{E}[\Delta v_i \cdot 1_{\{L(\mathbf{x}) \geq \mathrm{VaR}_\alpha(\mathbf{x})\}}] - \mathbb{E}[\Delta v_i | L(\mathbf{x}) = \mathrm{VaR}_\alpha(\mathbf{x})] \cdot (\alpha - \mathbb{P}[L(\mathbf{x}) < \mathrm{VaR}_\alpha(\mathbf{x})])\}. \tag{27.3}$$

The finite counterpart of marginal risk, known as incremental risk, measures the change in risk that results from adding an entire position to a portfolio.

Risk contribution By decomposing risk, a risk manager is able to identify the most significant sources of risk, or the portfolio's so-called hot spots [28]. Since coherent risk measures are subadditive, the portfolio risk is typically less than the sum of the incremental risks of its constituent positions due to diversification. However, risk measures that are homogeneous of degree one (which includes standard deviation, VaR, ES, and all coherent risk measures) admit a marginal decomposition:

$$\rho(\mathbf{x}) = \sum_{i=1}^{N} x_i \frac{\partial \rho(\mathbf{x})}{\partial x_i}. \tag{27.4}$$

In (27.4), each term in the summation is the product of the position size and the rate of change of risk with respect to that position. This product essentially represents the rate of change of risk with respect to a small percentage change in the size of the position. Define

$$C(x_i) = \frac{1}{\rho(\mathbf{x})} \times x_i \frac{\partial \rho(\mathbf{x})}{\partial x_i} \times 100\% \tag{27.5}$$

to be the percentage risk contribution of the ith position. Equation (27.5) must be interpreted on a marginal basis; it indicates the relative contributions to the change in risk that results if all positions are scaled by the same amount. At the best hedge position, a position's marginal risk, and therefore also its contribution, is zero.

Optimal portfolios Although the analytics described above are relatively simple, in that each position or subportfolio is examined in isolation, they represent powerful hands-on tools for proactively managing portfolio risk. In fact, their simplicity is a large factor in their overall appeal. At the same time, greater risk reductions can be obtained when multiple positions are modified concurrently. Moreover, additional considerations, such as the portfolio's return and budgetary restrictions, typically limit the trades that can be undertaken. Issues such as these can be addressed through the use of stochastic optimization models for portfolio risk management.

27.4 Parametric approach and normality

If the portfolio losses are normally distributed with mean zero, the task of risk management is greatly simplified. In this case, VaR and ES are constant multiples of volatility, and so these measures are effectively equivalent from a risk management perspective. The delta-normal method [24], a covariance-based approach to calculating risk, gives rise to simple, closed-form representations of VaR and ES and also of the associated risk analytics [28, 16, 31]. Furthermore, the mean variance model for trading off risk and return [29] yields efficient portfolios with respect to all three risk measures.

27.4.1 Normal risk measures

If $L(\mathbf{x}) \sim \mathcal{N}(0, \sigma(\mathbf{x}))$, then $\mathrm{VaR}_\alpha(\mathbf{x})$ satisfies

$$\int_{\mathrm{VaR}_\alpha(\mathbf{x})}^{\infty} \frac{1}{\sqrt{2\pi}\sigma(\mathbf{x})} e^{\frac{-(L(\mathbf{x}))^2}{2(\sigma(\mathbf{x}))^2}} dL(\mathbf{x}) = \alpha \tag{27.6}$$

and

$$\text{VaR}_\alpha(\mathbf{x}) = Z_\alpha \sigma(\mathbf{x}), \tag{27.7}$$

where Z_α is the standard normal z-value that delimits a probability of α in the right tail (e.g., $Z_{0.01} = 2.3263$).

$\text{ES}_\alpha(\mathbf{x})$ is the conditional expectation of all losses that exceed $\text{VaR}_\alpha(\mathbf{x})$. Computing this expectation using (27.6) yields

$$\text{ES}_\alpha(\mathbf{x}) = K_\alpha \sigma(\mathbf{x}), \tag{27.8}$$

where

$$K_\alpha = \frac{1}{\alpha} \int_{Z_\alpha}^{\infty} y \cdot \frac{1}{\sqrt{2\pi}} e^{\frac{-y^2}{2}} dy$$

is the conditional expectation of all standard normal variates exceeding Z_α (e.g., $K_{0.01} = 2.6652$).

27.4.2 Delta-normal method

The delta-normal method for calculating risk assumes the existence of a set of risk factors whose log price changes are joint normally distributed with zero mean; that is, if $\hat{r}_k$ is the log return of risk factor k, then $\hat{\mathbf{r}} \sim \mathcal{N}(0, \mathbf{Q}^*)$, where $\mathbf{Q}^*$ is the covariance matrix of risk factor returns.

The VaR map of a portfolio, denoted $\mathbf{m}(\mathbf{x})$, specifies the monetary value of the portfolio's position in each risk factor. The VaR map provides a reduced, or simplified, view of the portfolio from a risk management perspective. Note that

$$\mathbf{m}(\mathbf{x}) = \sum_{i=1}^{N} \mathbf{m}^i x_i, \tag{27.9}$$

where $\mathbf{m}^i$ is the VaR map of one unit of the ith instrument (i.e., m_k^i is the exposure to risk factor k that results from holding a single unit of instrument i).

If the value of each instrument is a linear function of the risk factor values, then the volatility of the portfolio losses is

$$\sigma(\mathbf{x}) = \sqrt{\mathbf{m}(\mathbf{x})^T \mathbf{Q}^* \mathbf{m}(\mathbf{x})}. \tag{27.10}$$

From (27.7), (27.8), and (27.10), it follows that the portfolio's risk can be expressed generally as

$$\rho(\mathbf{x}) = \sqrt{\mathbf{m}(\mathbf{x})^T \mathbf{Q} \mathbf{m}(\mathbf{x})}, \tag{27.11}$$

where $\mathbf{Q} = g^2 \mathbf{Q}^*$ is a scaled covariance matrix and the constant g equals 1, Z_α, or K_α for the volatility, VaR, or ES, respectively.

27.4.3 Delta-normal risk analytics

Since the delta-normal method yields a simple formula for the portfolio's risk (i.e., (27.11)), it is straightforward to obtain the risk analytics in closed form in this case.

Trade risk profile Under the delta-normal approach, the TRP has the same characteristic shape regardless of whether risk is measured by volatility, VaR, or ES. To construct the TRP for instrument i, we fix the positions in all instruments other than i to their current values and consolidate them into a subportfolio. Denote the volatilities of instrument i and the subportfolio by σ_i and σ_I, respectively, and their correlation by ρ_{iI}. The volatility of the portfolio is

$$\sigma(x_i) = \sqrt{(x_i\sigma_i)^2 + \sigma_I{}^2 + 2\rho_{iI}x_i\sigma_i\sigma_I}. \tag{27.12}$$

From (27.11) and (27.12) it follows that the TRP is a curve of the form

$$f(x_i) = \sqrt{ax_i{}^2 + bx_i + c}, \tag{27.13}$$

where $a = (g\sigma_i)^2$, $b = 2g^2\rho_{iI}\sigma_i\sigma_I$, and $c = (g\sigma_I)^2$.

Best hedge position To find the best hedge position, we first differentiate (27.13) with respect to x_i:

$$\frac{df(x_i)}{dx_i} = \frac{2ax_i + b}{2f(x_i)}.$$

Since $f(x_i)$ is strictly positive for risky portfolios, the best hedge position occurs at

$$x_i^* = -\frac{b}{2a} = -\frac{\rho_{iI}\sigma_I}{\sigma_i}.$$

Note that x_i^* is independent of g (i.e., the best hedge position minimizes the volatility, VaR, and ES).

Marginal risk From (27.11), the gradient with respect to the risk factor exposures is

$$\nabla_m \rho(\mathbf{x}) = \frac{\mathbf{Qm(x)}}{\rho(\mathbf{x})}.$$

The kth element of the gradient is the change in risk that results from increasing the portfolio's exposure to the kth risk factor (i.e., $m_k(\mathbf{x})$) by a single monetary unit.

Since the VaR map of the portfolio is the sum of the VaR maps for the positions (i.e., (27.9)), it follows that the derivative of the portfolio risk with respect to the ith position is

$$\frac{\partial\rho(\mathbf{x})}{\partial x_i} = (\mathbf{m}^i)^T(\nabla_m\rho(\mathbf{x})). \tag{27.14}$$

Risk contribution The percentage contribution of the ith position to the portfolio risk is obtained by substituting (27.14) into (27.5):

$$C(x_i) = \frac{1}{\rho(\mathbf{x})} \times x_i(\mathbf{m}^i)^T(\nabla_m\rho(\mathbf{x})) \times 100\%.$$

Mean variance optimization The classic mean variance optimization model proposed by [29] suggests that efficient portfolios are those that earn the highest expected return for a given variance of returns. When portfolio losses are normally distributed, this model is appropriate for managing risk.

Define r_i to be the expected return of instrument i, σ_{ik} to be the covariance between r_i and r_k, and a_i to be the fraction of the portfolio value allocated to instrument i. The mean variance problem is

$$\begin{aligned} \min \quad & \sum_{i=1}^{N}\sum_{k=1}^{N} \sigma_{ik} a_i a_k \\ \text{s.t.} \quad & \sum_{i=1}^{N} r_i a_i \geq R, \\ & \sum_{i=1}^{N} a_i = 1, \\ & a_i \geq 0 \quad \text{for } i = 1, \ldots, N, \end{aligned} \tag{27.15}$$

where R is a given target portfolio return. By setting $R = 0$, one effectively obtains the minimal variance portfolio, and solving the problem parametrically in R yields an efficient frontier.

Problem (27.15) can be reformulated in terms of values and position sizes by substituting

$$a_i = \left(\frac{v_i}{V}\right) x_i,$$

where v_i is the value of one unit of instrument i and V is the portfolio value. In this case, the problem minimizes the variance of the portfolio loss while achieving a specified expected increase in the portfolio value. Since VaR and ES are simply constant multiples of the volatility when portfolio losses are normally distributed with mean zero, the solution to (27.15) is optimal for all three of these risk measures.

27.5 Scenario-based approach

In practice, the normality assumption is often inappropriate. For example, portfolios that contain derivatives or that are exposed to credit risk typically have loss distributions that are skewed and fat tailed. A scenario-based approach, which simulates the portfolio over a set of risk factor scenarios, more readily accommodates realistic risk factor distributions and nonlinear or credit-risky instruments. In this case, various risk measures can be computed from the empirical loss distribution obtained from the simulation.

The fact that volatility is a poor measure of risk when losses are not normally distributed is well documented in the literature (e.g., [21, 35]). Thus, while it is possible to compute the standard deviation of the sample losses, we focus only on VaR and ES. Computing these measures is essentially a problem of quantile estimation, which is typically addressed in one of two ways.

Semiparametric methods fit a distribution to the sample losses and then estimate the quantile from this distribution (e.g., [20, 22]). Extreme value approaches [15], which focus

explicitly on the tail of the loss distribution, also belong in this category. In contrast, nonparametric methods make no explicit distributional assumptions—a quantile is derived directly from the data, which are assumed to be an independent, identically distributed sample from an unknown loss distribution. In this case, a point estimate of the quantile is calculated from the order statistics of the sample. (The kth order statistic is the kth smallest value in the sample.)

In this paper, we restrict our attention to a class of nonparametric methods known as L-estimators, which compute the quantile as a weighted average of the order statistics (see, for example, [13, 44]). Their simple, linear structure facilitates the calculation of risk analytics and the development of optimization models. However, the choice of a particular L-estimator can be of significance when computing marginal risk. In particular, an estimator that is based on only a single order statistic can yield unreliable estimates of the marginal VaR (see [30]).

27.5.1 Calculating portfolio losses

Given a particular base case scenario (e.g., representative of current market conditions), it is straightforward to calculate the gain or loss in portfolio value in each scenario. Let v_i^0 and v_{ij} denote, respectively, the unit value of instrument i in the base scenario and its unit value in scenario j, $j = 1, 2, \ldots, S$, at the appropriate time horizon. Thus, in scenario j, $\Delta v_{ij} = v_i^0 - v_{ij}$ is the unit loss of instrument i and the portfolio loss is

$$L_j(\mathbf{x}) = \sum_{i=1}^{N} \Delta v_{ij} x_i. \tag{27.16}$$

The fact that the portfolio loss is a linear function of the position sizes facilitates the calculation of risk analytics in a scenario-based environment. In particular, it is not necessary to resimulate the portfolio to compute the risk analytics; computing the values v_{ij}, which takes the bulk of the computational effort in a scenario-based approach, needs to be done only once.

27.5.2 L-estimators for VaR and ES

Consider a set of S scenarios and suppose, for ease of exposition, that the likelihood of each scenario is $1/S$. Let $L_{(k)}(\mathbf{x})$ denote the kth-order statistic of the set of S sample losses of the portfolio containing the positions $\mathbf{x}$, so that $L_{(1)}(\mathbf{x}) \leq L_{(2)}(\mathbf{x}) \leq \cdots \leq L_{(N)}(\mathbf{x})$. Using an L-estimator, the $100\alpha\%$ VaR is estimated as

$$\mathrm{VaR}_\alpha(\mathbf{x}) = \sum_{k=1}^{S} w_{\alpha,S,k}^V L_{(k)}(\mathbf{x}), \tag{27.17}$$

where the weights satisfy

$$\sum_{k=1}^{S} w_{\alpha,S,k}^V = 1. \tag{27.18}$$

The weights are independent of the position sizes and need to be calculated only once for any given sample size (S) and quantile (α).

To obtain an estimator for the expected shortfall, observe that in the case of a continuous distribution F, ES satisfies (suppressing the dependence on the positions $\mathbf{x}$ for the moment)

$$\begin{aligned} \mathrm{ES}_\alpha &= \mathbb{E}[F^{-1}(p) | p \geq \alpha] \\ &= \frac{1}{1-\alpha} \int_\alpha^1 F^{-1}(p) dp. \end{aligned} \tag{27.19}$$

Since the loss distribution is unknown, we replace $F^{-1}(p)$ in (27.19) by its estimate $\mathrm{VaR}_p(\mathbf{x})$ from (27.17) and estimate the $100\alpha\%$ ES as

$$\begin{aligned} \mathrm{ES}_\alpha(\mathbf{x}) &= \frac{1}{1-\alpha} \int_\alpha^1 \left(\sum_{k=1}^S w_{p,S,k}^V L_{(k)}(\mathbf{x}) \right) dp \\ &= \sum_{k=1}^S \left(\frac{1}{1-\alpha} \int_\alpha^1 w_{p,S,k}^V dp \right) L_{(k)}(\mathbf{x}). \end{aligned}$$

Defining

$$w_{\alpha,S,k}^E = \frac{1}{1-\alpha} \int_\alpha^1 w_{p,S,k}^V dp, \tag{27.20}$$

we obtain an L-estimator for expected shortfall

$$\mathrm{ES}_\alpha(\mathbf{x}) = \sum_{k=1}^S w_{\alpha,S,k}^E L_{(k)}(\mathbf{x}).$$

It is straightforward to verify from (27.20) that if (27.18) is satisfied, then

$$\sum_{k=1}^S w_{\alpha,S,k}^E = 1.$$

UECV estimator One common L-estimator for VaR is the sample quantile, also known as the upper empirical cumulative distribution function value (UECV). For example, given a sample of 100 losses, the UECV estimates VaR at the 95% level as the 96th-order statistic (i.e., the fifth-largest loss). More generally, the UECV weights for estimating VaR at level α are

$$w_{\alpha,S,k}^V = \begin{cases} 1 & \text{if } k = \lfloor S\alpha \rfloor + 1, \\ 0 & \text{otherwise.} \end{cases} \tag{27.21}$$

Alternatively, (27.21) can be expressed in terms of the quantile level p as

$$w_{p,S,k}^V = \begin{cases} 1 & \text{if } \frac{k-1}{S} \leq p < \frac{k}{S}, \\ 0 & \text{otherwise.} \end{cases} \tag{27.22}$$

To obtain the UECV weights for estimating the expected shortfall at level α, let $i = \lfloor S\alpha \rfloor + 1$ and write (27.20) as

$$w^E_{\alpha,S,k} = \frac{1}{1-\alpha}\left(\int_{\alpha}^{\frac{i}{S}} w^V_{p,S,k}dp + \int_{\frac{i}{S}}^{\frac{i+1}{S}} w^V_{p,S,k}dp + \cdots + \int_{\frac{S-1}{S}}^{1} w^V_{p,S,k}dp\right). \tag{27.23}$$

From (27.22) and (27.23), it follows that

$$w^E_{\alpha,S,k} = \begin{cases} \frac{\lfloor S\alpha \rfloor + 1 - S\alpha}{S(1-\alpha)} & \text{if } k = \lfloor S\alpha \rfloor + 1, \\ \frac{1}{S(1-\alpha)} & \text{if } (\lfloor S\alpha \rfloor + 1) < k \leq S, \\ 0 & \text{otherwise.} \end{cases}$$

If $S = 100$, then $\mathrm{ES}_{0.975}$ is a weighted average of the three largest losses:

$$\mathrm{ES}_{0.975}(\mathbf{x}) = \frac{1}{5}L_{(98)}(\mathbf{x}) + \frac{2}{5}L_{(99)}(\mathbf{x}) + \frac{2}{5}L_{(100)}(\mathbf{x}).$$

Harrell–Davis estimator The Harrell–Davis (HD) estimator [19] is based on the fact that, as the sample size increases, the expected value of order statistic $(S+1)\alpha$ converges to $F^{-1}(\alpha)$ for $0 < a < 1$. Thus, the HD estimator computes VaR_α as $\mathbb{E}[L_{((S+1)\alpha)}]$, regardless of the integrality of $(S+1)\alpha$. The resulting weights are

$$\begin{aligned} w^V_{\alpha,S,k} &= \frac{1}{\beta[(S+1)\alpha, (S+1)(1-\alpha)]}\int_{(k-1)/S}^{k/S} y^{(S+1)\alpha-1}(1-y)^{(S+1)(1-\alpha)-1}dy \\ &= I_{\frac{k}{S}}[(S+1)\alpha, (S+1)(1-\alpha)] \\ &\quad - I_{\frac{(k-1)}{S}}[(S+1)\alpha, (S+1)(1-\alpha)], \end{aligned} \tag{27.24}$$

where $I_x[a, b]$ is the incomplete beta function.

From (27.20) and (27.24), it follows that the HD estimator yields the following weights for estimating ES_α:

$$\begin{aligned} w^E_{\alpha,S,k} = \frac{1}{1-\alpha}\int_{\alpha}^{1} \{&I_{\frac{k}{S}}[(S+1)p, (S+1)(1-p)] \\ &- I_{\frac{(k-1)}{S}}[(S+1)p, (S+1)(1-p)]\}dp. \end{aligned} \tag{27.25}$$

The weights in (27.24) and (27.25) must be calculated using numerical procedures.

27.5.3 Risk analytics for *L*-estimators

The linearity of L-estimators facilitates the calculation of risk analytics for VaR and ES. (Technically, we compute analytics for the estimates of VaR and ES under the assumption that the actual measures are in fact differentiable, as discussed in [45].) To simplify the notation, let $\Delta v_{i(k)}$ denote the per unit loss of instrument i in the scenario that results in the kth smallest portfolio loss, given the positions $\mathbf{x}$. (Although we do not explicitly write $\Delta v_{i(k)}$ as a function of $\mathbf{x}$, this dependency will be recognized in the subsequent analysis.)

Since the results apply identically to VaR and ES, we let $w_{\alpha,S,k}$ and $\rho_\alpha(\mathbf{x})$ denote the weight and risk measure, respectively.

From (27.16) and (27.17)

$$\begin{aligned}\rho_\alpha(\mathbf{x}) &= \sum_{k=1}^{S} w_{\alpha,S,k} \sum_{i=1}^{N} \Delta v_{i(k)} x_i \\ &= \sum_{i=1}^{N} \omega_{\alpha,S,i} x_i,\end{aligned} \tag{27.26}$$

where

$$\omega_{\alpha,S,i} = \sum_{k=1}^{S} w_{\alpha,S,k} \Delta v_{i(k)}$$

represents the weighted loss per unit of instrument i. Note that $\omega_{\alpha,S,i}$ is constant as long as the order of the scenarios, ranked by the size of the portfolio loss, does not change. Since this ranking typically changes as the position sizes vary, it follows that $\rho_\alpha(\mathbf{x})$ is a piecewise linear function of $\mathbf{x}$.

Trade risk profile From the preceding discussion, it follows that under a scenario-based approach, L-estimators produce a TRP that is piecewise linear. The smoothness of the resulting TRP depends significantly on the particular choice of L-estimator; in particular, an estimator that relies on only a single order statistic, such as UECV, tends to produce a "jagged" TRP for VaR. The TRP for ES, since it is estimated as an average of multiple order statistics, tends to be smoother regardless of the estimator used.

The TRP for instrument i can be constructed by systematically detecting changes in the order statistics as the position size x_i is varied. (An algorithm that constructs the TRP for VaR using the UECV estimator is given in [32].) However, this approach, which effectively locates the endpoints of the linear segments comprising the TRP, can be costly from a computational standpoint. Instead, one may choose to simply compute the risk at a number of different position sizes and then interpolate linearly between them to construct the TRP. Since the intervals can be made arbitrarily small, the TRP can be obtained to any desired level of accuracy.

Best hedge position Given the piecewise linearity of the TRP, identifying the best hedge position requires finding the minimum of all sampled points. As such, obtaining an accurate estimate of the best hedge position is contingent on constructing a TRP that spans an appropriate range of position sizes and on using an interval that is sufficiently small. Since the best hedge position is determined primarily by the overall shape of the TRP, rather than by the slopes of the individual segments, the choice of L-estimator tends to have only a minor impact on estimates of x_i^*.

Marginal risk In the case of L-estimators, the marginal risk takes on a very simple form. From (27.26) it follows that

$$\frac{\partial \rho_\alpha(\mathbf{x})}{\partial x_i} = \omega_{\alpha,S,i} \tag{27.27}$$

is the marginal risk of instrument i. To view this result in terms of (27.2) and (27.3), one can interpret the weight as the probability that $L_{(k)}(\mathbf{x})$ equals $\mathrm{VaR}_\alpha(\mathbf{x})$.

Since the marginal risk equals the slope of the TRP for instrument i, it is a piecewise constant function of x_i. The slope is undefined at the vertices of the TRP. In this case, multiple scenarios give rise to an identical loss, and so the order statistics do not provide a unique ordering of the scenarios. We propose to consider two one-sided derivatives at such points. Thus, identical portfolio losses are ranked in order of increasing unit losses (Δv_{ij}) when computing the marginal risk in the positive direction, and in order of decreasing unit losses when computing the marginal risk in the negative direction.

Risk contribution From (27.26) and (27.27)

$$\rho_\alpha(\mathbf{x}) = \sum_{i=1}^{N} \frac{\partial \rho_\alpha(\mathbf{x})}{\partial x_i} x_i,$$

and so the marginal contribution of the ith position to the portfolio risk is

$$C(x_i) = \frac{\omega_{\alpha,S,i} x_i}{\rho_\alpha(\mathbf{x})} \times 100\%.$$

Scenario-based optimization Optimization models involving VaR and ES are typically based on quantile estimators that use a single order statistic since they are more likely to yield computationally tractable formulations than estimators that average multiple order statistics. The HD estimator, for instance, requires all portfolio losses to be ranked in sequence, which entails the use of $O(S^2)$ binary variables. If the number of scenarios is large, as is usually the case in practice, the differences between estimators are negligible. However, to verify the accuracy of the results, one could compute the risk of the optimal portfolio using several different estimators if desired.

Although VaR and ES are both estimated from the order statistics of the sample losses, they give rise to markedly different optimization models. VaR models must be formulated as integer programs (e.g., [47]) which, because of their size, can often be solved only approximately. In contrast, the convexity of ES permits an efficient linear program formulation [40].

Consider the problem of minimizing the risk of a portfolio with a current value V^0 that earns a specified expected return R. Further, suppose that the position size of instrument i must remain within the limits $[l_i, u_i]$. Given a set of S equally likely scenarios, the following integer program minimizes VaR_α (denoted by the variable κ):

$$\begin{aligned}
&\min && \kappa && \\
&\text{s.t.} && L_j(\mathbf{x}) - Mz_j - \kappa \le 0, && j = 1, \ldots, S, \\
& && \sum_{j=1}^{S} z_j \le \lceil S(1-\alpha) \rceil - 1, && \\
& && \sum_{i=1}^{N} v_i^0 x_i = V^0, && (27.28)
\end{aligned}$$

$$\sum_{i=1}^{N} v_i^0(r_i - R)x_i \geq 0,$$

$$l_i \leq x_i \leq u_i, \qquad i = 1, \ldots, N,$$

$$z_j \in \{0, 1\}, \qquad j = 1, \ldots, S.$$

In (27.28), $M \gg 0$ is a suitably chosen positive constant. The variable z_j equals 1 if the portfolio loss in scenario j exceeds VaR_α, and is zero otherwise. This formulation requires a total of S binary variables, making its exact solution difficult in practice, where S may number in the thousands.

Minimizing ES_α requires solving the linear program

$$\begin{aligned}
\min \quad & \kappa + \frac{1}{1-\alpha}\left(\frac{1}{S}\right)\sum_{j=1}^{S} y_j \\
\text{s.t.} \quad & L_j(\mathbf{x}) - \kappa - y_j \leq 0, \qquad j = 1, \ldots, S, \\
& \sum_{i=1}^{N} v_i^0 x_i = V^0, \\
& \sum_{i=1}^{N} v_i^0(r_i - R)x_i \geq 0, \\
& l_i \leq x_i \leq u_i, \qquad i = 1, \ldots, N, \\
& \kappa \geq 0, \\
& y_j \geq 0, \qquad j = 1, \ldots, S.
\end{aligned} \tag{27.29}$$

The concept of regret [9], which measures a portfolio's deviation from a benchmark, is ideally suited for scenario-based risk management The following linear program minimizes the expectation of losses that exceed the predefined threshold K:

$$\begin{aligned}
\min \quad & \frac{1}{S}\sum_{j=1}^{S} y_j \\
\text{s.t.} \quad & L_j(\mathbf{x}) - y_j \leq K, \qquad j = 1, \ldots, S, \\
& \sum_{i=1}^{N} v_i^0 x_i = V^0, \\
& \sum_{i=1}^{N} v_i^0(r_i - R)x_i \geq 0, \\
& l_i \leq x_i \leq u_i, \qquad i = 1, \ldots, N, \\
& y_j \geq 0, \qquad j = 1, \ldots, S.
\end{aligned} \tag{27.30}$$

More generally, the threshold K may represent a scenario-dependent benchmark, such as an index.

By varying the parameter R in (27.28)–(27.30), it is possible to construct an efficient frontier, consisting of portfolios that earn a specified return at minimal risk.

27.6 Examples

We now illustrate the risk management tools by applying them to several sample portfolios, representative of various practical risk management problems. First, we consider a collection of FX forward contracts, which might be entered by an institution hedging against currency fluctuations, for example. In this case, the assumption that portfolio losses are normally distributed with mean zero is found to be acceptable, and thus the delta-normal approach can be applied. Next, we examine a portfolio of options and equities that implements a strangle position, representing a possible strategy by a trading desk. As expected, the normality assumption does not hold in this case and scenario-based tools are required. Finally, we evaluate the credit risk of a portfolio of bonds from emerging markets, as might be held by a bank. Since credit loss distributions are typically highly skewed and fat tailed, scenario-based methods are again appropriate. We use optimization to rebalance the bond portfolio and improve the risk/return trade-off. More detailed analyses of these portfolios can be found in [4, 30, 31, 34, 35, 36].

27.6.1 FX portfolio

Table 27.1 shows a portfolio of FX forward contracts as of July 1, 1997. Given that the exchange rates, in U.S. dollars (USD), are 0.73 (CAD), 0.58 (DEM), 0.17 (FRF), and 0.0090 (JPY), the current value of the portfolio is 122,000 USD. Simulating the portfolio over 1000 scenarios produces a sample of losses with a mean and standard deviation of -2120 and 34,431, respectively. (The scenarios are generated under the assumption that the risk factor log returns are joint normally distributed with covariances as specified in the RiskMetrics risk factor data set of that day.) A Kolmogorov–Smirnov test does not reject, at the 95% level, the hypothesis that the losses are normally distributed with mean zero. (See also Figure 27.1, which plots the empirical loss distribution against the $\mathcal{N}(0, 34431)$ distribution.) Thus, we can safely use the delta-normal method to compute the portfolio risk over a 1-day time horizon. The portfolio's 99% VaR and 99% ES are 78,000 USD and 89,700 USD, respectively.

Table 27.1. *FX portfolio.*

Instrument	Currency	Days to maturity	Strike price (USD)	Position ($\times 10^6$)	Value ($\times 10^3$ USD)
CAD/USD .73 100d	CAD	100	0.7300	0.5	2.5
CAD/USD .74 30d	CAD	30	0.7400	1.0	−8.3
DEM/USD .57 60d	DEM	60	0.5700	6.0	73.2
DEM/USD .59 120d	DEM	120	0.5900	5.0	−28.2
FRF/USD .16 40d	FRF	40	0.1600	8.0	83.3
JPY/USD .0091 11d	JPY	11	0.0091	10.0	−0.9

The analysis (Table 27.2) indicates that the DEM contracts are the major source of risk, contributing approximately 83% of the current portfolio risk. In contrast, the CAD contracts, with a total contribution of -0.70%, act as a hedge. The marginal risk values

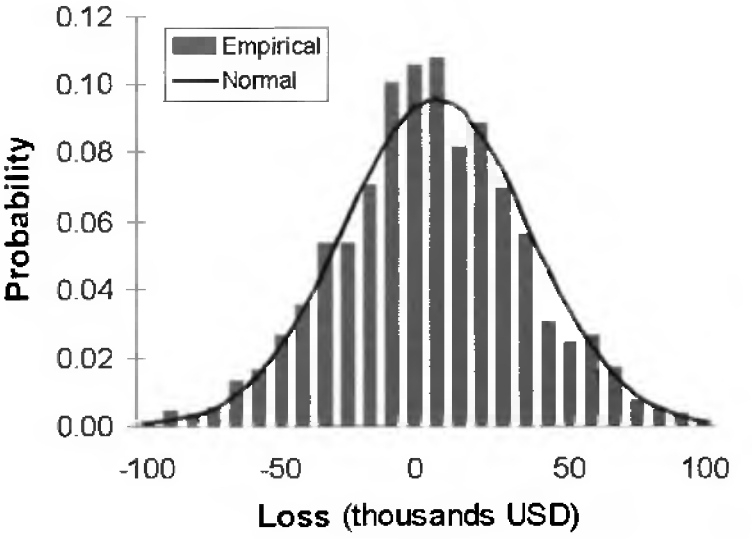


Figure 27.1. *Distribution of losses for the FX portfolio* (1000 *scenarios*).

Table 27.2. *Risk analytics for FX portfolio.*

Instrument	Risk contrib. (%)	Marginal VaR ($\times 10^{-4}$ USD)	Marginal ES ($\times 10^{-4}$ USD)	Best hedge position ($\times 10^{6}$)	Risk reduction (%)
DEM/USD .57 60d	45.4	59.24	67.87	−7.0	87.7
DEM/USD .59 120d	37.7	58.97	67.56	−8.0	87.6
FRF/USD .16 40d	16.5	16.19	18.55	−38.3	79.2
JPY/USD .0091 11d	1.1	0.88	1.01	−209.2	13.2
CAD/USD .73 100d	−0.2	−3.80	−4.36	1.4	0.2
CAD/USD .74 30d	−0.5	−3.85	-4.41	1.9	0.2

indicate that increasing the positions in the DEM, FRF, and JPY contracts increases the portfolio risk, while a similar increase in the CAD contracts reduces risk. This is also reflected by the individual best hedge positions, which require shorting DEM, FRF, and JPY contracts and purchasing CAD contracts.

The impact of holding the best hedge position in a given instrument can be measured in terms of the percentage risk reduction that can be achieved (i.e., the resulting decrease in risk expressed as a percentage of the current risk). The percentage reductions are identical for volatility, VaR, and ES. At their best hedge positions, the DEM contracts each reduce the risk by almost 88%, while each CAD contract offers a much smaller reduction of only 0.2%.

In many cases, however, it may not be feasible to hold an instrument at its best hedge position. For example, being short 7 million units of DEM/USD .57 60-day contracts may well run counter to the underlying objectives of the portfolio. Thus, it is useful to consult the TRP to determine the reduction that can be achieved within practical limitations. For

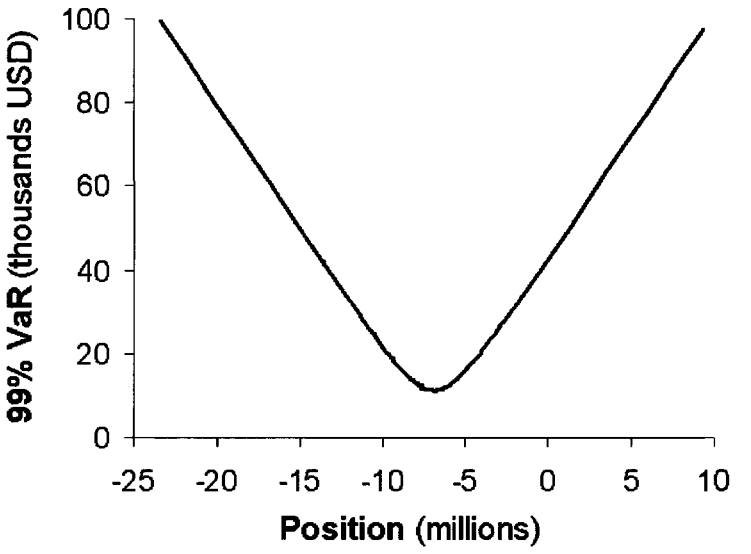


Figure 27.2. *TRP for DEM/USD .57 60d.*

example (Figure 27.2), eliminating the DEM/USD .57 60-day position reduces the 99% VaR by almost half (to approximately 43,000 USD).

27.6.2 NIKKEI strangle portfolio

Table 27.3 shows a representative trading desk portfolio, with a current value of 12,493 million JPY, that implements a strangle on two component stocks of the NIKKEI index. In addition to common shares of Komatsu (current price 840,000 JPY) and Mitsubishi (current price 860,000 JPY), the portfolio includes several European call and put options on these

Table 27.3. *NIKKEI portfolio.*

Instrument	Type	Days to Maturity	Strike price ($\times 10^3$ JPY)	Position ($\times 10^3$)	Value ($\times 10^3$ JPY)
Komatsu	Equity	na	na	2.5	2,100,000
Mitsubishi	Equity	na	na	2.0	1,720,000
Komatsu Cjul29 900	Call	7	900	−28.0	−11,593
Mitsubishi Cjul29 800	Call	7	800	−16.0	−967,280
Mitsubishi Csep30 836	Call	70	836	8.0	382,070
Mitsubishi EC 6mo 860	Call	184	860	11.5	563,340
Komatsu Cjun2 760	Call	316	760	7.5	1,020,110
Komatsu Cjun2 670	Call	316	670	22.5	5,150,461
Komatsu Paug31 760	Put	40	760	−10.0	−68,919
Komatsu Paug31 830	Put	40	830	10.0	187,167
Mitsubishi Psep30 800	Put	70	800	40.0	2,418,012

equities. The strangle incurs small losses if there are minor changes in the equity prices but earns increasing profits as these price changes become larger.

Simulating the portfolio over a set of 1000 scenarios on the index level (the equity prices are obtained from the CAPM model, with both stocks having positive betas) indicates that the losses are not normally distributed with mean zero (Figure 27.3). The UECV and HD estimators measure the 99% VaR of the portfolio to be 2.85 million JPY and 2.84 million JPY, respectively, and the 99% ES to be 3.65 million JPY and 3.79 million JPY, respectively.

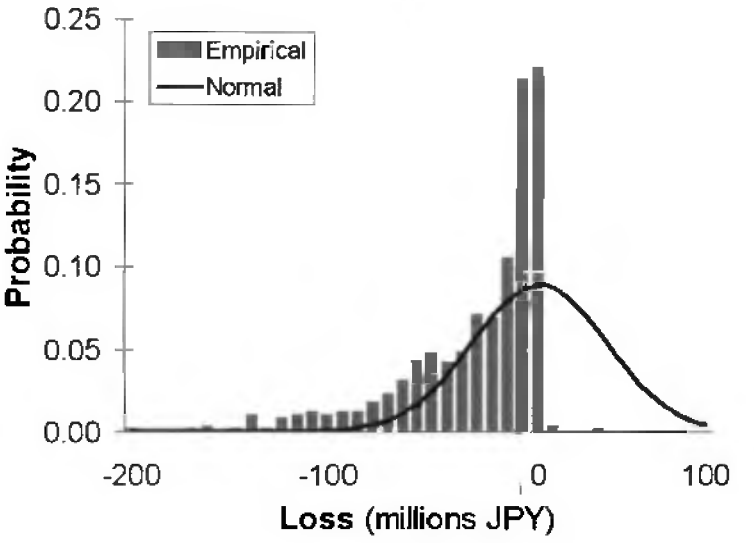


Figure 27.3. *Distribution of losses for the NIKKEI strangle* (1000 *scenarios*).

Before reviewing the VaR analytics for the strangle, it is instructive to first examine the TRPs for Mitsubishi EC 6mo 860, one of the call options in the portfolio (Figure 27.4). The TRP produced by the UECV estimator displays an undesirable jaggedness in this case, which can potentially result in a marginal VaR that is of the wrong sign. For example, the TRP slopes upward (the marginal VaR is positive) for position sizes in the ranges [5100, 5500] and [6500, 7600], which is inconsistent with the overall shape of the TRP. Essentially, this jaggedness is due to identical portfolio losses that result from an increase in the price of Mitsubushi EC 6mo 860 in one case and a price decrease in the other.

To reduce potential errors, it is possible to fit a smooth curve to the data (a third-order polynomial approximation is denoted UECV-s in Figure 27.4) before computing the marginal VaR and best hedge position. However, one drawback of this approach is that it requires constructing the TRP for a range of position sizes, which can be computationally inefficient when one is interested only in the marginal VaR at the current position size, for example (see [18]). Furthermore, the resulting curve depends on the specified functional form and on the range of position sizes selected.

The TRP from the HD estimator is smoother than that of the UECV estimator because the VaR is calculated as a weighted combination of several losses. Since there is no need to approximate the TRP in this case, it is possible to obtain a robust estimate of the marginal

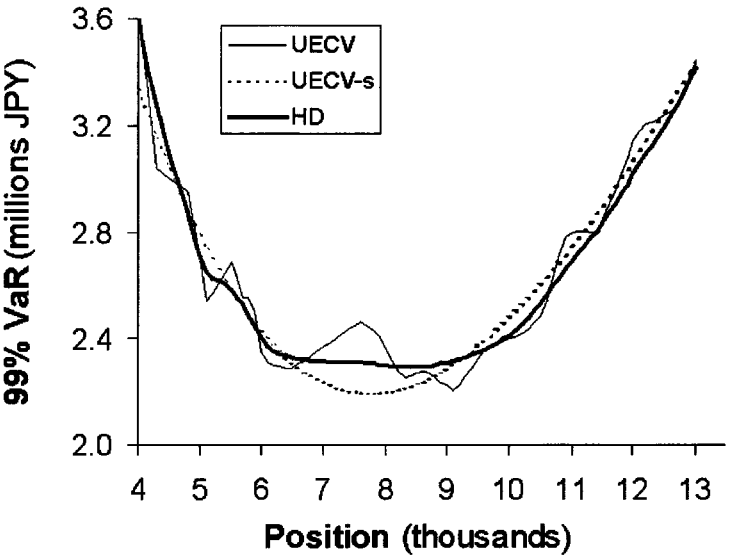


Figure 27.4. *99% VaR TRPs for Mitsubishi EC 6mo 860.*

Table 27.4. *99% VaR analytics for NIKKEI strangle.*

	VaR contrib. (%)		Marginal VaR (JPY)		Best hedge position ($\times 10^3$)		VaR reduction (%)	
Instrument	UECV	HD	UECV	HD	UECV	HD	UECV	HD
Komatsu Cjun2 670	2151	1341	2727	1692	20.1	19.6	41.3	41.2
Komatsu Cjun2 760	678	424	2576	1604	5.0	4.4	42.2	42.2
Mitsubishi Csep30 836	477	289	1699	1026	3.9	4.2	34.8	34.2
Mitsubishi EC 6mo 860	232	132	575	325	9.1	8.5	22.9	19.3
Komatsu	202	122	2300	1389	−0.4	−0.3	35.1	33.7
Mitsubishi	149	90	2119	1280	−1.2	−1.0	35.1	33.7
Komatsu Paug31 760	53	32	−152	−91	33.2	33.2	35.2	34.2
Komatsu Cjul29 900	−51	−31	52	31	−150.5	−145.5	34.7	33.3
Komatsu Paug31 830	−237	−143	−675	−407	19.8	19.8	34.6	34.0
Mitsubishi Cjul29 800	−1166	−706	2078	1254	−19.3	−19.1	34.8	33.6
Mitsubishi Psep30 800	−2387	−1449	−1702	−1029	44.2	43.8	34.7	34.4

VaR at the current position without having to first construct the TRP.

Table 27.4 lists the 99% VaR analytics for all positions in the portfolio. As might be expected, the magnitudes of the UECV estimates for risk contribution and marginal VaR are quite different from (larger by approximately 65% than) those of the HD estimates. However, their signs and relative sizes are consistent—for the current positions, positive risk contributions are due to the long calls, short puts, and common stocks, all of which appreciate in value when the broader index gains (under the assumed model), while the opposite is true

of the short calls and the long puts. The magnitudes of the VaR contributions are much larger that those for the FX portfolio; since the strangle constitutes a highly leveraged position, the risks incurred by individual positions tend to offset each other to a large extent.

The differences between estimators are negligible for the best hedge positions and the corresponding reductions in VaR since these values depend primarily on the overall shape of the curve rather than on its slope. The results suggest that risk can be reduced by taking a greater short position in the index by selling calls and equities or buying puts.

The ES 99% analytics (Table 27.5) are essentially consistent with those for VaR 99%. The positions are ranked identically with respect to risk contribution for both measures, although the ES contributions are smaller in magnitude than those for VaR. This implies that the ES risk is more evenly distributed among the positions than the VaR risk, which is also reflected by the fact that adopting the best hedge position affords less potential for reducing ES than VaR.

Table 27.5. *99% ES analytics for NIKKEI strangle.*

	ES contrib. (%)		Marginal ES (JPY)		Best hedge position ($\times 10^3$)		ES reduction (%)	
Instrument	**UECV**	**HD**	**UECV**	**HD**	**UECV**	**HD**	**UECV**	**HD**
Komatsu Cjun2 670	1138	1034	1846	1743	20.4	20.5	33.9	35.0
Komatsu Cjun2 760	360	327	1751	1655	5.0	5.3	34.8	35.0
Mitsubishi Csep30 836	244	220	1113	1043	5.3	5.2	28.7	29.6
Mitsubishi EC 6mo 860	118	106	375	348	6.1	6.9	15.0	16.1
Komatsu	103	93	1508	1413	0.5	0.4	28.1	29.1
Mitsubishi	76	69	1390	1302	−0.2	−0.2	28.1	29.1
Komatsu Paug31 760	27	24	−99	−93	20.8	21.3	29.0	30.0
Komatsu Cjul29 900	−26	−24	34	32	−116.2	−116.2	27.7	28.6
Komatsu Paug31 830	−121	−109	−441	−413	16.9	17.1	28.8	29.7
Mitsubishi Cjul29 800	−597	−538	1361	1275	−18.2	−18.3	28.2	29.1
Mitsubishi Psep30 800	−1223	−1103	−1116	−1046	42.7	42.8	28.9	29.8

The magnitudes of the UECV estimates for ES contribution and marginal ES exceed those of the HD estimates by only approximately 10%. The better agreement between the two estimators is due to averaging multiple order statistics when computing ES, which results in a smoother TRP for the UECV estimator (e.g., Figure 27.5).

27.6.3 Emerging markets bond portfolio

The market value of securities that are subject to credit risk, such as bonds and loans, depends on the perceived likelihood that the obligor (e.g., a government or a corporation) will be able to meet those obligations. Credit risk refers to the potential losses due to an obligor's default or, more generally, its transition to a lower credit rating.

Portfolio credit risk models account for correlations among the credit transitions of different obligors, thereby recognizing the benefits of diversification as a means of reducing credit risk. The tools described in this paper can be used effectively for managing the credit risk of portfolios that hold positions in multiple obligors.

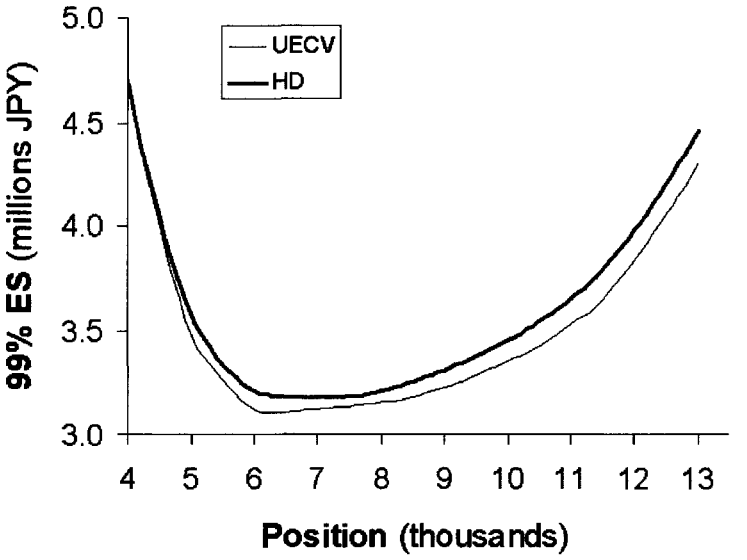


Figure 27.5. *99% ES TRPs for Mitsubishi EC 6mo 860.*

In measuring the credit risk of a portfolio of emerging markets bonds, Bucay and Rosen [6] employ a portfolio credit model that considers both credit migration and default events. The model assumes that there is a finite set of nondefault states (credit ratings), and the likelihood of an obligor migrating or defaulting within a one-year time horizon is specified in a credit transition matrix (provided by Standard and Poor's). Using implied forward rates, the model first computes exposures (i.e., the future value of an obligor's debt), under all possible credit states, for each obligor at a given time horizon. (In this case, there are seven credit states plus a default state.) A simulation is then performed on the joint credit states of all obligors at the horizon, and a portfolio value is obtained in each scenario by summing the exposures corresponding to the respective credit states. Finally, the portfolio loss is calculated by subtracting the resulting portfolio value from the forward value of the portfolio in the absence of any credit event.

The portfolio contains 197 long-dated corporate and sovereign bonds, issued by 86 obligors in 29 countries, with a mark-to-market value of 8.3 billion USD and a duration of approximately five years. The portfolio's annual expected return, computed using the one-year forward returns (assuming no credit migration) of each obligor's debt, is 7.26%; the additional expected return above the one-year risk-free rate of 5.86% compensates for the expected credit losses in this case. A simulation of the bond portfolio over 20,000 scenarios and a one-year horizon yields a characteristic credit loss distribution (Figure 27.6) that is highly skewed and fat tailed. The 99% VaR is 1.026 billion USD and 1.028 billion USD from the UECV and HD estimators, respectively, while the estimates of the 99% ES are 1.320 billion USD and 1.323 billion USD, respectively.

With this portfolio credit model, an obligor can incur only a small number of possible losses in each scenario (one for each credit state), which accordingly restricts the possible marginal VaR values found by the UECV estimator. The TRPs for Mexican debt (Figure

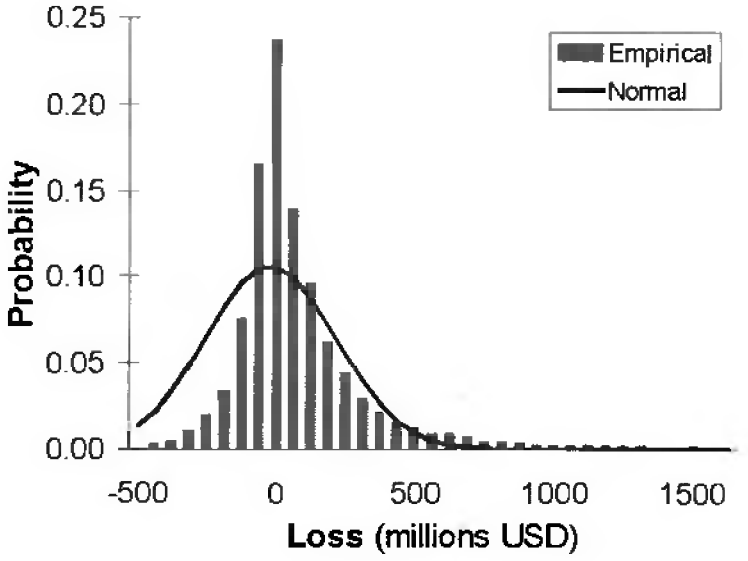


Figure 27.6. *Distribution of losses for the bond portfolio* (20,000 *scenarios*).

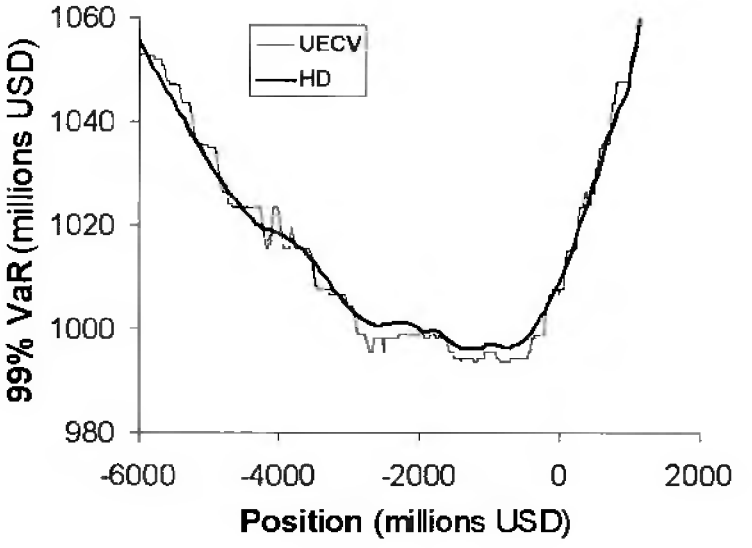


Figure 27.7. *TRPs for Mexico.*

27.7) reflect this fact; since an obligor is most likely to maintain its current rating over a one-year period, the UECV estimator often incorrectly computes the marginal VaR (and VaR contribution) to be zero, as indicated by the horizontal segments in the TRP.

Table 27.6 lists the risk analytics for the eight largest contributors to the 99% VaR of the portfolio, as determined by the HD estimator. (Note that the one-dollar marginal

Table 27.6. *99% VaR analytics for bond portfolio.*

Obligor	Current value ($\times 10^6$ USD)	VaR contrib. (%)		$1 Marginal VaR (USD)		Best hedge position ($\times 10^6$ USD)		VaR reduction (%)	
		UECV	HD	UECV	HD	UECV	HD	UECV	HD
Brazil	894	24.69	25.85	0.28	0.30	−4,494	−4,409	41	41
Russia	758	15.12	19.22	0.20	0.26	−6,599	−5,863	35	35
Venezuela	414	7.23	13.30	0.18	0.33	−1,375	−1,374	34	34
Argentina	636	8.83	12.47	0.14	0.20	−4,966	−4,986	28	28
Peru	279	17.54	7.57	0.65	0.28	−1,855	−1,865	28	28
Colombia	608	7.12	2.35	0.12	0.04	−27,040	−26,960	21	21
Mexico	491	0.00	2.09	0.00	0.04	−1,392	−1,135	3	3
Russia CCC	44	2.02	1.90	0.47	0.44	−812	−814	27	27

VaR is the change in VaR that results from investing one additional dollar in an obligor's debt.) Some differences are apparent between the HD and UECV estimators; the latter apparently overestimates the marginal VaRs and contributions of Peru and Colombia and underestimates those of Venezuela, Argentina, and Mexico. In particular, the contribution of zero by Mexico exemplifies the problem described previously.

The estimators provide similar results in terms of best hedges and the accompanying risk reductions. In this case, credit risk can be reduced by taking smaller or even short positions in the debt of the listed obligors, which might be achieved by entering into a total return swap, for example.

The eight largest contributors to 99% ES (Table 27.7) are almost identical to those for 99% VaR, with the exception that Morocco replaces Mexico. This implies that credit losses from Moroccan debt are more concentrated in the extreme tail (beyond the 99% quantile) of the portfolio loss distribution than those from Mexican debt. The relative contributions of the other seven obligors are virtually identical for 99% VaR and 99% ES. The differences between the UECV and HD estimates are negligible for ES since both estimators average multiple order statistics. The hedging strategy for 99% ES is also similar to that for 99%

Table 27.7. *99% ES analytics for bond portfolio.*

Obligor	Current value ($\times 10^6$ USD)	ES contrib. (%)		$1 Marginal ES (USD)		Best hedge position ($\times 10^6$ USD)		ES reduction (%)	
		UECV	HD	UECV	HD	UECV	HD	UECV	HD
Brazil	894	25.11	25.05	0.37	0.37	−5,120	−5,174	42	42
Russia	758	19.87	19.83	0.35	0.35	−7,231	−7,231	35	35
Venezuela	414	13.15	13.15	0.42	0.42	−1,774	−1,776	33	33
Argentina	636	11.25	11.52	0.23	0.24	−6,553	−6,492	25	25
Peru	279	8.52	8.40	0.40	0.40	−2,050	−2,052	26	26
Colombia	608	4.98	5.02	0.11	0.11	−27,390	−27,562	21	21
Morroco	131	1.60	1.61	0.16	0.16	−11,542	−11,570	22	22
Russia CCC	44	1.48	1.47	0.44	0.44	−943	−945	25	25

VaR; although the best hedge positions tend to be of slightly larger magnitude (i.e., more negative) for 99% ES, the potential risk reductions are essentially the same in both cases.

Since the risk contribution equals the product of the one-dollar marginal 99% ES and the position size, plotting these two values for each obligor provides a visual representation of the sources of risk (Figure 27.8). In a portfolio that is well balanced in terms of risk, large positions should have a small marginal risk and vice versa. Thus, significant contributors to the portfolio risk can be identified by their distance from the axes. In Figure 27.8, for example, the six largest contributors to the 99% ES are clearly outliers. Note also that Vietnam's negative marginal ES indicates that this obligor is actually hedging risk in this case.

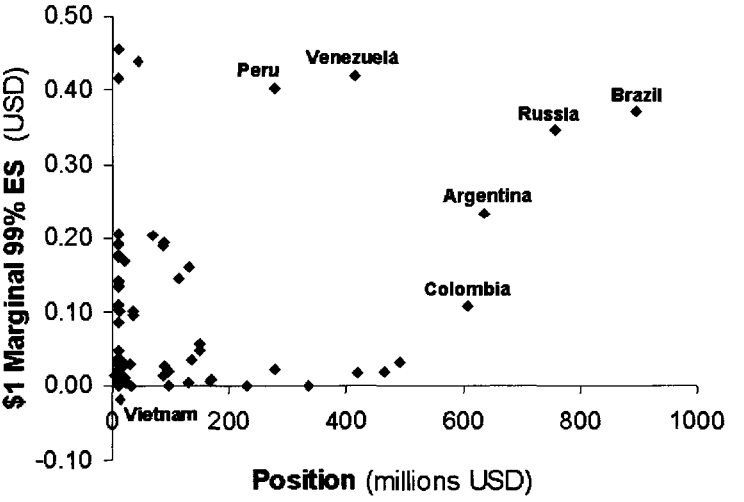


Figure 27.8. *Position size versus one-dollar marginal 99% ES for bond portfolio.*

Thus far, we have considered only the risk of the portfolio and not its potential return. For example, adopting the best hedge position of −5,120 million USD for Brazil reduces the 99% ES by 42% to 767 million USD. (Our ensuing discussion will deal exclusively with the UECV estimator.) However, the resulting expected return of the portfolio is only 1.79%, which is less than the risk-free rate.

By modifying multiple positions concurrently, optimization models allow greater risk reductions to be achieved without compromising return. Let us minimize the 99% ES while maintaining an expected return of $R = 7.26\%$ and limiting the current value of each obligor's debt to be at least zero (i.e., no short positions) and no more than one-fifth of the total portfolio value. In this case, solving (27.29) yields an optimal portfolio with a 99% ES of 264 million USD, thus reducing risk by 80% while maintaining the current expected return.

The loss distribution of the optimal portfolio (Figure 27.9) is markedly different from that of the original portfolio (Figure 27.6). The right tail becomes much shorter, and there is a greater likelihood of achieving large positive returns (represented in Figure 27.9 by negative losses). The optimal portfolio eliminates the positions in the seven of the eight contributors listed in Table 27.7 (only a small amount of Colombian debt is retained) and

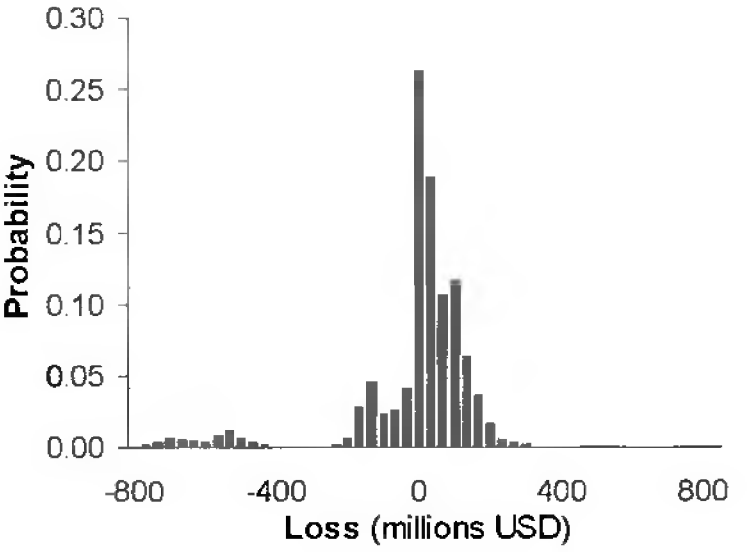


Figure 27.9. *Distribution of losses for optimized 99% ES bond portfolio* (20,000 *scenarios*).

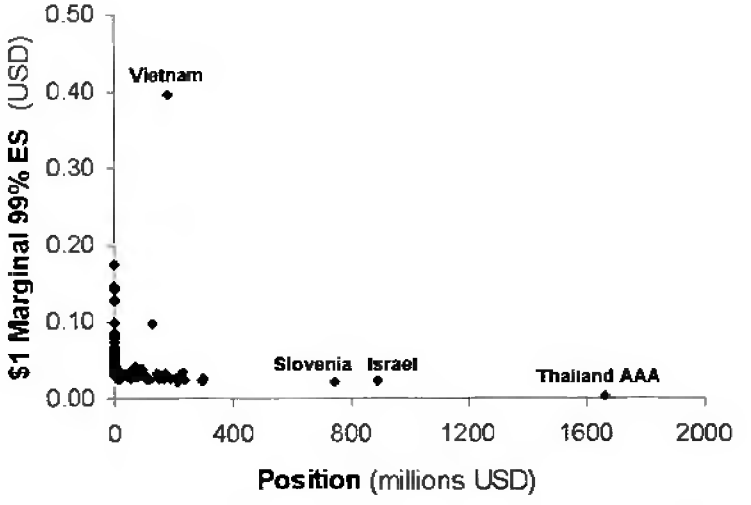


Figure 27.10. *Position size versus one-dollar marginal 99% ES for optimized bond portfolio.*

increases the exposure to Vietnam, which offers a high expected return. While Vietnam acts as a hedge in the original portfolio, it now becomes the primary source of risk, contributing approximately one-quarter of the 99% ES. In general, the risk contributions are much more evenly distributed among the obligors in the optimal portfolio (Figure 27.10) than in the original portfolio (Figure 27.8). (For simplicity, Figure 27.10 reports the average of the

marginal risk in the positive and negative directions.) For example, the largest position (Thailand AAA) constitutes 20% of the portfolio's value yet contributes only 2% of the risk.

Finding portfolios that minimize VaR is computationally intractable in this case. (Observe that (27.29) requires 20,000 binary variables.) However, Figure 27.9 motivates a heuristic approach for this problem. Minimizing ES, or alternatively, regret, effectively reshapes the loss distribution, and shortening the right tail generally (although not always) has the desirable effect of also reducing the VaR. In particular, the tail of the loss distribution is affected by the quantile or threshold used in the ES or regret models, respectively. Since (27.29) and (27.30) are linear programs, they can easily be solved repeatedly for various quantiles or thresholds in an effort to obtain a portfolio with a low VaR.

This approach of using ES or regret as a proxy for VaR is discussed in more detail in [35] and [27]. The first paper also provides a simple heuristic procedure for improving the VaR of an existing portfolio: We eliminate from consideration all tail scenarios (i.e., those corresponding to order statistics beyond the UECV estimate of VaR) and then minimize the maximum loss of the remaining scenarios. The resulting minimax problem is a linear program, and its optimal value represents the VaR as estimated by the UECV estimator.

Let us now compare the efficient portfolios for ES, regret, and variance in terms of their return and risk, as measured by 99% VaR (Figure 27.11). The ES frontier represents the lowest VaR among portfolios that are obtained by solving (27.29) for quantiles of 95%, 96%, 97%, 98%, 99%, and 99.9%, while those for regret are obtained by solving (27.30) for thresholds of (in hundreds of millions) 100, 200, 300, 400, 500, 600, and 700. The VaR-minimizing heuristic described above is applied to the portfolio with the lowest VaR among the optimal ES and regret portfolios.

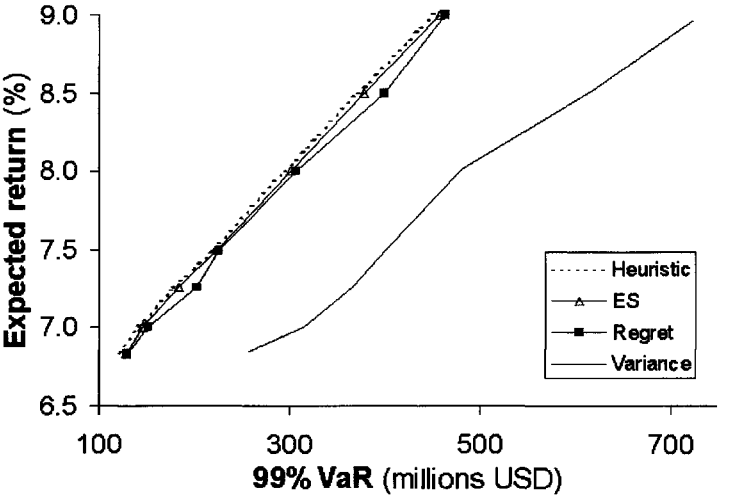


Figure 27.11. *Efficient frontiers for 99% VaR.*

In this case, the 99% VaR can be reduced by more than 80% (to 179 million USD) while maintaining an expected return of 7.26%. Note that the optimal mean variance portfolios are clearly inefficient with respect to 99% VaR, as expected given the nonnormality of the credit

loss distributions. For example, a mean variance efficient portfolio holding 300 million USD of capital against VaR at the 99% level returns less than 7%. In contrast, the ES and regret efficient portfolios holding a similar amount of capital return 8%, an improvement of more than 100 basis points.

27.7 Conclusions

Various measures exist for quantifying the risk of a portfolio based on its loss distribution. In addition to volatility, VaR has become a de facto standard for defining economic and regulatory capital; ES represents a coherent alternative to both volatility and VaR. While the accurate measurement of risk remains a critical requirement for financial institutions, effective risk management also requires tools for identifying the sources of risk in a portfolio and for implementing positional changes that mitigate this risk. Risk managers can beneficially apply a range of techniques in this regard; simple analytics that derive from an instrument's trade risk profile provide guidance when adjusting a single position, while stochastic optimization models can motivate changes on a broader scale.

If losses are normally distributed (with mean zero), VaR and ES are constant multiples of volatility, which simplifies the computation of risk analytics. When the normality assumption is invalid, as often occurs in practice, a scenario-based approach provides an effective alternative for risk managers. Although simulating the portfolio over a set of scenarios is computationally demanding, scenario-based risk management tools can be implemented with moderate effort beyond that required for measuring risk.

Quantile-based measures are more appropriate than volatility for quantifying risk when losses are not normally distributed. The method used to estimate quantiles of the loss distribution can be of significance when calculating risk analytics under the scenario-based approach. In particular, by using L-estimators, more reliable estimates of the marginal VaR and VaR contribution are obtained since the quantile is computed as a weighted average of multiple order statistics.

Given a sufficient number of scenarios, the choice of estimator generally has minimal impact on the quantile estimate itself, and thus estimates that are based on a single order statistic are acceptable when formulating optimization models. This has a significant impact on the solvability of risk management models in practice. For instance, ES can be optimized using linear programming, while VaR requires solving an integer program. We have shown that minimizing ES or regret for various quantiles or thresholds, respectively, yields portfolios that exhibit a low VaR, and this represents an effective heuristic approach for the VaR minimization problem.

Bibliography

[1] C. Acerbi, C. Nordio, and C. Sirtori, *Expected Shortfall as a Tool for Financial Risk Management*, Technical Report, Abaxbank, Milan, 2001.

[2] C. Acerbi and D. Tasche, *On the Coherence of Expected Shortfall*, Technical report, Technische Universität München, Munich, 2001.

[3] S. AGUAIS AND D. ROSEN, *Credit Risk: Enterprise Credit Risk Using Mark-to-Future*, Algorithmics, Toronto, 2001.

[4] F. ANDERSSON, H. MAUSSER, D. ROSEN, AND S. URYASEV, *Credit risk optimization with conditional value-at-risk criterion*, Math. Program. B, 89 (2001), pp. 273–291.

[5] P. ARTZNER, F. DELBAEN, J.-M. EBER, AND D. HEATH, *Coherent measures of risk*, Math. Finance, 9 (1999), pp. 203–228.

[6] N. BUCAY AND D. ROSEN, *Credit risk of an international bond portfolio: A case study*, ALGO Res. Quart., 2 (1999), pp. 9–29.

[7] D. CARIÑO, T. KENT, D. MYERS, C. STACY, M. SYLVANUS, A. TURNER, K. WATANABE, AND W. T. ZIEMBA, *The Russell-Yasuda Kasai model: An asset/liability model for a Japanese insurance company using multistage stochastic programming*, Interfaces, 24 (1994), pp. 29–49.

[8] D. CARIÑO AND W. T. ZIEMBA, *Formulation of the Russell Yasuda Kasai financial planning model*, Oper. Res., 46 (1998), pp. 433–449.

[9] R. DEMBO, *Scenario optimization*, Ann. Oper. Res., 30 (1991), pp. 63–80.

[10] R. DEMBO, A. AZIZ, D. ROSEN, AND M. ZERBS, *Mark-to-Future: A Framework for Measuring Risk and Reward*, Algorithmics, Toronto, 2000.

[11] R. DEMBO AND A. KING, *Tracking models and the optimal regret distribution in asset allocation*, Appl. Stoch. Models Data Anal., 8 (1992), pp. 151–157.

[12] R. DEMBO AND D. ROSEN, *The practice of portfolio replication: A practical overview of forward and inverse problems*, Ann. Oper. Res., 85 (1999), pp. 267–284.

[13] T. DIELMAN, C. LOWRY, AND R. PFAFFENBERGER, *A comparison of quantile estimators*, Comm. Statist. Simulation, 23 (1994), pp. 355–371.

[14] K. DOWD, *VaR by increments*, Risk, 11 (1998), pp. 31–32.

[15] P. EMBRECHTS, S. RESNICK, AND G. SAMORODNITSKY, *Living on the edge*, Risk, 11 (1998), pp. 96–100.

[16] M. GARMAN, *Improving on VaR*, Risk, 9 (1996), pp. 61–63.

[17] C. GOURIEROUX, J. LAURENT, AND O. SCAILLET, *Sensitivity analysis of values at risk*, J. Empirical Finance, 7 (2000), pp. 225–245.

[18] W. HALLERBACH, *Decomposing Value-at-Risk: A General Analysis*, Technical Report, Tinbergen Institute, Rotterdam, The Netherlands, 1999.

[19] F. HARRELL AND C. DAVIS, *A new distribution-free quantile estimator*, Biometrika, 69 (1982), pp. 635–640.

[20] R. HUGHEY, *A survey and comparison of methods for estimating extreme right tail-area quantiles*, Commun. Statist. Theory Methods, 20 (1991), pp. 1463–1496.

[21] N. JOBST AND S. ZENIOS, *The tail that wags the dog: Integrating credit risk in asset portfolios*, J. Risk Finance, 3 (2001), pp. 31–43.

[22] P. JORION, *Risk^2: Measuring the risk in value at risk*, Financial Anal. J., 52 (1996), pp. 47–56.

[23] P. JORION, *Value at Risk*, 2nd ed., McGraw–Hill, New York, 2001.

[24] JP MORGAN, INC., *Introduction to RiskMetrics*, 4th ed., JP Morgan, Inc., New York, 1995.

[25] S. KEALHOFER, *Managing default risk in portfolios of derivatives*, in Derivative Credit Risk, Risk Publications, London, 1996.

[26] H. KONNO AND H. YAMAZAKI, *Mean-absolute deviation portfolio optimization model and its application to Tokyo stock market*, Management Sci., 37 (1991), pp. 519–531.

[27] N. LARSEN, H. MAUSSER, AND S. URYASEV, *Algorithms for optimization of value-at-risk*, in Financial Engineering, E-Commerce and Supply Chain, P. Pardalos and V. Tsitsiringos, eds., Kluwer Academic Publishers, Norwell, MA, 2002, pp. 129–157.

[28] R. LITTERMAN, *Hot spots and hedges*, J. Portfolio Management, Special Issue (1996), pp. 52–75.

[29] H. MARKOWITZ, Portfolio selection, J. Finance, 7 (1952), pp. 77–91.

[30] H. MAUSSER, *Calculating quantile-based risk analytics with L-estimators*, ALGO Res. Quart., 4 (2001), pp. 33–47.

[31] H. MAUSSER AND D. ROSEN, *Beyond VaR: From measuring risk to managing risk*, ALGO Res. Quart., 1 (1998), pp. 5–20.

[32] H. MAUSSER AND D. ROSEN, *Marginal VaR Analysis: Parametric and Non-Parametric Methods*, Technical Report, Algorithmics, Toronto, 1998.

[33] H. MAUSSER AND D. ROSEN, *Beyond VaR: Triangular risk decomposition*, ALGO Res. Quart., 2 (1999), pp. 31–43.

[34] H. MAUSSER AND D. ROSEN, *Applying scenario optimization to portfolio credit risk*, ALGO Res. Quart., 2 (1999), pp. 19–33.

[35] H. MAUSSER AND D. ROSEN, *Efficient frontiers for credit risk*, ALGO Res. Quart., 2 (1999), pp. 35–48.

[36] H. MAUSSER AND D. ROSEN, *Managing risk with expected shortfall*, in Probabilistic Constrained Optimization: Methodology and Applications, S. Uryasev, ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 204–225.

[37] N. PATEL, *PortfolioRisk+ cracks tail risk conundrum*, Risk, 15 (2002), p. 10.

[38] G. PFLUG, *Some remarks on the Value-at-Risk and the conditional value-at-risk*, in Probabilistic Constrained Optimization: Methodology and Applications, S. Uryasev, ed., Kluwer Academic Publishers, Norwell, MA, 2000, pp. 272–281.

[39] D. REYNOLDS AND D. SYER, *A general simulation framework for operational loss distributions*, in Operational Risk: Regulation, Analysis and Management, C. Alexander, ed., Prentice-Hall, Englewood Cliffs, NJ, 2003.

[40] R. T. ROCKAFELLAR AND S. URYASEV, *Optimization of conditional value at risk*, J. Risk, 2 (2000), pp. 21–41.

[41] R. T. ROCKAFELLAR AND S. URYASEV, *Conditional Value-at-Risk for General Loss Distributions*, Technical Report 2001-5, Department of Industrial and Systems Engineering, University of Florida, Gainsville, FL, 2001.

[42] O. SCAILLET, *Nonparametric Estimation and Sensitivity of Expected Shortfall*, Technical Report, Catholic University of Louvain, Louvain-la-Neuve, Belgium, 2000.

[43] W. F. SHARPE, *Capital asset prices: A theory of market equilibrium under conditions of risk*, J. Finance, 19 (1964), pp. 425–442.

[44] S. SHEATHER AND J. MARRON, *Kernel quantile estimators*, J. Amer. Statist. Assoc., 85 (1990), pp. 410–416.

[45] D.TASCHE, *Conditional Expectation as Quantile Derivative*, Technical Report, Technische Universität München, Munich, 2000.

[46] D. TASCHE, *Expected Shortfall and Beyond*, Technical Report, Technische Universität München, Munich, 2002.

[47] R. ZAGST AND J. KEHRBAUM, *Portfolio Optimization under Limited Value at Risk*, Technical Report, RiskLab GmbH, Munich, 1998.

[48] S. ZENIOS, ED., *Financial Optimization*, Cambridge University Press, Cambridge, UK, 1993.

[49] S. ZENIOS AND P. KANG, *Mean-absolute deviation portfolio optimization for mortgage-backed securities*, Ann. Oper. Res., 45 (1993), pp. 443–450.

[50] W. ZIEMBA AND J. MULVEY, EDS., *Worldwide Asset and Liability Modeling*, Cambridge University Press, Cambridge, UK, 1998.

Chapter 28

Price Protection Strategies for an Oil Company

E. A. Medova and A. Sembos†*

28.1 Introduction

In this chapter we attempt to link the results of our research into logistics planning for a consortium of oil companies facing uncertain demands and prices [2, 3] with our current work on risk management and real options evaluation. The logistics work has been carried out in the European Community ESPRIT project Hydrocarbon and Chemical Logistics under Uncertainty via Stochastic Optimization (HChLOUSO). Logistics planning deals with supply, transformation, storage, and transportation activities in a complex network structure over various planning horizons (strategic decisions for the long term and tactical decisions for the medium term). As a feasible solution to such problems is seldom initially achieved, it is common practice in industry to search for a solution by minimizing the cost of infeasibilities and correspondingly adjusting constraints [6]. When internal resources of companies are exhausted (or in surplus), excess demand (or supply) may be handled externally by buying (or selling) the required products in the spot commodity markets.

In our ESPRIT project we observed the importance of trading activities and proposed elimination of infeasibilities through trading. This problem has been formulated as a dynamic stochastic program with additional variables representing the trading activities and leads to robust first-stage solutions in the presence of future price and demand uncertainties [2].

The volatility of crude oil prices has a significant impact on the planning decisions and budgets of oil companies. It is common for producing companies to develop a *hedging program* to insure that the company is protected against a collapse in crude oil prices. Here we integrate such financial planning with logistics planning and illustrate the importance of this

*Cambridge Systems Associates Limited and Centre for Financial Research, Judge Institute of Management, University of Cambridge, Cambridge, UK (eam28@cam.ac.uk).

†Credit Suisse First Boston, London, UK (asembos@hotmail.com).

integration with the example of the Metallgesellschaft financial collapse. This spectacular loss of nearly $1.3 billion is usually presented as a derivatives failure and is studied by risk managers [1, 4, 5, 10]. It is still debatable whether it was "unhedgeable risks, poor hedging strategy, or just bad luck" [5] or was it simply speculation on the derivative markets? To answer this question we look at the Metallgesellschaft case as a stochastic optimization problem. Analysis of the MG Refining and Marketing (MGRM) case points to where the deficiencies in hedging occurred and leads to the formulation of an integrated logistics and financial planning problem.

A brief description of oil commodity markets and related traded financial instruments is given in section 28.2, where the volatilities of oil price products are illustrated graphically. The case of Metallgesellschaft and in particular its "synthetic storage" derivative strategy is analyzed in section 28.3. Section 28.4 proposes and analyzes a stochastic programming framework for formulation of a hedging policy. In section 28.5 a strategic corporate plan integrating operational and financial decisions is proposed. The firm's operations can be thought of as a collection of real assets generating positive/negative cash flows. Therefore the hedging policy provides real and financial flexibility as a part of a single strategy given by the solution of the stochastic programming model formulated. Implementation issues and numerical results for selected problem instances are also discussed in this section. Section 28.6 concludes.

28.2 Volatility of oil markets

As oil prices play a significant role in the planning decisions of oil companies, it is worth examining their behavior over time. Until the late 1960s the price of oil—crude oil and petroleum products—was relatively stable and most oil companies entered into long-term agreements with the oil producing countries to satisfy their needs. However, the formation of OPEC (Organization of Petroleum Exporting Countries) in the 1960s marked a new era for oil prices. OPEC countries now produce around 40% of the world's crude oil and their oil exports represent about 60% of the oil traded internationally. By controlling supply OPEC greatly influences oil prices.

Figure 28.1 shows the price of crude oil from 1949 to 2002. The graph shows a steady increase in the crude oil price until 1973. Between 1973 and 1974, the oil price increased suddenly from $2.90 to $12.00 per barrel due to an oil embargo resulting from changes in resource ownership. Between 1974 and 1978, oil prices continued to rise more gradually. The Iranian crisis of 1978 and 1979 caused a sudden increase in price from $12 to $30 per barrel. OPEC's decision to increase production of crude oil in 1986 caused a sharp decline to $12. The 1991 Gulf War is also easily identified on this historical picture of crude oil prices. All these political and economic factors have exposed companies to significant price risk. The price of crude oil in mid 2000 of more than $32 exceeded the level of 1979 and has become a significant factor in economic policy to date.

We analyze the bear market of 1992–1995, which was marked by the MGRM affair. Most definitions of *hedging* as a means of protection against losses are very generic and actually lead to a variety of different strategies implemented by companies. Generally, hedging has led to the emergence of an oil derivatives market and a variety of hedging instruments—forwards, futures, options, and swaps with different maturities. Oil futures

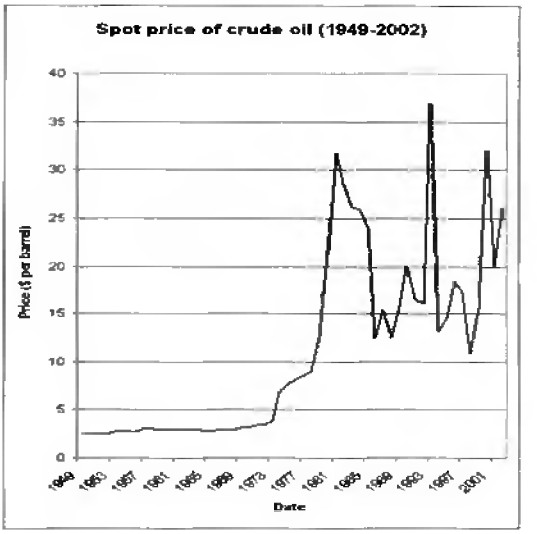
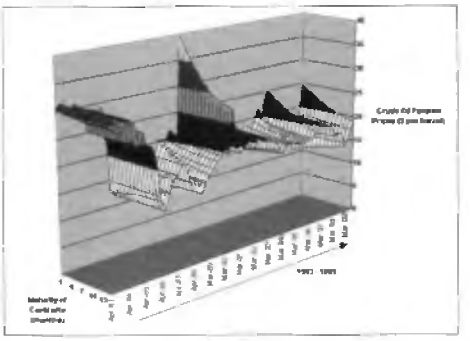


Figure 28.1. *Spot price of crude oil.*

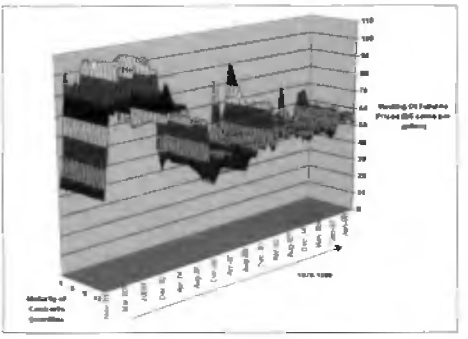
are among the most actively traded futures in the world. West Texas Intermediate (WTI) crude oil, heating oil #2, unleaded gasoline, and natural gas are all traded on the New York Mercantile Exchange (NYMEX), and Brent crude and gas oil are traded on London's International Petroleum Exchange (IPE). The futures data presented in Figures 28.2(a), (b), and (c) show, respectively, 1-month, 2-month, ..., 15-month crude oil futures from 1983 to 1999, heating oil #2 futures from 1978 to 1999, and unleaded gasoline futures from 1984 to 1999 (all traded on the NYMEX). These data were constructed from daily data by extracting all Fridays to yield weekly frequency, estimating all missing entries using linear interpolation.

The relations between spot and future prices define market conditions. The market is in *backwardation* when futures prices are below the spot price and in *contango* when futures prices are above the spot price. For commodities such as oil products which incur significant costs of physical storage over time, normal market conditions lead to backwardation.

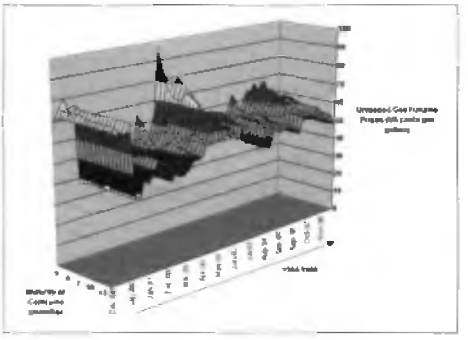
Oil companies usually enter forward supply contracts which commit them to supplying final product volumes, such as heating oil and unleaded gas, to end users at specific future dates at fixed prices. The companies also enter forward contracts with their suppliers of crude oil which commit them to buying crude oil at specific time points at fixed prices. If crude oil prices fall below the fixed price specified in a forward-supply contract, then the company finds itself in the unprofitable position of being contracted to buy crude oil at prices above spot prices. Similarly, if oil product prices rise above the fixed price specified in the contracts, then the company is in the position of having to sell its products at prices below spot prices at a loss of potential profit. Of course, when crude oil prices move appropriately the company may also find itself in a profitable position. In an ideal situation, one can achieve a so-called perfect hedge, which completely eliminates the risk associated with a future commitment to deliver by taking an equal and opposite position in the futures



(a) Crude oil futures prices.



(b) Heating oil #2 futures prices.



(c) Unleaded gas futures prices.

Figure 28.2. *Futures prices of crude oil and petroleum products.*

market. However, this strategy implies the existence of futures contracts that match *exactly* the supply commitments. Depending on the maturities of forward contracts, the availability of matching futures varies. Even today, there is no oil futures contract with maturity greater than 36 months (i.e., 3 years) ahead. Alternatively, using merely long-term *over-the-counter* (OTC)—i.e., tailor-made—derivatives might expose the company to great credit risk—setting aside the difficulty of finding an appropriate counterparty in the first place, given the illiquidity of long-term OTC derivatives markets. Choosing an appropriate hedging strategy to reduce price risk is a complex practical problem that needs careful consideration. It is important to examine the state of the futures markets when rebalancing any hedging portfolio and when deciding on the optimal *hedge ratio*. We consider below situations when the market is either in backwardation or contango together with the effects of these states on the performance of the hedging strategy.

Existing complex hedging strategies of large oil companies transform the company's complete dependence on spot oil prices into a variety of exposures to forward, futures, option, and swap markets. Hedging programs generate either gains or losses compared with the strategy of not hedging at all. Use of hedging to generate profits is sometimes perceived as speculation. A recent RISK volume on crude oil hedging [8] starts with the simple message, "there is no consistent easy way to obtain speculative profits from trading in crude oil financial markets." This difference in perception of hedging and speculation can be resolved by recourse to the company objectives.

We assume that the prime objective of hedging is to reduce potential cash-flow losses over a sufficiently long-term planning period. The same could be said regarding other activities of an oil company resulting in consistent successful strategies for the company as an energy purchaser, energy transformer, or energy producer.

28.3 The case of Metallgesellschaft

The debate over MGRM's hedging program has been mainly carried out in the risk management literature with an emphasis on the use of derivatives and stemming in part from different assumptions about the goals of the hedging program [10].

In 1993–1994, MG Corporation (the U.S. subsidiary of Metallgesellschaft AG) lost more than $1.3 billion on its positions in energy futures and swaps when prices for crude oil, heating oil, and gasoline fell sharply. Some reports characterized MGRM's oil trading activities as "a game of roulette," but another view is that its derivatives activities were in fact part of a complex strategy—a fully integrated oil business in the United States. MGRM's efforts to develop a fully integrated oil business in the United States are witnessed by the following facts [10]:

- Long-term customer relationships were based on forward-supply contracts involving approximately 160 million barrels of gasoline and heating oil over 10 years at fixed prices of $3 or $5 a barrel higher than spot prices with a cash-out option for counterparties.

- MGRM acquired a 49% interest in Castle Energy, a U.S. oil exploration company which then became an oil refiner.

- To ensure a supply of energy products MGRM purchased Castle's entire output of refined products (about 126,000 barrels a day) at guaranteed margins for up to 10 years into the future.

- MGRM set out to develop an infrastructure to support the storage and transportation of various oil products.

Prior to a discussion of this fully integrated strategy we focus on MGRM's financial activities and on one aspect of its derivatives strategy termed "synthetic storage" [1, 4, 5, 9]. The term *synthetic storage* means that the company holds oil derivatives rather than physically storing oil. Synthetic storage is beneficial to producers who in appropriate market conditions can achieve the lower costs of storage embodied in futures prices rather than paying their own actual costs of physical storage.

MGRM maintained a *one-to-one hedge*, which means that MGRM's total derivatives position was equal to its forward-supply commitments, i.e., 160 million barrels initially. The hedging portfolio consisted of short-dated energy futures—1-month or 2-month futures contracts with underlying products being WTI crude oil, heating oil, and gasoline, traded on the NYMEX—and 3-month OTC swaps (receiving finished product and paying crude), in the proportions $33\frac{1}{3}\%$ and $66\frac{2}{3}\%$, respectively. Given their long-term forward supply commitments and their hedging portfolio of short-term derivatives, MGRM followed a *stack and roll* or synthetic storage strategy. More particularly, 4 days before expiration MGRM would close out its positions in the near-month futures and buy new futures contracts. On each settlement date, the total position in futures and swaps was reduced by the amount of product delivered to end users during that period as part of the forward-supply agreements, thus always maintaining a one-to-one hedge. At the end of 1993, MGRM reported large losses on its positions in futures and swaps. As a result of a fall in oil prices, margin calls on its futures positions and losses on the rollover costs of maturing futures and swap positions were incurred. Unfortunately for MGRM the oil futures market was in contango for most of 1993, which meant much larger costs than normal when it rolled its derivatives positions forward. Mathematical formulation of MGRM's hedging strategy clarifies the pitfalls encountered.

28.3.1 Mathematical formulation of MGRM's strategy

In general, a synthetic storage strategy may be formulated as a solution of a *dynamic stochastic program* [2]. The objective and constraints corresponding to MGRM's hedging strategy [1] are given below.

Sets

$P = \{$ products $\} = \{$ crude oil, heating oil, gasoline $\}$,

$F = \{$ monthly futures $\}$,

$S = \{$ OTC three month swaps $\}$,

$D = \{$ derivatives $\} := F \cup S$,

$D^h = \{d \in D : d$ is used to hedge exposure to heating oil prices $\}$,

$D^g = \{d \in D : d$ is used to hedge exposure to unleaded gasoline prices $\}$,

$T = \{$ time periods $\}$.

Parameters

$\text{demand}_{p,t}$: fixed demand for product p in period t (in fact, the volume of the supply commitment for time period t);

$p^c_{p,t}$: cash inflow upon (i.e., price of) selling a unit of product p in period t (set in the forward-supply agreements);

$\mathit{underlying}_d$: the underlying product of derivative d;

$\mathit{maturity}_d$: the maturity of derivative d;

$\mathbf{spot}_{d,t}$: spot price of the underlying product of derivative d in period t;

$\mathbf{f}_{d,t}$: price of derivative contract d in period t.

We assume that $\mathbf{f}_{d,t}$ is the random price of derivative d per unit of underlying product when it is purchased by MGRM in period t. (Throughout the paper random variables are denoted by boldface type.) We also assume that when derivative d is sold by the firm in period t its value per unit of underlying product is the spot price of the product. Thus bid-ask spread and other minor frictions are ignored for simplicity.

Variables

$V^+ := \{V^+_{d,t} : d \in D,\ t \in T\}$, where $V^+_{d,t}$ is the volume of derivative d purchased at the end of time period t—(purchasing);

$V := \{V_{d,s,t} : d \in D, s, t \in T,\ s \le t\}$, where $V_{d,s,t}$ is the volume of derivative d purchased during time period s and held during time period t—(holding);

$V^- := \{V^-_{d,s,t} : d \in D, s, t \in T,\ s \le t\}$, where $V^-_{d,s,t}$ is the volume of derivative d purchased at the end of time period s and sold at the end of time period t—(selling).

Objective function

The objective of MGRM was to maximize its profit over the 10 years of its forward-supply contracts. MGRM's *operating revenue* was given by its income from these fixed-supply contracts. However, the cost/profit inherent in its hedging strategy needs to be taken into account in deriving profit over the (actual 10-year) planning period. We assume a zero

interest rate for simplicity of expression; this assumption is easily relaxed. The objective is to *maximize*

$$\text{Expected Profit} = \underbrace{\sum_{p\in P}\sum_{t\in T} p^{c}_{p,t} \times \text{demand}_{p,t}}_{\substack{\text{total cash inflow from forward-supply}\\ \text{contracts (assuming zero interest rate)}}}$$

$$- \underbrace{\mathbb{E}\sum_{d\in D}\sum_{t\in T} \mathbf{f}_{d,t} \times \mathbf{V}^{+}_{d,t}}_{\substack{\text{total cash outflow upon purchasing}\\ \text{derivatives } d \text{ in period } t}}$$

$$+ \underbrace{\mathbb{E}\sum_{d\in D}\sum_{t\in T}\sum_{s=1}^{t} \mathbf{spot}_{d,t} \times \mathbf{V}^{-}_{d,s,t}}_{\substack{\text{total cash inflow upon selling in period } t\\ \text{derivatives } d \text{ bought in period } s \le t}} \tag{28.1}$$

Constraints

All constraints are assumed to hold *almost surely* (a.s.), i.e., with probability one.

One-to-one hedge. MGRM's total derivatives position was at all times equal to its forward-supply commitments:

$$\underbrace{\sum_{d\in D^{h}}\sum_{s=1}^{t} \mathbf{V}_{d,s,t}}_{\substack{\text{volume of heating}\\ \text{oil derivatives held}\\ \text{in time period } t}} = \underbrace{\sum_{t_1=t+1}^{T_{\text{plan}}} \text{demand}_{\text{h. oil},t_1}}_{\substack{\text{total remaining demand}\\ \text{for heating oil from time}\\ \text{period } t \text{ onward}}} \quad \forall\, t \in T, \tag{28.2}$$

$$\underbrace{\sum_{d\in D^{g}}\sum_{s=1}^{t} \mathbf{V}_{d,s,t}}_{\substack{\text{volume of gasoline}\\ \text{derivatives held}\\ \text{in time period } t}} = \underbrace{\sum_{t_1=t+1}^{T_{\text{plan}}} \text{demand}_{\text{gasoline},t_1}}_{\substack{\text{total remaining demand}\\ \text{for gasoline from time}\\ \text{period } t \text{ onward}}} \quad \forall\, t \in T. \tag{28.3}$$

Rollover. All positions in futures are closed out just before expiration because MGRM used futures only for hedging purposes and in favorable market circumstances never intended to take delivery and pass on to customers the product underlying the contracts.

$$\underbrace{\sum_{t=s}^{s+\text{maturity}(d)} \mathbf{V}^{-}_{d,s,t}}_{\substack{\text{volume of futures } d \\ \text{purchased in period } s \text{ and} \\ \text{sold just before maturity}}} := \underbrace{\mathbf{V}^{-}_{d,t}}_{\substack{\text{volume of futures } d \\ \text{sold at the end of} \\ \text{period } t}} \quad \forall\, (d,\ t) \in D \times T. \tag{28.4}$$

Derivatives purchase inventory balance. The derivatives bought must be added to the hedging portfolio

$$\mathbf{V}_{d,t,t} = \mathbf{V}^{+}_{d,t} \qquad \forall\, (d,\ t) \in D \times T. \tag{28.5}$$

Derivatives sale inventory balance. The derivatives sold must be removed from the hedging portfolio

$$\mathbf{V}_{d,s,t} = \mathbf{V}_{d,s,t-1} - \mathbf{V}^{-}_{d,s,t} \qquad \forall\, d \in D,\ \forall s, t \in T :\ s \leq t. \tag{28.6}$$

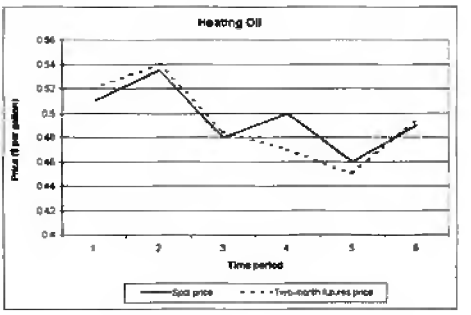
Composition of the hedging portfolio. MGRM's position in futures and swaps accounted for $33\frac{1}{3}\%$ and $66\frac{2}{3}\%$ of the hedging portfolio, respectively:

$$\underbrace{\frac{1}{3}\sum_{d\in D}\sum_{s=1}^{t} \mathbf{V}_{d,s,t}}_{\substack{\frac{1}{3} \text{ of the total derivatives} \\ \text{volume held during period } t}} = \underbrace{\sum_{d\in F}\sum_{s=1}^{t} \mathbf{V}_{d,s,t}}_{\substack{\text{total futures volume} \\ \text{held in period } t}} \quad \forall\, t \in T. \tag{28.7}$$

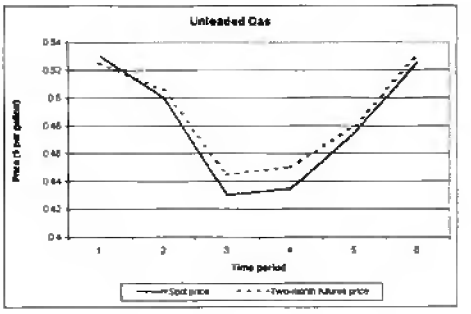
28.3.2 Analysis of MGRM's hedging strategy

A small deterministic example—i.e., one possible scenario of the above dynamic stochastic program—is presented here to illustrate the nature and weaknesses of MGRM's hedging model. For simplicity we assume that the hedging portfolio consists of futures only. This example involves two usable products—heating oil and gasoline (products 1 and 2)—and six time periods, each of which corresponds to two months. The company enters forward-supply contracts with its customers, and therefore the demand for its products is fixed for the six periods. The demand for heating oil and unleaded gas in periods 1, 2, 3, and 4 equals 0, but it is nonzero for time periods 5 and 6 ($\text{demand}_{1,5} = 700{,}000$, $\text{demand}_{1,6} = 650{,}000$, $\text{demand}_{2,5} = 500{,}000$, $\text{demand}_{2,6} = 450{,}000$). The price specified in the fixed-supply contracts is close to the spot price of the products when the contracts are entered into. More particularly, the price of heating oil equals 53 cents per gallon and unleaded gas is 52 cents per gallon.

The hedging portfolio consists of two-month futures, with underlying products heating oil and gasoline, which are rolled forward just before expiration (i.e., near the end of the second month). Assume that the price of the two-month futures contracts and the spot



(a) Heating oil.



(b) Unleaded gas.

Figure 28.3. *Spot prices and two-month futures prices for heating oil and unleaded gas.*

Table 28.1. *Scenario of spot prices and two-month futures prices (in U.S. dollars per gallon) for heating oil and unleaded gas.*

Period	Heating oil		Unleaded gasoline	
	Spot price	Two-month futures	Spot price	Two-month futures
1	0.51	0.52	0.53	0.525
2	0.535	0.54	0.50	0.505
3	0.48	0.485	0.43	0.445
4	0.50	0.47	0.435	0.45
5	0.46	0.45	0.475	0.48
6	0.49	0.495	0.525	0.53

price of the corresponding underlying products take the values illustrated in Figure 28.3 and summarized in Table 28.1.

The market for heating oil is in contango for the first three periods and in the last period—since the spot price of heating oil is less than the two-month heating oil futures

Table 28.2. *Values attached to decision variables.*

Time period	Purchasing	Holding	Selling
1	$V^+_{1,1} = 1,350,000$ $V^+_{2,1} = 950,000$	$V_{1,1,1} = 1,350,000$ $V_{2,1,1} = 950,000$	
2	$V^+_{1,2} = 1,350,000$ $V^+_{2,2} = 950,000$	$V_{1,2,2} = 1,350,000$ $V_{2,1,2} = 950,000$	$V^-_{1,1,2} = 1,350,000$ $V^-_{2,1,2} = 950,000$
3	$V^+_{1,3} = 1,350,000$ $V^+_{2,3} = 950,000$	$V_{1,3,3} = 1,350,000$ $V_{2,3,3} = 950,000$	$V^-_{1,2,3} = 1,350,000$ $V^-_{2,2,3} = 950,000$
4	$V^+_{1,4} = 1,350,000$ $V^+_{2,4} = 950,000$	$V_{1,4,4} = 1,350,000$ $V_{2,4,4} = 950,000$	$V^-_{1,3,4} = 1,350,000$ $V^-_{2,3,4} = 950,000$
5	$V^+_{1,5} = 650,000$ $V^+_{2,5} = 450,000$	$V_{1,5,5} = 650,000$ $V_{2,5,5} = 450,000$	$V^-_{1,4,5} = 1,350,000$ $V^-_{2,4,5} = 950,000$
6			$V^-_{1,5,6} = 650,000$ $V^-_{2,5,6} = 450,000$

price—and in backwardation in periods 4 and 5. The market for unleaded gas is in backwardation in the first period and in contango in the remaining five periods. The state of the market will determine the scale of the rollover costs.

The financial operations in each period corresponding to the implementation of MGRM's hedging strategy are summarized in Table 28.2. All scenario-dependent decision variables not mentioned in the table have value 0. The portfolio hedges the forward-supply commitments gallon for gallon at all times. The volume of heating oil that the company is contracted to supply over the six time periods amounts to 1,350,000 gallons, 700,000 of which will be delivered in time period 5 and the remaining 650,000 gallons in time period 6. The heating oil futures held during the first four time periods correspond to 1,350,000 gallons; in period 5 the position is reduced by the 700,000 gallons supplied to customers. The remaining positions in heating oil futures are closed out after satisfaction of the heating oil supply contract in the last period. A similar situation holds for the position in unleaded gas futures. Since forward-supply commitments are hedged with two-month—one-period—futures, the position in maturing contracts is closed at the end of each period and "new" two-month futures are bought to maintain a one-to-one hedge. However, by selling maturing contracts at the spot price of the underlying product and by buying two-month futures at the forward price of the underlying, rollover costs/benefits are incurred depending on whether the market is in contango or backwardation.

For our six-period example the rollover costs are detailed in Figure 28.4. Leaving aside the cost of establishing the initial position in futures in time period 1, each of the outflows outweighs inflows in the first three time periods. This is because the markets for heating oil and unleaded gas are in contango in periods 2 and 3, so rolling future contracts forward is costly. In time periods 4 and 5 the market for heating oil is in backwardation, and the benefits from rolling these futures contracts forward outweigh the costs of rolling gasoline futures forward. Moreover, in period 5 the position in futures is reduced so that outflows are reduced further. Due to the fall in the price of heating oil over the five periods, however, the cash inflow received in time period 5 is less than the outflow incurred when

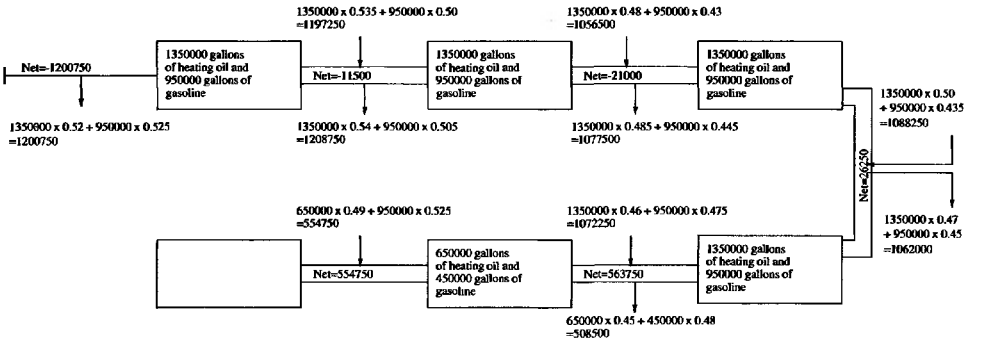


Figure 28.4. *Costs and revenues per period in the implementation of MGRM's hedging strategy.*

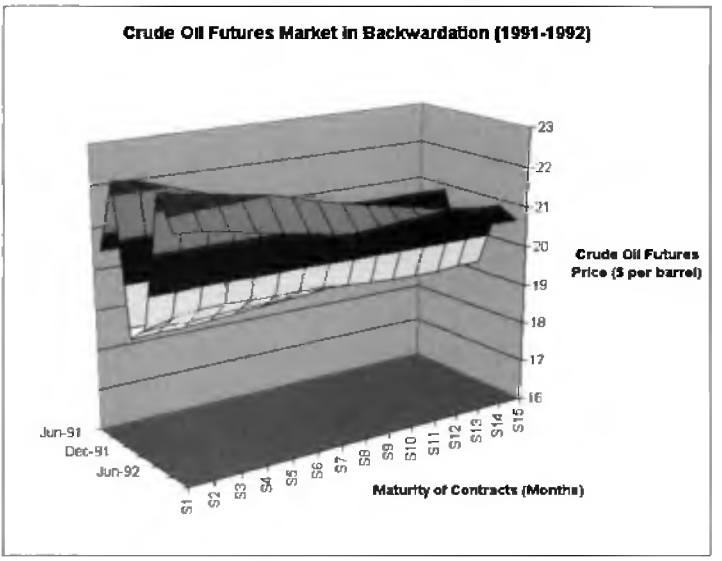


Figure 28.5. *Backwardation price relationship in the crude oil futures market and rising crude oil prices during* 1992.

the initial position was established in time period 1. Finally, all the remaining positions in futures are closed out in time period 6.

MGRM's hedging strategy is not flexible and, while price scenario-dependent, is not adjustable to market conditions. It corresponds to the scenario-dependent solution of the optimization problem specified by the fixed right-hand side (equality) constraints (28.2)–(28.5) presented in section 28.3.1. Its actual hedging strategy was thus fixed by the solution of a deterministic problem given by the observed relations between the spot and future prices—at the time in backwardation (Figure 28.5).

Our small example shows the scale of rollover costs during the first three time periods

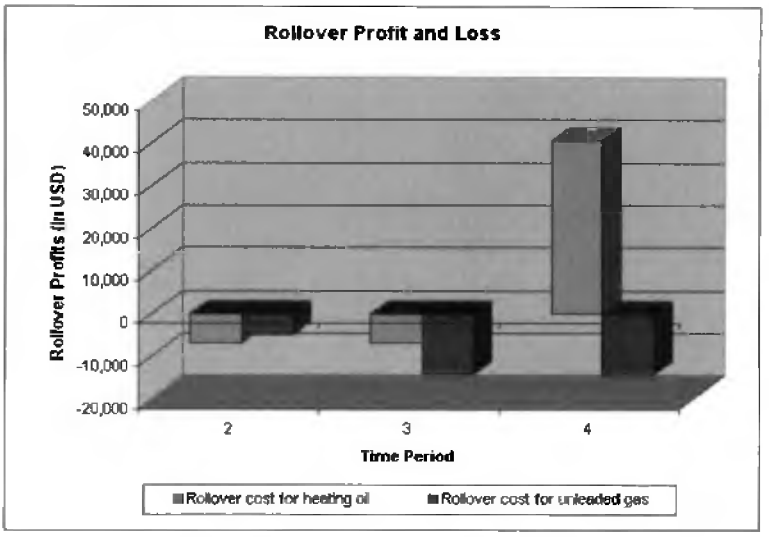


Figure 28.6. *Rollover costs during time periods* 2, 3, *and* 4 *in the implementation of MGRM's strategy.*

when the market was in contango (see Figure 28.6). When MGRM's strategy was put into practice in 1993, the company's forward-supply commitments and the corresponding derivatives positions spanned a horizon of 10 years, during which prices could move in any direction and the prevailing market state (i.e., contango or backwardation) could be reversed. Although oil markets are usually in backwardation, there have been extended periods during which they were in contango. In fact, oil prices fell sharply from the end of 1992, and oil futures markets were in a contango price relationship for the whole of 1993 (see Figure 28.7). MGRM's nonadjustable hedging strategy turned out to be very costly—at least in the short run.

Rollover risk should be taken into account in designing a hedging strategy which should be flexible enough to allow the minimization of rollover losses when the market is in contango. Furthermore, the stacking strategy used by MGRM was characterized by cash flow asymmetry over time between futures outflows and inflows from forward-supply commitments. This is also clear in our small example, where the long-term contracts produce cash inflow only during the last two periods; see Figure 28.8.

For the above simplified hedging program the total outflows from buying futures and rolling them forward together with the total inflows from the forward-supply contracts during the six time periods provide a profit of $112,100,000.

In retrospect, if we consider MGRM's situation over the 10 years of its contractual arrangements, then it would probably have broken even or made a profit. In the actual situation, margin calls due to the futures position mark-to-market in a regime of falling prices, as well as the credit risk associated with long-term forward contracts, proved to be critical to MGRM—especially when management decided to liquidate the hedge prematurely.

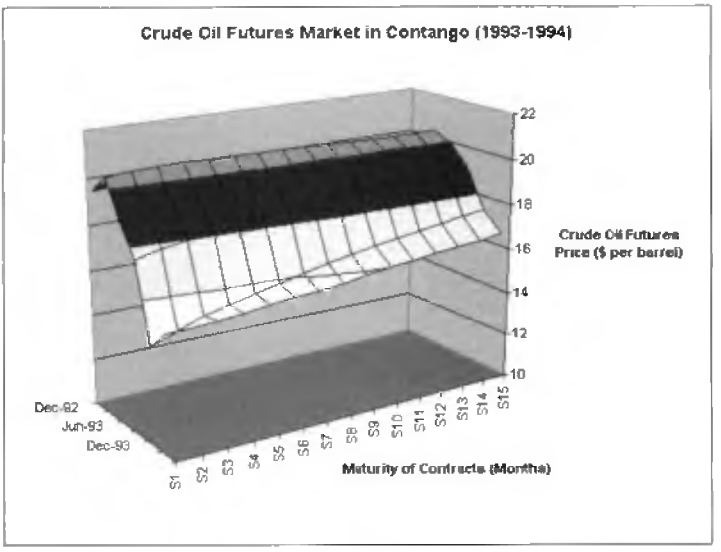


Figure 28.7. *Contango price relationship in the crude oil futures market and falling crude oil prices during* 1993.

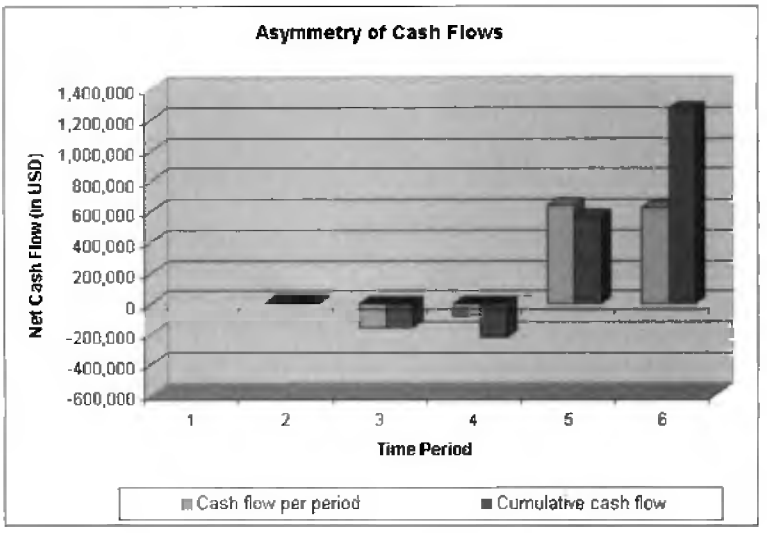


Figure 28.8. *Net cash flow per period and cumulative cash flow in the implementation of MGRM's strategy.*

28.4 Stochastic programming formulation of a hedging policy

The stochastic programming formulation of hedging provides a dynamic financial analysis technique for identification of the optimal policy under the uncertainty involved in future price movements. The hedging oil price exposure (HOPE) model [13] has been formulated to minimize the losses incurred from the fixed contracts when the market moves in an unfavorable direction. The major drawback of MGRM's strategy can be improved upon by allowing a flexible hedge ratio to be determined optimally. The resulting decisions will depend on the stochastic market environment given by a number of scenarios for the prices of oil products as compared with the fixed prices specified in the company's forward-supply commitments. Hedging will occur only when the market is expected to move in an unfavorable direction, and the hedging policy determined will react to the expected future market evolution as determined by the scenario tree for the relevant price processes.

Simulation of spot and derivative oil prices is a complex task which is only briefly outlined here (for details, see [3] and [12]). A two-factor model [11] has been chosen for representation of each stochastic price process. These factors are the spot price of the oil product and its instantaneous convenience yield. Forecasting of oil product prices is complicated by the fact that the only observable processes are those for futures contracts—one-month, three-month, five-month, seven-month, and nine-month. We have analyzed oil product futures prices provided by the NYMEX. Given historical data on futures prices of crude oil or oil products—one product at a time—over a specified period, the Kalman filtering technique is used to calculate the parameters of the stochastic data processes generating the spot price and convenience yield over the same period. The data paths generated for the spot prices of the different oil products—crude oil, heating oil, and unleaded gas—and their convenience yields are then used to calculate the correlations between them. Then multiple scenarios for spot and futures prices of each product are generated simultaneously using Monte Carlo simulation.

Model implementation involves the conditional generation of a scenario tree sample from the stochastic data process, and model solution determines an optimal decision process over these scenarios. We employ the notation of section 28.3.1 and consider a vector stochastic data process

$$\omega = \{\mathbf{f}_{p,d,m,t} : p \in P,\ d \in D,\ m = 0, \ldots, mat_d,\ t = 1, \ldots, T_{\text{plan}}\}$$

and a vector decision process

$$\mathbf{x} = \{\mathbf{x}_t := (\mathbf{V}^-_{c_{d,s,t}}, \mathbf{V}_{c_{d,s,t}}, \mathbf{V}^+_{c_{d,t}}, \mathbf{V}^+_{f_{d,t}}, \mathbf{V}_{f_{d,s,t}}, \mathbf{V}^-_{f_{d,s,t}}, \mathbf{C}_t) : d \in D,\ s \in T,\ t = s, \ldots, T_{\text{plan}}\},$$

where

$\mathbf{f}_{p,d,m}$ denotes the futures price of product p underlying contract d, with m periods to maturity ($m = 0$ denotes the spot price of product p),

$\mathbf{V}^-_c$ denotes crude oil short-sold,

$\mathbf{V}_c$ denotes crude oil owned,

$\mathbf{V}^+_c$ denotes crude oil bought back,

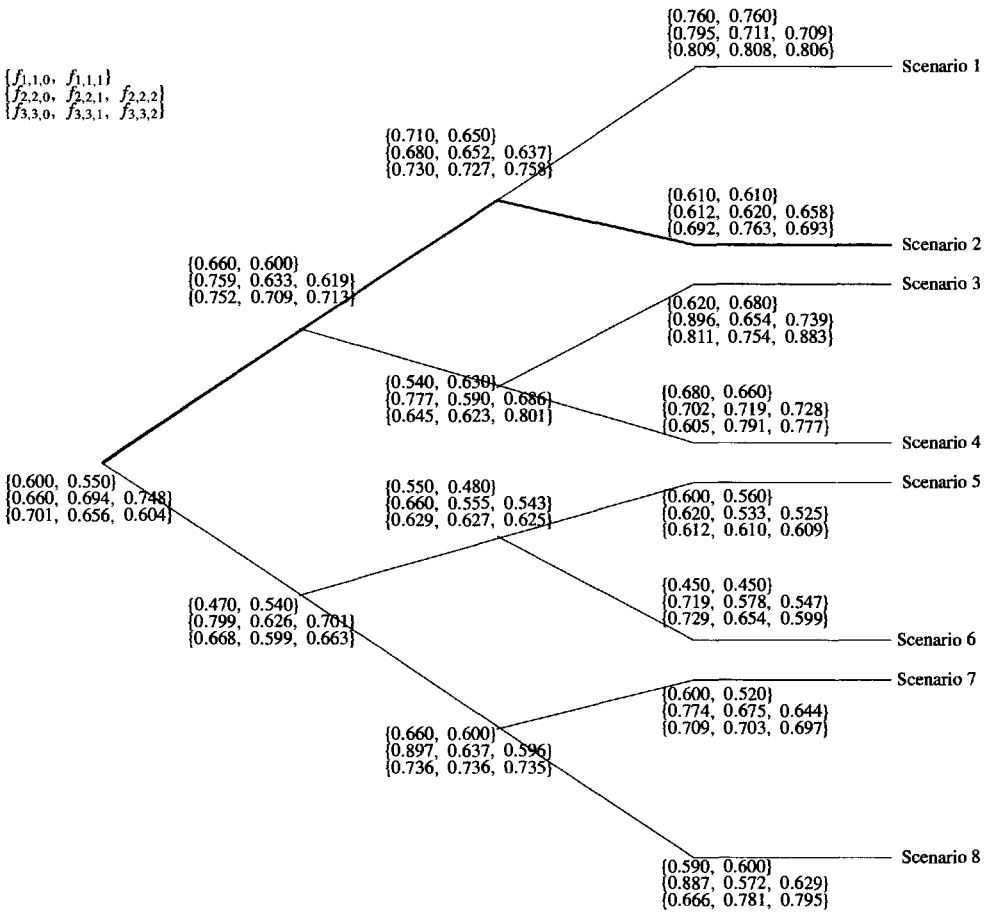


Figure 28.9. *Scenario tree for the evolution of prices of crude oil, heating oil, and unleaded gas.*

$\mathbf{V}_f^+$ denotes final product purchased,

$\mathbf{V}_f$ denotes final product held,

$\mathbf{V}_f^+$ denotes final product sold,

C denotes cash balance that maps a data trajectory to a vector decision trajectory.

Recall that boldface denotes random variables or processes.

The HOPE model is the full implementation of the dynamic stochastic programming model of section 28.3.1. Its deterministic equivalent was formulated and solved using commercial modeling language and linear programming software (AMPL, Stochgen 2.1, and CPLEX 7.0). See [13] for more details. The HOPE model was first run on a small example with a scenario tree for eight equiprobable four period scenarios for the evolution of crude oil, unleaded gas, and heating oil spot and futures prices, as shown in Figure 28.9.

This simple stochastic problem formulation of a hedging policy provides a less profitable solution than the average of the solutions of the individual deterministic scenarios (the distribution problem solution), as shown in Table 28.3. The hedging policy of this solution, however, is robust against all given combinations of market prices and cannot "see the future" as in the scenario analysis of the distribution problem. This latter wait-and-see problem thus overestimates profits (see [2]). For computational results and analysis of larger instances of the HOPE model with up to 65,536 scenarios, see [13].

Table 28.3. *Objective function values in the implementation of the HOPE model.*

Eight-scenario problem	Distribution problem	Multistage stochastic
Cash balance at the end of planning period	(average over eight scenarios) 18,358.8	7,430.5

28.5 Integrated corporate policy via dynamic stochastic programming

Current use of financial derivatives to replicate their production and logistics operations has come to be an acceptable practice for energy companies. However, Enron's spectacular collapse in late 2001 illustrates the danger of the replacement of traditional assets and activities with financial ones. Enron's strategy undermined the asset-owning business on which the company had been built and was largely concerned with creating an "asset-light" firm, i.e., an aggressive trading business, as promoted by its chief executive, J. Skilling (see [7]).

In our formulation we return attention to the real activities of the firm and ask two main questions:

- Is the goal of strategic corporate planning to coordinate real and financial activities while maximizing profit?
- Should physical activities such as production, storage, and transportation be used to fulfill forward-supply contracts when market conditions are such that the current hedging program (involving financial contracts) suffers losses?

Integration of the real and financial activities of a company requires an understanding of and expertise in both logistics and financial planning. Dynamic stochastic programming provides techniques for solving the strategic logistics planning problem as well as the detailed analysis of financial strategy and a hedging policy (see [2, 12, 13]).

28.5.1 Discussion of the DROP logistics/production model

In our previous work [2] we observed that trading products at spot prices in local markets may be viewed as a simple hedging policy: losses or opportunity costs due to lack of

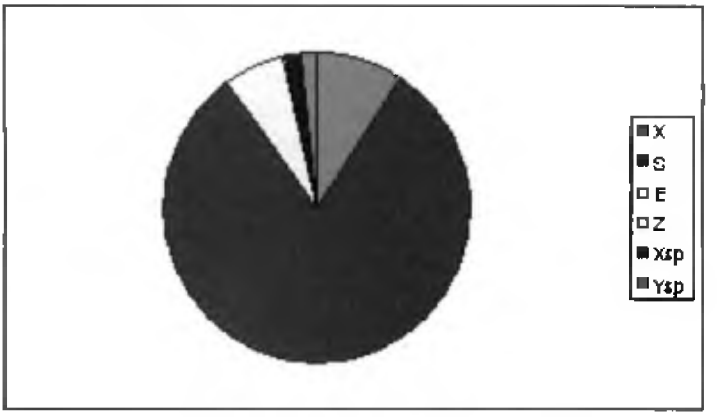


Figure 28.10. *Case* 1: *Storage is the dominating activity.*

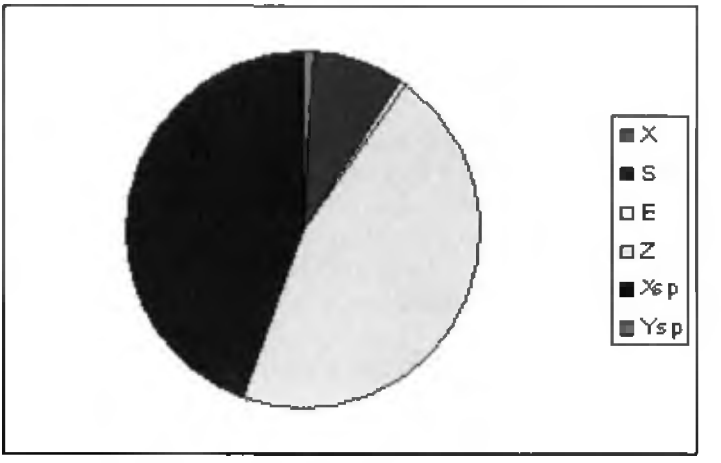


Figure 28.11. *Case* 2: *Refining is the dominating activity.*

internal resources or surplus positions are covered by trading. In our depot and refinery optimization problem (DROP) model—developed in the course of the HChLOUSO project [2, 6, 12, 13]—we optimize the cost of major activities over a specified time horizon and reach feasibility using product market trading at spot prices. Analysis of the totality of optimal decisions for the various oil products in the different locations—the amounts to supply, to store, to transport, to refine, and to buy or to sell: X, S, E, Z, X^{sp}, Y^{sp}—identified cases which had one or another activity absent but was replaced by amounts traded on the spot markets. (The DROP model does not allow purchase and sale of oil products in the same markets, i.e., it rules out pure trading.)

Figures 28.10 and 28.11 illustrate the overall (i.e., across products by value) solutions corresponding to two different cases of spot oil prices for the same problem.

Figure 28.10 shows that the oil company satisfies most of its needs for crude oil by entering contractual agreements with potential suppliers—operators. The amount of crude oil supplied by operators is represented by the variable X. The proportion of crude oil supplied through contracts X is greater than the proportion of crude oil bought in the spot market X^{sp}. As a result, the oil company may need to store the supplied crude oil for an extended period before refining it and distributing the product to its customers. The great amount of storage needed is apparent on the diagram—storage S is the dominating activity. Moreover, if we compare the amount of storage needed with the amount of products refined Z, we see that the company undertakes a lot of effort storing products rather than producing them. Figure 28.11 illustrates a case in which it is preferable for the company to satisfy its crude oil needs by buying crude oil in the spot market—spot market purchases are represented by X^{sp}. The amount supplied through contractual agreements X is limited. As a result, the oil company does not need to store crude oil for extended periods of time and hence the storage volume S is now considerably limited. By contrast with the first case, the refining activity—represented by Z—becomes the dominating activity. The two cases differ in the emphasis they give to spot market transactions. In the second case price movements encourage the company to perform most of its transactions in the spot markets.

Existence of financial options and other derivatives that mimic real activities requires inclusion of derivative instruments in the strategic planning problem. In formulating an integrated corporate policy we assume that a variety of derivatives contracts are available on the market. The choice of hedging instruments at each point in time will depend on market conditions and the expected future market evolution. The stochastic program itself is able to choose the instrument mix optimally without any further specification. As we have seen, the volume of spot supplies relative to the volume supplied through forward contracts will have a direct effect on the scale of other activities such as storage and refining—and hence on costs.

28.5.2 Mathematical formulation of an integrated corporate planning model

The goal for a corporate planning (CORPLAN) model is to maximize an oil company's profit in a self-financing fashion while satisfying demand over a specified planning period.

The demand for final oil products is assumed to be stochastic and consists partly of forward contracts, which commit the oil company to selling volumes of final oil products to external parties as well as to satisfying market demand for final oil products. The forward contracts are assumed to be fixed before the beginning of the planning period, so they constitute the deterministic part of the total demand volume, the remaining market demand being stochastic. The oil company is also assumed to have entered forward contracts with suppliers of crude oil, which commit it to buying volumes of crude oil at a fixed price at specific points in time. This volume can be supplemented with more crude oil, which the company can buy from the spot market; however, the company is prohibited from selling crude oil in the spot market.

The mathematical representation of our model will include all decisions and constraints of the logistics model [2] and all decisions and constraints of the financial hedging model of section 28.4. The notation related to logistic activities is summarized below:

$X_{n,o,p,t}$ is the product p volume supplied by operator o to node n during time period t with corresponding cost $c^x_{n,p,t}$.

$Z_{r,f,t}$ is the product volume for the refining process f that takes place in refinery r during time period t and its corresponding cost $c^z_{r,f,t}$.

$E_{i,j,p,m,t}$ is the volume of product p whose transportation from transportation network node i to node j by transport means m starts during time period t with cost $c^e_{i,j,p,m,t}$.

$S_{n,p,t}$ is the volume of stock of product p at node n at the end of time period t and the cost of its storage $c^s_{n,p,t}$.

$X^{\text{spot}}_{n,p,t}$ is the spot purchase volume and $Y^{\text{spot}}_{n,p,t}$ is the spot selling volume of product p at node n during time period t with corresponding cost/price $c^{\text{spot}}_{n,p,t}$ or $p^{\text{spot}}_{n,p,t}$.

The definition of the sets needed to specify the subscript ranges of these variables and the underlying geographical transportation network is given in the appendix. Notation for financial decisions and parameters is the same as in the hedging models of sections 28.3.1 and 28.4.

For simplicity we specify here only the *deterministic*—i.e., one scenario for all stochastic prices and demands—version of the model. (See [13] for details of the stochastic version.)

The objective function *maximizes* the cash balance at the end of the planning period, $C_{T_{\text{plan}}}$, where the C_t, $t = 1, \ldots, T_{\text{plan}}$, include the profits/losses obtained from the trading of derivative instruments and of final products on the spot markets. The total profit or cash flow at the end of the planning period consists of the profit from logistic operations and the profit from trading operations including the profits/losses obtained from the use of derivative instruments. Again we abstract from reality for simplicity by assuming a zero interest rate to avoid discounting or compounding cash flows. A simulated market interest rate process can be included to relax this assumption.

The objective function is

$$\text{Max} \quad \text{Terminal Wealth } C_{T_{\text{plan}}}$$

$$= \underbrace{\sum_{t\in T}\sum_{p\in P} p^{\dot{c}}_{p,t} \times \text{demand}_{p,t}}_{\substack{\text{total inflow from forward-supply} \\ \text{contracts (assuming interest is zero)}}}$$

$$- \sum_{n\in N}\sum_{p\in P_1} c^x_{n,p,t} \times X_{n,p,t} \quad \text{SUPPLY}$$

$$- \sum_{r\in R}\sum_{f\in F_r}\sum t \in T c^z_{r,f,t} \times Z_{r,f,t} \quad \text{REFINING}$$

$$- \sum_{p\in P}\sum_{m\in M}\sum_{(i,j)\in L_m}\sum_{t\in T} c^e_{i,j,p,m,t} \times E_{i,j,p,m,t} \quad \text{TRANSPORTATION}$$

$$
\begin{aligned}
&- \sum_{n\in N}\sum_{p\in P}\sum_{t\in T} c^{s}_{n,p,t} \times S_{n,p,t} \quad \text{STOCK}\\
&- \sum_{n\in N}\sum_{p\in P_1}\sum_{t\in T} c^{\text{spot}}_{n,p,t} \times X^{\text{spot}}_{n,p,t} \quad \text{SPOT PURCHASE}\\
&+ \sum_{n\in N}\sum_{p\in P_2}\sum_{t\in T} p^{\text{spot}}_{n,p,t} \times Y^{\text{spot}}_{n,p,t} \quad \text{SPOT SALE}\\
&- \sum_{d\in D}\sum_{t\in T} p^{+}_{d,t} \times V^{+}_{d,t} \quad \text{DERIVATIVE PURCHASE}\\
&+ \sum_{d\in D}\sum_{t\in T}\sum_{s=1}^{t} p^{-}_{d,s,t} \times V^{-}_{d,s,t} \quad \text{DERIVATIVE SALE}\\
&+ \sum_{d\in D}\sum_{t=s}^{s+\text{maturity}(d)} r_{d,s,t} \times V_{d,s,t} \text{ POSITIVE MARK-TO-MARKET}\\
&- \sum_{d\in D}\sum_{t=s}^{s+\text{maturity}(d)} s_{d,s,t} \times V_{d,s,t} \text{ NEGATIVE MARK-TO-MARKET.} \qquad (28.8)
\end{aligned}
$$

The constraints imposed on the variables are of four types: logical constraints, product balance constraints, capacity constraints, and financial constraints. These are described below in terms of the set definitions in the appendix.

Logical constraints

- All variables must be nonnegative, since they represent product volumes.
- Physical supply of final products (from third-party contracts) is not allowed.

$$X_{n,o,p,t} = 0 \quad \forall\, n, o, p, t \in N \times O \times P_2 \times T, \qquad (28.9)$$

$$X^{\text{spot}}_{n,p,t} = 0 \quad \forall\, n, p, t \in N \times P_2 \times T. \qquad (28.10)$$

- Selling crude oil products in spot markets is not allowed.

$$Y^{\text{spot}}_{n,p,t} = 0 \quad \forall\, n, p, t \in N \times P_1 \times T. \qquad (28.11)$$

- Transformation of products can take place only at refineries.

$$Z_{n,f,t} = 0 \quad \forall\, n \in N : n \notin R, \ \forall\, f \in F, \ \forall\, t \in T. \qquad (28.12)$$

- Transformation of products can take place at a particular refinery only if the required technology is available at the refinery.

$$Z_{n,f,t} = 0 \quad \forall\, n \in N, \ \forall\, f \notin F_n, \ \forall\, t \in T. \qquad (28.13)$$

- A product can be given as input to a refinery for transformation by a particular technology only if the technology needs the product as an input.

$$Z^p_{n,p,f,t} = 0 \quad \forall n \in N, \ \forall p \in P_1, \ \forall f \notin F^{\text{in}}_p, \ \forall t \in T. \tag{28.14}$$

- A product will be given as output by a transformation technology only if the technology produces that product as an output.

$$Z^p_{n,p,f,t} = 0 \quad \forall n \in N, \ \forall p \in P_2, \ \forall f \notin F^{\text{out}}_p, \ \forall t \in T. \tag{28.15}$$

- Transportation of products between a pair of nodes can exist only if there is a link between the nodes.

$$E_{i,j,p,m,t} = 0 \quad \forall i, j \in N : (i,j) \notin L_m, \ \forall p \in P, \ \forall m \in M, \ \forall t \in T. \tag{28.16}$$

- Transportation of a product between a pair of nodes by a particular transportation means can exist only if that product can be transported by the particular means.

$$E_{i,j,p,m,t} = 0 \quad \forall i, j \in N, \ \forall p \notin P_m, \ \forall m \in M, \ \forall t \in T. \tag{28.17}$$

- Transportation from one node to another can be started only if it is allowed to be started during the time period of interest.

$$E_{i,j,p,m,t} = 0 \quad \forall i, j \in N, \ \forall p \in P_m, \ \forall m \in M, \ \forall t \notin T_{i,j,p,m}. \tag{28.18}$$

Product balance constraints

- During each time interval the overall volume set aside for transportation or refining does not exceed resources.

$$\begin{aligned} &\sum_{f \in F_{n_1}: n_1 \in R} Z^p_{n_1,p,f,t} + \sum_{n_2 \in N: \ (n_1,n_2) \in L_m} \sum_{m \in M: p \in P_m} E_{n_1,n_2,p,m,t} \\ &\quad \le S_{n_1,p,t-1} + \sum_{o \in O} X_{n_1,o,p,t} + X^{\text{spot}}_{n_1,p,t} \quad \forall n_1, \ p, \ t \in N \times P_1 \times T. \end{aligned} \tag{28.19}$$

- Stock at the end of the previous time period is sufficient to satisfy distribution and demand requirements during the current time period.

$$\begin{aligned} S_{n_1,p,t-1} \ge &\sum_{n_2 \in N: \ (n_1,n_1) \in L_m} \sum_{m \in M: p \in P_m} E_{n_1,n_2,p,m,t} + \sum_{o \in O} d_{n_1,o,p,t} + Y^{\text{spot}}_{n_1,p,t} \\ &\forall n_1, \ p, \ t \in N \times P_2 \times T. \end{aligned} \tag{28.20}$$

- Each technology uses the correct mixture of crude oils and produces the correct proportion of final products.

$$Z^p_{n_1,p,f,t} = g_{f,p} Z_{n_1,f,t} \quad \forall n_1, \ p, \ f, t \in N \times P_1 \times F_{n_1} \times T. \tag{28.21}$$

- Product balance equation for input products (i.e., crude oils):

$$
\begin{aligned}
& S_{n_1,p,t-1} && \text{stock of product } p \text{ in node } n_1 \text{ at the beginning of time period } t \\
+ & \sum_{o \in O} X_{n_1,o,p,t} + X^{\text{spot}}_{n_1,p,t} && \text{supply of product } p \text{ to node } n_1 \text{ during time period } t \\
+ & \underbrace{\sum_{m \in M:\ p \in P_m} \ \sum_{n_2 \in N:\ (n_2,n_1) \in L_m} E_{n_2,n_1,p,m,t-\Delta_{n_2 n_1 pm}}}_{\text{total volume of product } p \text{ which arrives at node } n_1 \text{ from other nodes during time period } t} + e_{n_1,p,t} \\
- & \sum_{f \in F_{n_1} \cap F^{in}_p:\ n_1 \in R} Z^p_{n_1,p,f,t} && \text{volume of product } p \text{ that is given as input to the refinery during time period } t \\
- & \underbrace{\sum_{m \in M:\ p \in P_m} \ \sum_{n_2 \in N:\ (n_1,n_2) \in L_m} E_{n_1,n_2,p,m,t}}_{\text{total volume of product } p \text{ whose transportation from node } n_1 \text{ to another node starts during time period } t} \\
= & \ S_{n_1,p,t} && \text{stock of product } p \text{ in node } n_1 \text{ at the end of time period } t
\end{aligned}
\tag{28.22}
$$

$$\forall\, n_1, p, t \in N \times P_1 \times T.$$

- Product balance equation for output products (i.e., final products):

$$
\begin{aligned}
& S_{n_1,p,t-1} && \text{stock of product } p \text{ in node } n_1 \text{ at the beginning of time period } t \\
+ & \underbrace{\sum_{m \in M:\ p \in P_m} \ \sum_{n_2 \in N:\ (n_2,n_1) \in L_m} E_{n_2,n_1,p,m,t-\Delta_{n_2 n_1 pm}}}_{\text{total volume of product } p \text{ which arrives at node } n_1 \text{ from another node during time period } t} + e_{n_1,p,t} \\
+ & \sum_{f \in F_{n_1} \cap F^{out}_p:\ n_1 \in R} Z^p_{n_1,p,f,t} && \text{volume of product } p \text{ that is given as output by the refinery during time period } t \\
- & \underbrace{\sum_{m \in M:\ p \in P_m} \ \sum_{n_2 \in N:\ (n_1,n_2) \in L_m} E_{n_1,n_2,p,m,t}}_{\text{total volume of product } p \text{ whose transportation from node } n_1 \text{ to another node starts during time period } t}
\end{aligned}
$$

$$
\begin{aligned}
&- \sum_{o \in O} d_{n_1,o,p,t} - Y^{\text{spot}}_{n_1,p,t} && \text{volume of product } p \text{ provided to satisfy demand in node } n_1 \text{ during time period } t + \text{excess product } p \text{ volume sold in the spot market} \\
&= S_{n_1,p,t} && \text{stock of product } p \text{ in node } n_1 \text{ at the end of time period } t
\end{aligned} \tag{28.23}
$$

$$\forall\, n_1, p, t \in N \times P_2 \times T.$$

Capacity constraints

- Pipeline capacity bounds:

$$\sum_{t \in T^U_u} \sum_{p \in P_m : t \in T_{n_1,n_2,p,m}} E_{n\hat{1},n\hat{2},p,m,t} \leq e^c_{n1,n2,m,u} \tag{28.24}$$

$$\forall (n\hat{1}, n\hat{2}) \in L_m,\ \forall m \in M^c,\ \forall u \in U.$$

- Tanker, ship, or wagon capacity bounds:

$$\sum_{p \in P_m} \sum_{(n_1,n_2) \in L_m} \sum_{t' \in T^t_{n_1,n_2,p,m}} E_{n_1,n_2,p,m,t'} \leq e^d_{m,t} \tag{28.25}$$

$$\forall m \in M^d,\ \forall t \in T,$$

where $\begin{cases} T^t_{n_1,n_2,p,m} := \{t - \Delta_{n_1,n_2,p,m}, \ldots, t\} \text{ if } \Delta_{n_1,n_2,p,m} < t, \\ T^t_{n_1,n_2,p,m} := \{1, \ldots, t\} \text{ otherwise.} \end{cases}$

- Operator supply contract bounds:

$$\underline{x}_{n,o,p,v} \leq \sum_{t \in T^V_v} X_{n,o,p,v} \leq \overline{x}_{n,o,p,v} \quad \forall\ n, o, p, v \in N \times O \times P \times V. \tag{28.26}$$

- Refinery capacity bounds:

$$\underline{z}_{n,f,h} \leq \sum_{t \in T^H_h} Z_{n,f,t} \leq \overline{z}_{n,f,h} \quad \forall\ n, f, h \in N \times F_n \times H. \tag{28.27}$$

In addition to the above constraint, there is an upper bound on the transformation capacity of each technology accounted for as follows:

$$Z_{n,f,t} \leq \overline{z}^T_{n,f,t} \quad \forall\ n, f, t \in N \times F_n \times T. \tag{28.28}$$

- Storage capacity bounds:

$$\underline{s}_{n,p,t} \leq S_{n,p,t} \leq \overline{s}_{n,p,t}, \quad \forall\ n, p, t \in N \times P \times T. \tag{28.29}$$

Financial constraints

- Hedging exposure to crude oil prices:

$$\underbrace{\sum_{d\in D^c}\sum_{s=1}^{t} V_{d,s,t}}_{\substack{\text{volume of derivatives with underlying product crude oil added to the portfolio in period } t}} \leq \underbrace{\sum_{t_1=t+1}^{T_{\text{plan}}}\sum_{p\in P_1} \text{supplied}_{p,t_1}}_{\substack{\text{total remaining supply of crude oil from time period } t \text{ onward}}} \quad \forall\, t \in T. \tag{28.30}$$

- Hedging exposure to final oil prices:

$$\underbrace{\sum_{d\in D^f}\sum_{s=1}^{t} V_{d,s,t}}_{\substack{\text{volume of derivatives with underlying final product added to the portfolio in period } t}} \leq \underbrace{\sum_{t_1=t+1}^{T_{\text{plan}}}\sum_{p\in P_2} \text{demand}_{p,t_1}}_{\substack{\text{total remaining demand for final products from time period } t \text{ onward}}} \quad \forall\, t \in T. \tag{28.31}$$

- Controlling cash inflows and outflows:

$$\underbrace{\sum_{p\in P} p^c_{p,t} \times \text{demand}_{p,t}}_{\substack{\text{total revenue from contractual sales of oil products}}} + \underbrace{\sum_{n\in N}\sum p \in P_2 p^{\text{spot}}_{n,p,t} \times Y^{\text{spot}}_{n,p,t}}_{\substack{\text{total revenue from selling oil in the spot market}}}$$

$$+ \underbrace{\sum_{d\in D}\sum_{s=1}^{t} p^-_{d,s,t} \times V^-_{d,s,t}}_{\substack{\text{total revenue from selling derivatives}}} - \underbrace{\sum_{n\in N}\sum_{p\in P_1} c^x_{n,p,t} \times X_{n,p,t}}_{\substack{\text{total cost of buying crude oil from suppliers}}}$$

$$+ \underbrace{\sum_{n\in N}\sum_{p\in P_1} c^{\text{spot}}_{n,p,t} \times X^{\text{spot}}_{n,p,t}}_{\substack{\text{total cost of buying crude oil in the spot market}}} - \underbrace{\sum_{r\in R}\sum_{f\in F_r} c^z_{r,f,t} \times Z_{r,f,t}}_{\substack{\text{total cost of refining oil products}}}$$

$$- \underbrace{\sum_{p\in P}\sum_{m\in M}\sum_{(i,j)\in L_m} c^e_{i,j,p,m,t} \times E_{i,j,p,m,t}}_{\substack{\text{total cost of transporting oil products}}} - \underbrace{\sum_{n\in N}\sum_{p\in P} c^s_{n,p,t} \times S_{n,p,t}}_{\substack{\text{total cost of storing oil}}}$$

$$\underbrace{-\sum_{d\in D} p^{+}_{d,t} \times V_{d,t}}_{\text{total cost of buying derivatives}} \geq \text{bound} \quad \forall\, t \in T. \tag{28.32}$$

28.6 Implementation of the integrated CORPLAN model

Initially the CORPLAN model was run in deterministic mode for one randomly selected scenario on a small example of seven distribution network nodes (two of which correspond to refineries) and five time periods. Figure 28.12 illustrates the benefits of using an integrated policy.

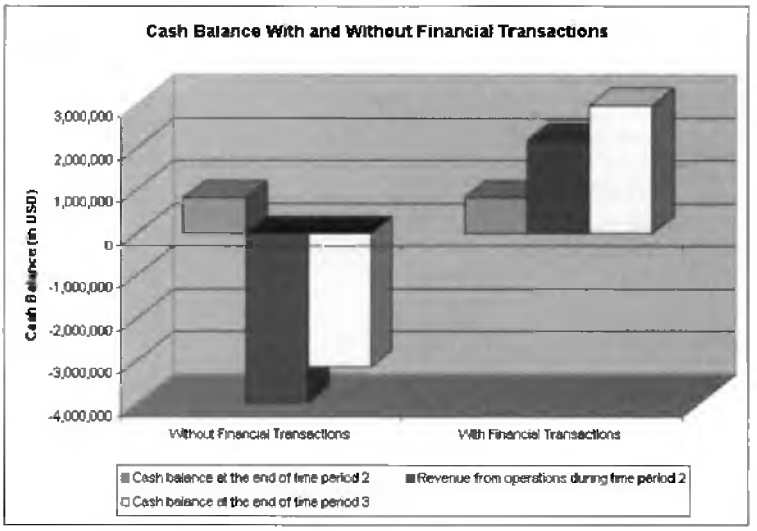


Figure 28.12. *Cash balance at the end of time period* 3 *with and without financial transactions in the implementation of the deterministic CORPLAN model.*

For the stochastic CORPLAN model the data are generated hierarchically, by generating world prices first and then using them for the generation of local prices in the distribution network (see Figure 28.13). In the integrated model nodal demand is generated together with nodal prices by using the historical correlation data (see [2]).

Figure 28.14 shows the spot and future prices of crude oil generated by the HOPE simulator.

An integrated corporate policy model is a very-large-scale stochastic program. The Stochgen module of the generic stochastic programming software STOCHASTICS (see Chapter 9 of this volume) has been used for the generation of the CORPLAN model and CPLEX 6.5 for its solution. The model has been run for an example of local demand generated over a 25-node transportation network. The results are summarized in Table 28.4. The prob-

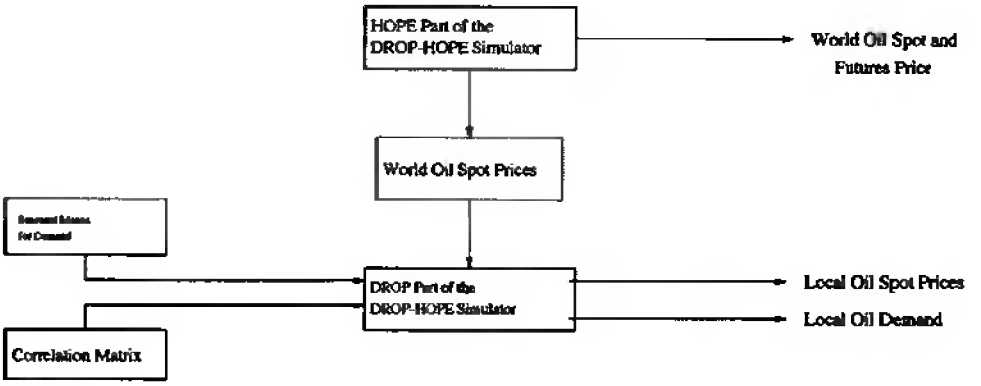


Figure 28.13. *Simulation of data processes for the integrated model.*

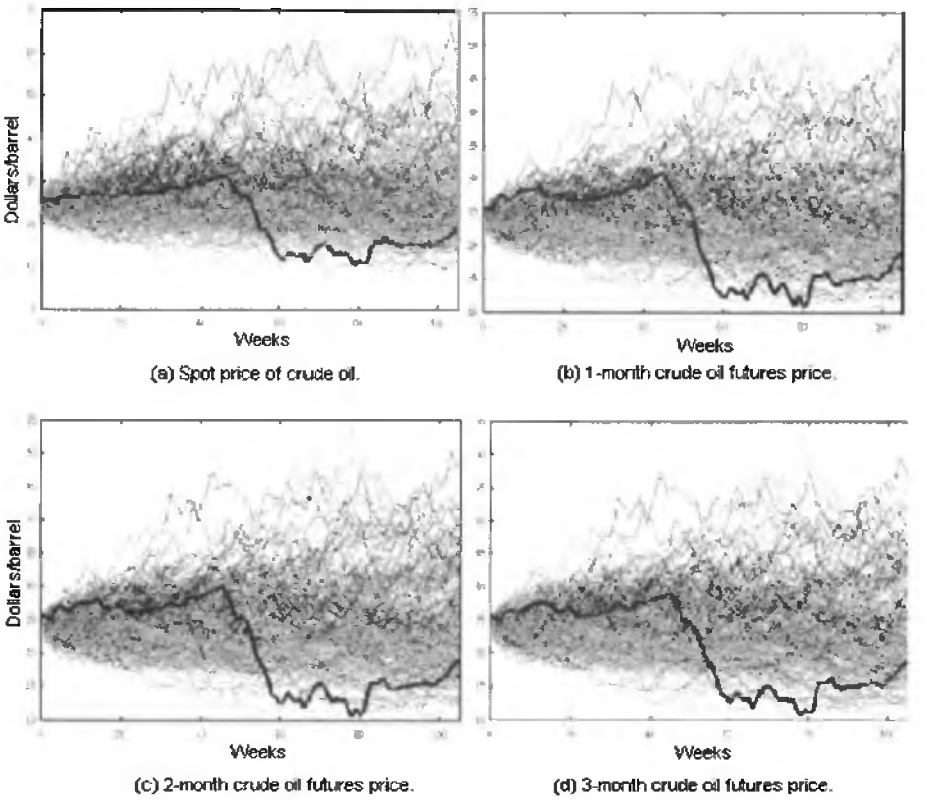


Figure 28.14. *Spot and futures prices of crude oil generated by the HOPE simulator.*

lem instances are generated by simulating the data for demand and price processes with 8 scenarios, 24 scenarios, and 27 scenarios. The planning horizon consists of 30 periods. The first five time periods are the implementable stage. In the two-stage problems, the tree

Table 28.4. *Stochastic programming results for the CORPLAN model.*

Problem	Scenarios	Deterministic eqnivalent			Optimal	Time (secs)	
		Rows	Columns	Nonzeros		Simplex	Barrier
Case_1_1	1	13521	43456	220537	129,615,736.6	1080.79	158.52
2-Stage							
Case_2_8	8	94665	297031	1539631	137,443,338.7	196919.18	2833.16
Case_2_24	24	280137	876631	4554703	—	—	—
Case_2_27	27	314913	985306	5120029	—	—	—
Multistage							
Case_4_8	8	90775	282541	1470901	142,347,248.6	185296.36	2241.45
Case_4_24	24	267689	830263	4334767	142,451,901.6	—	21114.02
Case_2_27	27	298575	924448	4831363	145,075,290.4	—	49092.19

branches in the first nonimplementable time period, i.e., period 6, and from then on the scenarios are independent. In the multistage (four-stage) problems, the tree branches in each of the first three nonimplementable time periods, i.e., periods 6, 7, and 8. The 8-scenario and 27-scenario trees have a binomial and trinomial structure, respectively, in the first three nonimplementable time periods. The 24-scenario tree unfolds unevenly over the first three nonimplementable time periods. More particularly, in time period 6 it branches four times, in time period 7 it branches three times, and in time period 8 it branches two times.

The optimal objective value of the CORPLAN problems corresponds to the expected terminal wealth of the company. In all problems, the initial cash balance equals \$100,000,000, so that the company makes a profit of around \$30,000,000 from its logistic and financial operations. For example, in the two-stage, 8-scenario problem the company's expected profit amounts to \$137,443,339−\$100,000,000 = \$37,443,339. But the expected cost of the company's logistic operations for the two-stage, 8-scenario DROP problem is equal to \$346,734,228, so the company's operating revenues amount to \$384,177,567. Then the company's expected profit margin is about 10%, which is reasonable.

The results of the CORPLAN model (summarized in Table 28.4) require detailed analysis of the optimal decisions in each implementation. Given the size of the problems and the large number of variables in each, it is virtually impossible to analyze the optimal decisions without appropriate software and visualization tools. In general, we observe that the CORPLAN model improves on the optimal solution of the DROP physical operations model and that the inclusion of financial operations in the model allows for much more flexibility in decisions. The multistage problems are more realistic than their two-stage counterparts in that more future decisions face uncertainty. They give higher profits than their counterparts due to better exploitation of this flexibility.

The complexity of the CORPLAN model is reflected in the amount of time it takes to solve it using commercial software (CPLEX 6.5) on an Athlon 650 Mhz PC with 1 GB RAM running under Red Hat LINUX 7.0. For large problems, the barrier algorithm is faster than the primal simplex algorithm, but even then solving a CORPLAN problem takes 1.5 times longer than solving the corresponding DROP logistics problem. The more realistic multistage versions are smaller and more easily solvable than their two-stage counterparts.

From an implementation point of view, all instances of the CORPLAN model are difficult to solve (blank cells in Table 28.4 represent unsolved problems) and when solvable, an instance requires a substantial amount of time by standard methods. This is a disadvantage which might indeed make it impractical for tactical-level planning. Alternative solution methods which analyze and exploit the problem's structure, such as decomposition techniques, are reported in Chapter 9 of this volume and in [12]. All problems in Table 28.4 have been solved to optimality in order of magnitude faster times by the `solgen` module of the STOCHASTICS system, which implements nested Benders decomposition.

28.7 Conclusions and future research directions

Our analysis of the MGMR case using a stochastic optimization formulation of the company's hedging strategy led us to think about hedging in coordination with physical activities. Combining hedging with logistics management should improve the effectiveness of strategic planning decisions. In the presence of volatile energy markets this is a very challenging problem. The implementation and solution of our proposed model relies on the existence of organized derivatives markets for trading. In fact, the growing number of energy derivatives and the existence of established markets for trading them creates new opportunities for risk management based on reliable quantitative models.

Dynamic stochastic programming has provided us with the framework for formulating integrated corporate planning problems. The generic stochastic programming software STOCHASTICS used here allows solution of large enough problems in a reasonable time. However, it should be emphasized that a correctly formulated integrated problem, whereby a sophisticated hedging strategy is combined with complex logistics operations management, is difficult to implement and solve accurately. Some of the difficulties encountered include obtaining confidential data on the costs of physical operations, simulating the forward markets accurately, formulating clearly the company's overall objectives, and, most important, implementing such a computationally intensive problem solution.

Appendix

Specification of the CORPLAN model details

The detailed mathematical formulation of the CORPLAN model was given in section 28.5.2. We hereby include the sets, which identify the parameters and variables of the model, as well as the parameters and bounds of the DROP logistics planning part of the CORPLAN model.

Sets

$N := \{\text{nodes}\}$,

$L := \{(n_1, n_2) : n_1, n_2 N,\ n_1 \neq n_2\}$,

$T := \{\text{time periods}\}$,

O := {operators},

P := {products}, $P_i := \begin{cases} \text{\{crude oils\}}, & i=1 \\ \text{\{final products\}}, & i=2 \end{cases}$,

R := {refineries}, $R \subset N$,

F := {transformation technologies},

F_n := {transformation technology available at node n},

F_p^{in} := {transformation technology that uses product p as input},

F_p^{out} := {transformation technology that produces product p as output},

M := {transportation means},

M^c := {transportation means that allow continuous flows of product},

M^d := {transportation means that allow discrete flows of product},

P_m := $\{p \in P :$ product p can be transported by means $m\}$,

L_m := $\{(n_1, n_2) \in L :$ means m can be used from node n_1 to node $n_2\}$,

$T_{n_1,n_2,p,m}$:= $\{t \in T:$ the transportation of product p from node n_1 to node n_2 by transport means m is allowed to start during time period $t\}$,

U := {types of horizon segments[1] used for bounding product volumes transported},

T_u^U := $\{t \in T :$ time periods t included in u, for $u \in U\}$,

V := {types of horizon segments used for bounding product volumes supplied},

T_v^V := $\{t \in T :$ time periods t included in v, for $v \in V\}$,

H := {types of horizon segments used for bounding product volumes refined},

T_h^H := $\{t \in T :$ time periods t included in h, for $h \in H\}$,

Θ := {entities},

N_θ^Θ := $\{n \in N :$ n belongs to entity $\theta \in \Theta\}$, $N_\theta^\Theta \subset N$,

P_θ^Θ := $\{p \in P :$ p belongs to entity $\theta \in \Theta\}$, $P_\theta^\Theta \subset P$,

T_θ^Θ := $\{t \in T :$ t belongs to entity $\theta \in \Theta\}$, $T_\theta^\Theta \subset T$.

[1] A *horizon segment* is a set of periods, not necessarily consecutive, included in the set of consecutive time periods defining the problem's horizon for planning. A *type* of horizon segment is a horizon segment defined by the time periods it includes.

Parameters

$c^x_{n,o,p,t}$: cost of supplying one unit of product p (by operator o) at node n during time period t.

$c^z_{n,f,t}$: cost of transforming one unit of input product volume at node n by technology f, when the transformation (i.e., refining) takes place during time period t and is output for the beginning of period $t+1$ at the end of period t. The input volume is a mixture of products which are entered in fixed proportions for transformation by the particular technology.

$c^e_{n_1,n_2,p,m,t}$: cost of transporting one unit of product p from node n_1 to node n_2 by means m if the transportation starts during time period t.

$c^s_{n,p,t}$: cost of stocking one unit of product p at node n during period t and paid at the beginning of time period $t+1$.

$c^{\text{spot}}_{n,p,t}$: cost of (spot) purchasing and supplying one unit of product p at node n during time period t paid at the beginning of period t.

$p^{\text{spot}}_{n,p,t}$: price charged for (spot) selling one unit of product p at node n during time period t paid at the beginning of period t.

$s^0_{n,p}$: initial stock volume of product p at the beginning of time period 1.

$g_{f,p}$: If p belongs to the set of crude oils, this parameter denotes the proportion of product p in the total input volume used for transformation by technology f. If p belongs to the set of final products, this parameter denotes the proportion of product p produced by technology f. Note that $\sum_{p\in P_1:\ f\in F^{in}_p} g_{f,p} = 1$, where P_1 is the set of crude oil products, whereas this need not hold for final products because there might be some wastage in the refining process.

$\Delta_{n_1,n_2,p,m}$: integer number of time periods minus one required for transportation of product p from node n_1 to node n_2 by transport means m. This means that if the transportation is expected to last at most one time period, then the products transported will reach their destination within the time period of their departure.

$d_{n,o,p,t}$: product p volume that must be set aside at node n at the beginning of time period t in order to guarantee satisfaction of demand (for operator o) during time period t.

$e_{n,p,t}$: product p volume that arrives in transit at node n during time period t and whose transportation from another node started before the beginning of the planning horizon.

Bounds

$e^c_{n_1,n_2,m,u}$: upper bound on the total product volume in transit from node n_1 to node n_2 by continuous transport means m (i.e., pipelines) during some time period included in the horizon segment u.

$e^{d}_{m,t}$: upper bound on the total product volume in transit during time period t. This is the bound imposed by capacity on discrete transport means, such as ships, tankers, and wagons.

$\underline{x}_{n,o,p,v}$: lower bound on the volume of product p supplied by operator o to node n during the time periods included in the horizon segment v.

$\overline{x}_{n,o,p,v}$: upper bound on the volume of product p supplied by operator o to node n during the time periods included in the horizon segment v.

$\underline{z}_{n,f,h}$: lower bound on the product volume refined in refinery n by technology f during the time periods included in the horizon segment h.

$\overline{z}_{n,f,h}$: upper bound on the product volume refined in refinery n by technology f during the time periods included in the horizon segment h.

$\overline{z}^{T}_{n,f,t}$: upper bound on the product volume transformed by technology f at refinery n during time period t.

$\underline{s}_{n.p.t}$: lower bound on the product volume stocked at node n during time period t.

$\overline{s}_{n.p.t}$: upper bound on the product volume stocked in node n during time period t.

$\underline{s}_{\theta}$: lower bound on the product stock at entity θ.

$\overline{s}_{\theta}$: upper bound on the product stock at entity θ.

Acknowledgments

The authors thank Michael Dempster for his guidance, comments, and criticism. James Scott helped to solve large instances of the dynamic stochastic programming problems discussed as a part of the development and testing of CSA's STOCHASTICS software. George Hong and Shahab Khokhar's implementation of a futures-spot commodity price simulator has been applied to the simulation of oil commodity prices. This research was partially funded by the European Commission, and the second author gratefully acknowledges the support of the UK ESRC for her doctoral research.

Bibliography

[1] C. L. CULP AND M. H. MILLER, *Metallgesellschaft and the economics of synthetic storage*, J. Appl. Corporate Finance, 7 (1995), pp. 62–76.

[2] M. A. H. DEMPSTER, N. HICKS-PEDRÓN, E. A. MEDOVA, J. E. SCOTT, AND A. SEMBOS, *Planning logistics operations in the oil industry*, J. Oper. Res. Soc., 51 (2000), pp. 1271–1288.

[3] M. A. H. DEMPSTER, S. S. G. HONG, AND S. Q. KHOKHAR, *Implementation of a Model of the Stochastic Behaviour of Commodity Prices*, Technical Report, Centre for Financial Research, University of Cambridge, Cambridge, UK, 1999.

[4] F. R. EDWARDS, *Derivatives can be hazardous to your health: The case of Metallgesellschaft*, in The Emerging Framework of Financial Regulation, C. A. E. Goodhart, ed., Central Banking Publications, London, 1998, pp. 351–377.

[5] F. R. EDWARDS AND M. S. CANTER, *The collapse of Metalgesellschaft: Unhedgeable risks, poor hedging strategy, or just bad luck?*, in The Emerging Framework of Financial Regulation, C. A. E. Goodhart, ed., Central Banking Publications, London, 1998, pp. 381–441.

[6] L. F. ESCUDERO, F. J. QUINTANA, AND J. SALMERÒN, *CORO, A modelling and algorithmic framework for oil supply, transformation and distribution optimization under uncertainty*, Eur. J. Oper. Res., 114 (1999), pp. 638–656.

[7] *Enron: Over there and over paying*, Financial Times (12 Feb. 2002), London.

[8] E. N. KRAPELS AND M. PRATT, *Crude Oil Hedging*, Risk Books, London, 1998.

[9] A. KUPRIANOV, *Derivatives debacles: Case studies of large losses in derivatives markets*, in Derivatives Handbook: Risk Management and Control, R. J. Schwartz and W. S. Clifford, eds., John Wiley, New York, 1997, pp. 605–631.

[10] A. S. MELLO AND J. E. PARSONS, 1997, *Maturity structure of a hedge matters: Lessons from the Metallgesellschaft debacle*, in Derivatives Handbook: Risk Management and Control, R. J. Schwartz and W. S. Clifford, eds., John Wiley, New York, 1997, pp. 575–594.

[11] E. S. SCHWARTZ, *The stochastic behaviour of commodity prices: Implications for valuation and hedging*, J. Finance, 52 (1997), pp. 923–973.

[12] J. E. SCOTT, *Modelling and Solution of Large-Scale Stochastic Programs*, Ph.D. thesis, Centre for Financial Research, Judge Institute of Management, University of Cambridge, Cambridge, UK, 2001.

[13] A. SEMBOS, *Integrated Logistics and Financial Planning in the Oil Industry*, Ph.D. thesis, Centre for Financial Research, Judge Institute of Management, University of Cambridge, Cambridge, UK, 2001.

This page intentionally left blank

Chapter 29

Numerical Comparison of Conditional Value-at-Risk and Conditional Drawdown-at-Risk Approaches: Application to Hedge Funds

P. Krokhmal, S. Uryasev,* and G. Zrazhevsky**

29.1 Introduction

This paper applies risk management methodologies to the optimization of a portfolio of hedge funds (fund of funds). We compare risk management techniques based on two recently developed risk measures, conditional value at risk (CVaR) and conditional drawdown at risk (CDaR) [12, 35, 36]. Both risk management techniques utilize stochastic programming approaches and allow for construction of *linear portfolio rebalancing strategies* and, as a result, have proven their high efficiency in various portfolio management applications [5, 12, 27, 35, 36]. The choice of hedge funds, as a subject for the portfolio optimization strategy, was stimulated by a strong interest in this class of assets among both practitioners and scholars, as well as by challenges related to relatively small data sets available for hedge funds.

Recent studies[1] of the hedge funds industry are mostly concentrated on the classification of hedge funds and the relevant investigation of their activity. However, this paper is focused on possible realization of investment opportunities existing in this market from the viewpoint of portfolio rebalancing strategies. (For an extensive discussion of stochastic programming approaches to hedge fund management, see [46].)

Hedge funds are investment pools employing sophisticated trading and arbitrage techniques, including leverage and short selling, wide usage of derivative securities, etc. Generally, hedge funds restrict share ownership to high-net-worth individuals and institutions and are not allowed to offer their securities to the general public. Many hedge funds are limited to 99 investors. This private nature of hedge funds has resulted in few regulations and disclosure requirements, compared, for example, with mutual funds. (However, stricter

*Risk Management and Financial Engineering Laboratory, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611 (krokhmal@ufl.edu, uryasev@ise.ufl.edu, zrazhevsky@ise.ufl.edu).

[1] See, for example, [3, 4, 9, 19, 20, 21, 29].

regulations exist for hedge funds trading futures.) Also, the hedge funds may take advantage of specialized, risk-seeking investment and trading strategies, which other investment vehicles are not allowed to use.

The first official[2] hedge fund was established in the United States by A. W. Jones in 1949, and its activity was characterized by the use of short selling and leverage, which were separately considered risky trading techniques, but in combination could limit market risk. The term "hedge fund" applies to the structure of the Jones fund's portfolio, which was split between long positions in stocks that would gain in value if market went up and short positions in stocks that would protect against market drop. Also, Jones introduced another two initiatives, which became common practice in the hedge fund industry and which with more or less variation have survived to this day: he made the manager's incentive fee a function of a fund's profits and kept his own capital in the fund, in this way making the incentives of a fund's clients and of his own coherent.

Nowadays, hedge funds are a rapidly growing part of the financial industry. According to Van Hedge Fund Advisors, the number of hedge funds at the end of 1998 was 5830, and they manage $311 billion in capital, with between $800 billion and $1 trillion in total assets. Nearly 80% of hedge funds have market capitalization less than $100 million, and around 50% are smaller than $25 million, which indicates a high number of new entries. More than 90% of hedge funds are located in the United States.

Hedge funds are subject to far fewer regulations than other pooled investment vehicles, especially to regulations designed to protect investors. This applies to such regulations as regulations on liquidity, requirements that a fund's shares be redeemable at any time, protecting conflicts of interests, ensuring fairness of pricing of fund shares, disclosure requirements, limiting usage of leverage, short selling, etc. This minimal oversight is a consequence of the fact that a hedge funds' investors qualify as sophisticated high-income individuals and institutions, who can stand up for themselves. Hedge funds offer their securities as private placements, on an individual basis, rather than through public advertisement, which allows them to avoid disclosing publicly their financial performance or asset positions. However, hedge funds must provide to investors some information about their activity, and of course they are subject to statutes governing fraud and other criminal activities.

As market's subjects, hedge funds are subordinate to regulations that protect market integrity by detecting attempts to manipulate or dominate markets by individual participants. For example, in the United States hedge funds and other investors active on currency futures markets must regularly report large positions in certain currencies. Also, many option exchanges have developed a large option position reporting system to track changes in large positions and identify outsized short uncovered positions.

In this paper, we consider the problem of managing a fund of funds, i.e., constructing optimal portfolios from sets of hedge funds, subject to various risk constraints, which control different types of risks. However, the practical use of the strategies is limited by restrictive assumptions[3] imposed in this case study: (1) liquidity considerations are not taken into account; (2) there are no transaction costs; (3) considered funds may be closed to new investors; (4) credit and other risks which are not directly reflected in the historical return

[2]Ziemba [46] traces early unofficial hedge funds, such as the Keynes Chest Fund, that existed from the 1920s to the 1940s.

[3]These assumptions can be relaxed and incorporated into the model as linear constraints. Here we focus on comparison of risk constraints and have not included other constraints.

data are not taken into account; and (5) survivorship bias is not considered. The obtained results cannot be treated as direct recommendations for investing in the hedge funds market but rather as a description of the risk management methodologies and portfolio optimization techniques in a realistic environment. For an overview of the potential problems related to the data analysis and portfolio optimization of hedge funds, see [30].

Section 29.2 presents an overview of linear portfolio optimization algorithms and the related risk measures, which are explored in this paper. Section 29.3 contains a description of our case study, results of in-sample and out-of-sample experiments, and their detailed discussion. Section 29.4 presents the concluding remarks.

29.2 Risk management using CVaR and CDaR

Formal portfolio management methodologies assume some measure of risk that affects allocation of instruments in the portfolio. The classical Markowitz theory, for example, identifies risk with the volatility (standard deviation) of a portfolio. In this study we investigate a portfolio optimization problem with three different constraints on risk: CVaR [35, 36], CDaR [12], and market neutrality.[4] CVaR and CDaR risk measures represent relatively new developments in the risk management field. Application of these risk measures to portfolio allocation problems relies on the scenario representation of uncertainties and stochastic programming approaches.

A *linear portfolio rebalancing algorithm* is a trading (investment) strategy with mathematical model that can be formulated as a linear programming (LP) problem. The focus on LP techniques in application to portfolio rebalancing and trading problems is explained by the exceptional effectiveness and robustness of LP algorithms, which becomes especially important in finance applications. Recent developments (see, for example, [5, 10, 11, 12, 13, 14, 15, 16, 27, 35, 36, 41, 42, 44, 47]) show that LP-based algorithms can successfully handle portfolio allocation problems with thousands and even millions of decision variables and scenarios, which makes those algorithms attractive to institutional investors.

In the cited papers, along with CVaR and CDaR, other, much earlier established measures of risk, such as maximum loss, mean-absolute deviation, low partial moment with power one, and expected regret,[5] have been employed in the framework of linear portfolio rebalancing algorithms (see, for example, [45]). Some of these risk measures are quite closely related to the CVaR concept.[6] We restricted ourselves to considering CVaR- and CDaR-based risk management techniques.

[4]There are different interpretations for the term "market neutral" (see, for instance, [8]). In this paper market neutrality means zero beta.

[5]Low partial moment with power one is defined as the expectation of losses exceeding some fixed threshold; see [22]. Expected regret (see, for example, [15]) is a concept similar to the lower partial moment. However, the expected regret may be calculated with respect to a random benchmark, while the low partial moment is calculated with respect to a fixed threshold.

[6]Maximum loss is a limiting case of CVaR risk measure (see below). Also, Testuri and Uryasev [40] showed that the CVaR constraint and the low partial moment constraint with power one are equivalent in the sense that the efficient frontier for portfolio with CVaR constraint can be generated by the low partial moment approach. Therefore, the risk management with CVaR and with low partial moment leads to similar results. However, the CVaR approach allows for direct controlling of percentiles, while the low partial moment penalizes losses exceeding some fixed thresholds.

However, the class of linear trading or portfolio optimization techniques is far from encompassing the entire universe of portfolio management techniques. For example, the famous portfolio optimization model by Markowitz [31, 32], which utilizes the mean-variance approach, belongs to the class of quadratic programming (QP) problems; the well-known constant proportion rule leads to nonconvex multiextremum problems, etc.

29.2.1 CVaR

The CVaR measure (see [35, 36]) develops and enhances the ideas of risk management, which have been put in the framework of value at risk (VaR) (see, for example, [17, 23, 37, 39]). Incorporating such merits as an easy-to-understand concept, simple and convenient representation of risks (one number), and applicability to a wide range of instruments, VaR has evolved into a current industry standard for estimating risks of financial losses. Basically, VaR answers the question, "what is the maximum loss, which is expected to be exceeded, say, in only 5% of the cases within the given time horizon?" For example, if the daily VaR for the portfolio of some fund XYZ is equal to \$10 million at confidence level 0.95, there is only a 5% chance of losses exceeding \$10 million during a trading day.

The formal definition of VaR is as follows. Consider a loss function $f(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ is a decision vector (e.g., portfolio positions) and $\mathbf{y}$ is a stochastic vector standing for market uncertainties. (In this paper, $\mathbf{y}$ is the vector of returns of instruments in the portfolio.) Let $\Psi(\mathbf{x}, \zeta)$ be the cumulative distribution function of $f(\mathbf{x}, \mathbf{y})$,

$$\Psi(\mathbf{x}, \zeta) = P[f(\mathbf{x}, \mathbf{y}) \leq \zeta].$$

Then the VaR function $\zeta_\alpha(\mathbf{x})$ with the confidence level α is the α-quantile of $f(\mathbf{x}, \mathbf{y})$ (see Figure 29.1):

$$\zeta_\alpha(\mathbf{x}) = \min_{\zeta \in \mathbb{R}} \{\Psi(\mathbf{x}, \zeta) \geq \alpha\}.$$

Using VaR as a risk measure in portfolio optimization is, however, a very difficult problem if the return distributions of a portfolio's instruments are not normal or lognormal. The

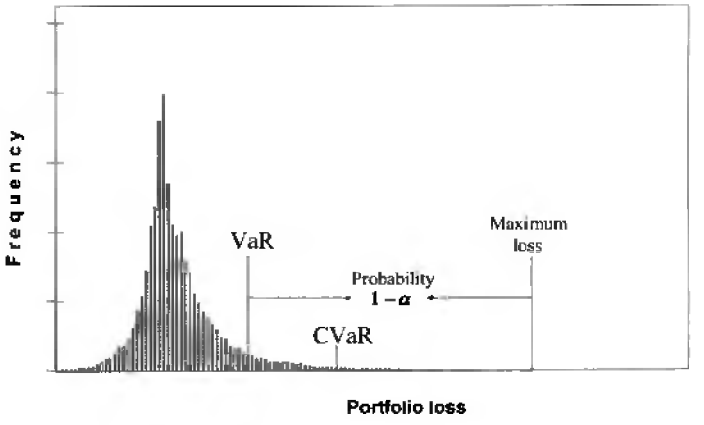


Figure 29.1. *Loss distribution, VaR, CVaR, and maximum loss.*

optimization difficulties with VaR are caused by its nonconvex and nonsubadditive nature [6, 7, 33]. Nonconvexity of VaR means that as a function of portfolio positions, it has multiple local extrema, which precludes using efficient optimization techniques.

The difficulties with controlling and optimizing VaR in nonnormal portfolios have forced the search for similar percentile risk measures, which would also quantify downside risks and at the same time could be efficiently controlled and optimized. From this viewpoint, CVaR is a perfect candidate for conducting VaR-style portfolio management.

For continuous distributions, CVaR is defined as an average (expectation) of high losses residing in the α-tail of the loss distribution, or, equivalently, as a conditional expectation of losses exceeding the α-VaR level (Figure 29.1). From this it follows that CVaR incorporates information on VaR and on the losses exceeding VaR.

For general (noncontinuous) distributions, Rockafellar and Uryasev [36] defined α-CVaR function $\phi_\alpha(\mathbf{x})$ as the α-tail expectation of a random variable z,

$$\phi_\alpha(\mathbf{x}) = E_{\alpha\text{-tail}}[z],$$

where the α-tail cumulative distribution functions of z have the form

$$\Psi_\alpha(\mathbf{x}, \zeta) = P[z \le \zeta] = \begin{cases} 0, & \zeta < \zeta_\alpha(\mathbf{x}), \\ [\Psi(\mathbf{x}, \zeta) - \alpha]/[1 - \alpha], & \zeta \ge \zeta_\alpha(\mathbf{x}). \end{cases}$$

Also, Acerbi, Nordio, and Sirtori [1] and Acerbi and Tasche [2] redefined expected shortfall similar to the CVaR definition presented above.

Along with α-CVaR function $\phi_\alpha(\mathbf{x})$, the following functions, called upper and lower CVaR (α-CVaR$^+$ and α-CVaR$^-$), are considered:

$$\phi_\alpha^+(\mathbf{x}) = E[f(\mathbf{x}, \mathbf{y}) | f(\mathbf{x}, \mathbf{y}) > \zeta_\alpha(\mathbf{x})],$$
$$\phi_\alpha^-(\mathbf{x}) = E[f(\mathbf{x}, \mathbf{y}) | f(\mathbf{x}, \mathbf{y}) \ge \zeta_\alpha(\mathbf{x})].$$

The CVaR functions satisfy the inequality

$$\phi_\alpha^-(\mathbf{x}) \le \phi_\alpha(\mathbf{x}) \le \phi_\alpha^+(\mathbf{x}).$$

Rockafellar and Uryasev [36] showed that α-CVaR can be presented as a convex combination of α-VaR and α-CVaR$^+$,

$$\phi_\alpha(\mathbf{x}) = \lambda_\alpha(\mathbf{x})\zeta_\alpha(\mathbf{x}) + [1 - \lambda_\alpha(\mathbf{x})]\phi_\alpha^+(\mathbf{x}),$$

where

$$\lambda_\alpha(\mathbf{x}) = \frac{[\Psi(\mathbf{x}, \zeta_\alpha(\mathbf{x})) - \alpha]}{[1 - \alpha]}, \quad 0 \le \lambda_\alpha(\mathbf{x}) \le 1.$$

For a discrete loss distribution, where the stochastic parameter $\mathbf{y}$ may take values $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_J$ with probabilities θ_j, $j = 1, \ldots, J$, the α-VaR and α-CVaR functions, respectively, are[7]

$$\zeta_\alpha(\mathbf{x}) = f(\mathbf{x}, \mathbf{y}_{j_\alpha}),$$

[7]This proposition has been derived with the assumption that, without loss of generality, scenarios $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_J$ satisfy inequalities $f(\mathbf{x}, \mathbf{y}_1) \le \cdots \le f(\mathbf{x}, \mathbf{y}_J)$.

$$\phi_\alpha(\mathbf{x}) = \frac{1}{1-\alpha}\left[\left(\sum_{j=1}^{j_\alpha}\theta_j - \alpha\right) f(\mathbf{x}, \mathbf{y}_{j_\alpha}) + \sum_{j=j_\alpha+1}^{J}\theta_j f(\mathbf{x}, \mathbf{y}_j)\right],$$

where j_α satisfies

$$\sum_{j=1}^{j_\alpha-1}\theta_j < \alpha \le \sum_{j=1}^{j_\alpha}\theta_j.$$

For values of confidence level α close to 1, CVaR coincides with the maximum loss (see Figure 29.1).

While inheriting some of the good properties of VaR, such as measuring downside risks and representing them by a single number, applicability to instruments with nonnormal distributions, etc., CVaR has substantial advantages over VaR from the risk management standpoint. First, CVaR is a convex function[8] of portfolio positions. Hence, it has a convex set of minimum points on a convex set, which greatly simplifies control and optimization of CVaR. Calculation of CVaR, as well as its optimization, can be performed by means of a convex programming shortcut [35, 36], where the optimal value of CVaR is calculated simultaneously with the corresponding VaR; for linear or piecewise-linear loss functions these procedures can be reduced to linear programming problems. Also, unlike α-VaR, α-CVaR is continuous with respect to confidence level α. A comprehensive description of the CVaR risk measure and CVaR-related optimization methodologies can be found in [35, 36]. Also, Rockafellar and Uryasev [35] showed that for normal loss distributions, the CVaR methodology is equivalent to the standard mean-variance approach. Similar results also were independently proved for elliptic distributions by Embrechts, McNeil, and Straumann [18].

According to Rockafellar and Uryasev [35, 36], the optimization problem with multiple CVaR constraints

$$\begin{aligned} &\min_{\mathbf{x}\in X} && g(\mathbf{x}) \\ &\text{subject to} && \phi_{\alpha_i}(\mathbf{x}) \le \omega_i, \quad i = 1, \ldots, I, \end{aligned}$$

is equivalent to the following problem:

$$\begin{aligned} &\min_{\mathbf{x}\in X,\ \zeta_k\in\mathbf{R}\ \forall k} && g(\mathbf{x}) \\ &\text{subject to} && \zeta_k + \frac{1}{1-\alpha_k}\sum_{j=1}^{J}\theta_j \max\{0, f(\mathbf{x}, \mathbf{y}_j) - \zeta_k\} \le \omega_k, \quad k = 1, \ldots, K, \end{aligned}$$

provided that the objective function $g(\mathbf{x})$ and the loss function $f(\mathbf{x}, \mathbf{y})$ are convex in $\mathbf{x} \in X$. When the objective and loss functions are linear in $\mathbf{x}$ and constraints $\mathbf{x} \in X$ are given by linear inequalities, the last optimization problem can be reduced to LP; see [35, 36].

Except for the fact that CVaR can be easily controlled and optimized, CVaR is a more adequate measure of risk as compared to VaR because it accounts for losses beyond the VaR level. The fundamental difference between VaR and CVaR as risk measures are that VaR

[8] For a background on convex functions and sets, see [34].

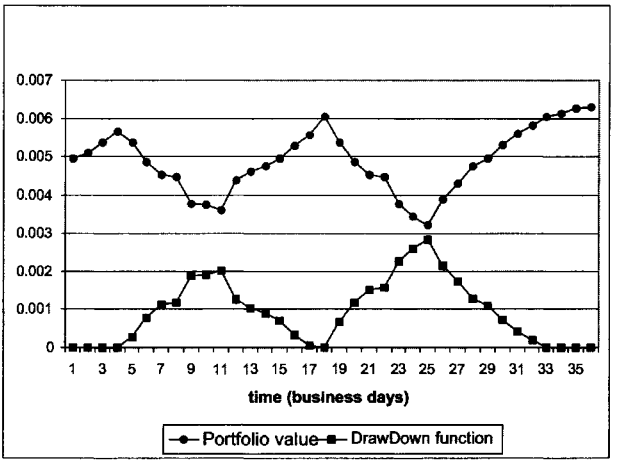


Figure 29.2. *Portfolio value and drawdown.*

is the "optimistic" low bound of the losses in the tail, while CVaR gives the value of the *expected* losses in the tail. In risk management, we may prefer to be neutral or conservative rather than optimistic. Moreover, CVaR satisfies several nice mathematical properties and is coherent in the sense of [6, 7].

29.2.2 CDaR

CDaR is a portfolio performance measure [12] closely related to CVaR. By definition, a portfolio's *drawdown* on a sample path is the drop of the uncompounded[9] portfolio value as compared to the maximal value attained in the previous moments on the sample path. Suppose, for instance, that we start observing a portfolio in January, 2001, and record its uncompounded value every month.[10] If the initial portfolio value was \$100,000,000 and in February it reached \$130,000,000, then the portfolio drawdown as of February, 2001, is \$0. If in March, 2001, the portfolio value drops to \$90,000,000, then the current drawdown equals \$40,000,000 (in absolute terms), or 30.77%. Mathematically, the drawdown function for a portfolio is

$$\tilde{f}(\mathbf{x}, t) = \max_{0 \le \tau \le t} \{v_\tau(\mathbf{x})\} - v_t(\mathbf{x}), \tag{29.1}$$

where $\mathbf{x}$ is the vector of portfolio positions and $v_t(\mathbf{x})$ is the *uncompounded* portfolio value at time t. We assume that the initial portfolio value is equal to 1; therefore, the drawdown is the uncompounded portfolio return starting from the previous maximum point. Figure 29.2 illustrates the relation between the portfolio value and the drawdown.

[9]Drawdowns are calculated with uncompounded portfolio returns. This is related to the fact that risk measures based on drawdowns of uncompounded portfolios have nice mathematical properties. In particular, these measures are convex in portfolio positions. Suppose that at the initial moment $t = 0$ the portfolio value equals v and portfolio returns in the moments $t = 1, \ldots, T$ equal $r_1, \ldots, r_T$. By definition, the uncompounded portfolio value v_τ at time moment τ equals $v_\tau = v \sum_{t=1}^{\tau} r_t$. We assume that the initial portfolio value $v = 1$.

[10]Usually, portfolio value is observed much more frequently. However, for the hedge funds considered in this paper, data are available on a monthly basis.

The drawdown quantifies the financial losses in a conservative way: it calculates losses for the most "unfavorable" investment moment in the past as compared to the current (discrete) moment. This approach reflects quite well the preferences of investors who define their allowed losses in percentages of their initial investments (e.g., an investor may consider it unacceptable to lose more than 10% of his investment). While an investor may accept small drawdowns in his account, he would definitely start worrying about his capital in the case of a large drawdown. Such drawdown may indicate that something is wrong with that fund, and maybe it is time to move the money to a more successful investment pool. The mutual and hedge fund concerns are focused on keeping existing accounts and attracting new ones; therefore, they should ensure that clients' accounts do not have large drawdowns.

One can conclude that *drawdown* accounts not only for the amount of losses over some period but also for the *sequence* of these losses. This highlights the unique feature of the drawdown concept: it is a loss measure "with memory," taking into account the time sequence of losses.

For a specified sample path, the drawdown function is defined for each time moment. However, to evaluate performance of a portfolio on the whole sample path, we would like to have a function that aggregates all drawdown information over a given time period into one measure. As this function one can pick, for example, the maximum drawdown,

$$\text{MaxDD} = \max_{0 \le t \le T} \{\tilde{f}(\mathbf{x}, t)\},$$

or the average drawdown,

$$\text{AverDD} = \frac{1}{T} \int_0^T \tilde{f}(\mathbf{x}, t) dt.$$

However, both these functions may inadequately measure losses. The maximum drawdown is based on one worst-case event in the sample path. This event may represent some very specific circumstances, which may not appear in the future. The risk management decisions based only on this event may be too conservative.

On the other hand, the average drawdown takes into account all drawdowns in the sample path. However, small drawdowns are acceptable (e.g., 1–2% drawdowns) and averaging may mask large drawdowns.

Chekhlov, Uryasev, and Zabarankin [12] suggested a new drawdown measure, CDaR, that combines both the drawdown concept and the CVaR approach. For instance, 0.95-CDaR can be thought of as an average of 5% of the highest drawdowns. Formally, α-CDaR is α-CVaR with drawdown loss function $\tilde{f}(\mathbf{x}, t)$ given by (29.1). Namely, assume that possible realizations of the random vectors describing uncertainties in the loss function are represented by a sample path (time-dependent scenario), which may be obtained from historical or simulated data. In this paper, it is assumed that we know one sample path of returns of instruments included in the portfolio. Let r_{ij} be the rate of return of the ith instrument in the jth trading period (that corresponds to the jth month in the case study; see below), $j = 1, \ldots, J$. Suppose that the initial portfolio value equals 1. Let $x_i, i = 1, \ldots, n$, be weights of instruments in the portfolio. The uncompounded portfolio value at time j

equals

$$v_j(\mathbf{x}) = \sum_{i=1}^{n} \left(1 + \sum_{s=1}^{j} r_{is}\right) x_i.$$

The drawdown function $\tilde{f}(\mathbf{x}, \mathbf{r}_j)$ at time j is defined as the drop in the portfolio value compared to the maximum value achieved before the time moment j,

$$\tilde{f}(\mathbf{x}, j) = \max_{1 \le k \le j} \left\{ \sum_{i=1}^{n} \left(\sum_{s=1}^{k} r_{is} \right) x_i \right\} - \sum_{i=1}^{n} \left(\sum_{s=1}^{j} r_{is} \right) x_i.$$

Then, the CDaR function $\Delta_\alpha(\mathbf{x})$ is defined as follows. If the parameter α and number of scenarios J are such that their product $(1-\alpha)J$ is an integer number, then $\Delta_\alpha(\mathbf{x})$ is defined as

$$\Delta_\alpha(\mathbf{x}) = \eta_\alpha + \frac{1}{(1-\alpha)J} \sum_{j=1}^{J} \max \left\{ 0, \max_{1 \le k \le j} \left[\sum_{i=1}^{n} \left(\sum_{s=1}^{k} r_{is} \right) x_i \right] - \sum_{i=1}^{n} \left(\sum_{s=1}^{j} r_{is} \right) x_i - \eta_\alpha \right\},$$

where $\eta_\alpha = \eta_\alpha(\mathbf{x})$ is the threshold that is exceeded by $(1-\alpha)J$ drawdowns. In this case the drawdown function $\Delta_\alpha(\mathbf{x})$ is the average of the worst case $(1-\alpha)J$ drawdowns observed in the considered sample path. If $(1-\alpha)J$ is not an integer, then the CDaR function, $\Delta_\alpha(\mathbf{x})$, is the solution of

$$\Delta_\alpha(\mathbf{x}) = \min_{\eta} \left\{ \eta + \frac{1}{1-\alpha} \frac{1}{J} \times \sum_{j=1}^{J} \max \left[0, \max_{1 \le k \le j} \left\{ \sum_{i=1}^{n} \left(\sum_{s=1}^{k} r_{is} \right) x_i \right\} - \sum_{i=1}^{n} \left(\sum_{s=1}^{j} r_{is} \right) x_i - \eta \right] \right\}.$$

The CDaR risk measure holds nice properties of CVaR such as convexity with respect to portfolio positions. Also, CDaR can be efficiently treated with linear optimization algorithms [12].

29.2.3 Market neutrality

The market itself constitutes a risk factor. If the instruments in the portfolio are positively correlated with the market, then the portfolio would follow not only market growth but also market drops. Naturally, portfolio managers are willing to avoid situations of the second type by constructing portfolios that are uncorrelated with the market, or are *market neutral*. To be market uncorrelated, the portfolio must have zero beta,

$$\beta_p = \sum_{i=1}^{n} \beta_i x_i = 0,$$

where $x_1, \ldots, x_n$ denote the proportions in which the total portfolio capital is distributed among n assets and β_i are betas of individual assets,

$$\beta_i = \frac{\mathrm{Cov}(r_i, r_M)}{\mathrm{Var}(r_M)},$$

where r_M stands for market rate of return. Instruments' betas, β_i, can be estimated, for example, using historical data:

$$\beta_i = \left(\sum_{j=1}^{J} (r_{M,j} - \bar{r}_M)^2 \right)^{-1} \sum_{j=1}^{J} (r_{i,j} - \bar{r}_i)(r_{M,j} - \bar{r}_M),$$

where J is the number of historical observations and $\bar{r}$ denotes the sample average, $\bar{r} = J^{-1} \sum r_j$. As a proxy for market returns r_M, historical returns of the S&P 500 index can be used.

In our case study, we investigate the effect of constructing a market neutral (zero-beta) portfolio by including a market neutrality constraint in the portfolio optimization problem. We compare the performance of the optimal portfolios obtained with and without this market neutrality constraint.

29.2.4 Problem formulation

This section presents the generic problem formulation, which was used to construct an optimal portfolio. We suppose that some historical sample path of returns of n instruments is available. Based on this sample path, we calculate the expected return of the portfolio and the various risk measures for that portfolio. We maximize the expected return of the portfolio subject to different operating, trading, and risk constraints,

$$\max_x \quad E\left[\sum_{i=1}^{n} r_i x_i \right] \tag{29.2}$$

subject to

$$0 \le x_i \le 1, \quad i = 1, \ldots, n, \tag{29.3}$$

$$\sum_{i=1}^{n} x_i \le 1, \tag{29.4}$$

$$\Phi_{\mathrm{Risk}}(x_1, \ldots, x_n) \le \omega, \tag{29.5}$$

$$-k \le \sum_{i=1}^{n} \beta_i x_i \le k, \tag{29.6}$$

where x_i is the portfolio position (weight) of asset i, r_i is the (random) rate of return, and β_i is market beta of instrument i.

The objective function (29.2) represents the expected return of the portfolio. The first constraint (29.3) of the optimization problem imposes limitations on the amount of funds

invested in a single instrument. (We do not allow short positions.) The second constraint (29.4) is the budget constraint. Constraints (29.5) and (29.6) control risks of financial losses. The key constraint in the presented approach is the risk constraint (29.5). Function $\Phi_{\text{Risk}}(x_1, \ldots, x_n)$ represents either a CVaR or a CDaR risk measure, and risk tolerance level ω is the fraction of the portfolio value that is allowed for risk exposure.

Constraint (29.6), with β_i representing market's beta for instrument i, forces the portfolio to be market neutral in the zero-beta sense, i.e., the portfolio correlation with the market is bounded. The coefficient k in (29.6) is a small number that sets the portfolio's beta close to zero. To investigate the effects of imposing a zero-beta requirement on the portfolio-rebalancing algorithm, we solved the optimization problem with and without this constraint. Constraint (29.6) significantly improves the out-of-sample performance of the algorithm.

The risk measures considered in this paper allow for formulating the risk constraint (29.5) in terms of linear inequalities, which makes the optimization problem (29.2)–(29.6) linear, given the linearity of objective function and other constraints. Below we present the explicit form of the risk constraint (29.5) for CVaR and CDaR risk measures.

29.2.5 CVaR constraint

The loss function incorporated into CVaR constraint is the negative portfolio's return,

$$f(\mathbf{x}, \mathbf{y}) = -\sum_{i=1}^{n} r_i x_i, \tag{29.7}$$

where the vector of instruments' returns $\mathbf{y} = \mathbf{r} = (r_1, \ldots, r_n)$ is random. The risk constraint (29.5), $\phi_\alpha(\mathbf{x}) \le \omega$, where the CVaR risk function replaces the function $\Phi_{\text{Risk}}(\mathbf{x})$, is

$$\zeta + \frac{1}{(1-\alpha)J} \sum_{j=1}^{J} \max\left\{0, -\sum_{i=1}^{n} r_{ij} x_i - \zeta\right\} \le \omega, \tag{29.8}$$

where r_{ij} is return of the ith instrument in scenario j, $j = 1, \ldots, J$. Since the loss function (29.7) is linear, the risk constraint (29.8) can be equivalently represented by the linear inequalities

$$\begin{aligned} &\zeta + \frac{1}{1-\alpha}\frac{1}{J}\sum_{j=1}^{J} w_j \le \omega, \\ &-\sum_{i=1}^{n} r_{ij} x_i - \zeta \le w_j, \quad j = 1, \ldots, J, \\ &\zeta \in \mathrm{R}, \quad w_j \ge 0, \quad j = 1, \ldots, J. \end{aligned} \tag{29.9}$$

This representation allows for reducing the optimization problem (29.2)–(29.6) with the CVaR constraint to a linear programming problem.

29.2.6 CDaR constraint

The CDaR risk constraint $\Delta_\alpha(\mathbf{x}) \le \omega$ has the form

$$\eta + \frac{1}{1-\alpha}\frac{1}{J}\sum_{j=1}^{J}\max\left[0, \max_{1\le k\le j}\left\{\sum_{i=1}^{n}\left(\sum_{s=1}^{k} r_{is}\right)x_i\right\} - \sum_{i=1}^{n}\left(\sum_{s=1}^{j} r_{is}\right)x_i - \eta\right] \le \omega,$$

and it can be reduced to a set of linear constraints similarly to the CVaR constraint.

29.3 Case study: Portfolio of hedge funds

The case study investigates investment opportunities and tests portfolio management strategies for a portfolio of hedge funds. Hedge funds are subject to less regulation compared with mutual or pension funds. Hence, very little information on a hedge funds' activities is publicly available. (For example, many funds report their share prices only monthly.) On the other hand, fewer regulations and weaker government control provide more room for aggressive, risk-seeking trading and investment strategies. As a consequence, the revenues in this industry are on average much higher than elsewhere, but the risk exposure is also higher. (For example, the typical life of a hedge fund is about five years, and very few of them perform well in the long run.) Data availability and sizes of data sets impose challenging requirements on portfolio rebalancing algorithms. Also, the specific nature of hedge fund securities imposes some limitations on using them in trading or rebalancing algorithms. For example, hedge funds are far from being perfectly liquid: hedge funds may *not* be publicly traded or may be closed to new investors. From this point of view, our results contain a rather schematic representation of investment opportunities existing in the hedge fund market and do not give direct recommendations on investing in that market. The goal of this study is to compare recently developed risk management approaches and to demonstrate their high numerical efficiency in a realistic setting.

The data set used for conducting the numerical experiments was provided by the Foundation for Managed Derivatives Research. It contained monthly data for more than 5000 hedge funds, from which we selected those with significantly long history and some minimum level of capitalization. To pass the selection, a hedge fund should have 66 months of historical data from December, 1995, to May, 2001, and its capitalization should be at least \$5 million at the beginning of this period. The total number of funds that satisfied these criteria and accordingly constituted the investment pool for our algorithm was 301. In this data set, the field with the names of hedge funds was unavailable; therefore, we identified the hedge funds with numbers, i.e., HF 1, HF 2, and so on. The historical returns from the data set were used to generate scenarios for algorithm (29.2)–(29.6). Each scenario is a vector of monthly returns for all securities involved in the optimization, and all scenarios are assigned equal probabilities.

We performed separate runs of the optimization problem (29.2)–(29.5), with and without constraint (29.6) with CVaR and CDaR risk measure in constraint (29.5), varying such parameters as confidence levels, risk tolerance levels, etc.

The case study consisted of two sets of numerical experiments. The first set of *in-sample* experiments included the calculation of efficient frontiers and the analysis of the optimal portfolio structure for each of the risk measures. The second set of experiments,

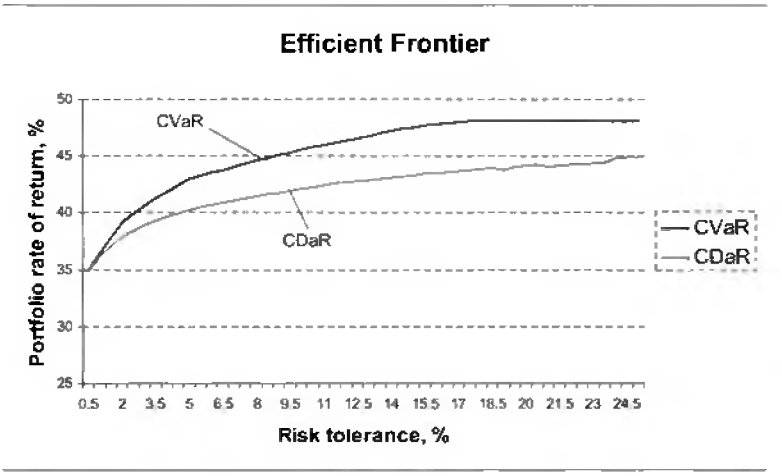


Figure 29.3. *Efficient frontiers for portfolios with various risk constraints. The market neutrality constraint is inactive.*

out-of-sample testing, was designed to demonstrate the performance of our approach in a simulated historical environment.

29.3.1 In-sample results

Efficient frontier For constructing the efficient frontier for the optimal portfolio with different risk constraints, we solved the optimization problem (29.2)–(29.5) with different risk tolerance levels ω in constraint (29.5), varied from $\omega = 0.005$ to $\omega = 0.25$. The parameter α in CVaR and CDaR risk constraints was set to $\alpha = 0.90$. The efficient frontier is presented in Figure 29.3, where the portfolio rate of return means expected yearly rate of return. In these runs, the market neutrality constraint (29.6) is inactive.

For optimal portfolios, in the sense of problem (29.2)–(29.5), there exists an upper bound (equal to 48.13%) for the portfolio's rate of return. Optimal portfolios with the CVaR constraint reach this bound at about 18% risk tolerance level, but the CDaR-constrained portfolio does not achieve the maximal expected return within the given range of ω values. CDaR is a relatively conservative constraint imposing requirements not only on the magnitude of loses but also on the time sequence of losses. (Small consecutive losses may lead to large drawdown, without a significant increase of CVaR.)

Figure 29.4 presents efficient frontiers of an optimal portfolio (29.2)–(29.5) with active market neutrality constraint (29.6), where coefficient k equals 0.01. Imposing the extra constraint (29.6) causes a decrease in the in-sample optimal expected return. For example, the saturation level of the portfolio's expected return is now 41.94%, and both portfolios reach that level at much lower values of risk tolerance ω. However, the market neutrality constraint almost does not affect the curves of efficient portfolios in the leftmost points of efficient frontiers, which correspond to the lowest values of risk tolerance ω.

Quite high rates of return for CVaR- and CDaR-efficient portfolios are explained by

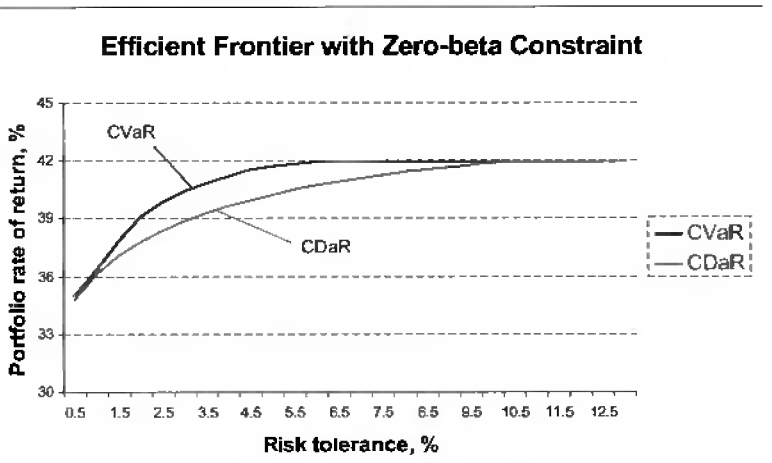


Figure 29.4. *Efficient frontier for market neutral portfolio with various risk constraints* ($k = 0.01$).

Table 29.1. *Portfolio weights for optimal portfolio with CVaR and CDaR constraints.*

	HF49	HF84	HF93	HF100	HF126	HF169	HF196	HF209	HF219	HF231	HF258	HF259	HF298
CVaR	4.39%	0.00%	8.75%	7.37%	0.87%	1.01%	1.56%	22.47%	25.93%	16.92%	1.42%	8.94%	0.38%
CDaR	11.02%	4.19%	8.14%	6.63%	0.00%	5.43%	0.00%	21.47%	13.72%	18.30%	3.41%	5.87%	1.83%

the fact that 301 funds, selected to form the optimal portfolios, constitute about 6% of the initial hedge fund pool and already are "the best of the best" in our data sample.

Optimal portfolio configuration We now discuss the structure of the optimal portfolio with various risk constraints. We selected those optimal portfolios on the efficient frontiers whose expected return is equal to 35%. (The market neutrality constraint is not active.)

Table 29.1 shows the configuration (portfolio weights) of the optimal portfolios with CVaR and CDaR constraints. Among the 301 available instruments, only a few of them contribute to constructing the optimal portfolio. Moreover, a closer look at Table 29.1 shows that nearly two-thirds of the portfolio value for both risk measures is formed by three hedge funds, HF 209, HF 219, and HF 231. In general, CVaR- and CDaR-optimal portfolios have quite similar structures.

29.3.2 Out-of-sample calculations

The out-of-sample testing of the portfolio optimization algorithm (29.2)–(29.6) sheds light on the actual performance of the approaches. The question is how well the algorithms with

different risk measures utilize the scenario information based on past history in producing a successful portfolio management strategy. An answer can be obtained, for instance, by interpreting the results of the preceding section as follows. Suppose we were back in May, 2001, and we would like to invest a certain amount of money in a portfolio of hedge funds to deliver the highest reward under a specified risk level. Then, according to in-sample results, the best portfolio would be the one on the efficient frontier of a particular rebalancing strategy. In fact, such a portfolio offers the best return-to-risk ratio *provided that the historical distribution of returns will repeat in the future.*

To estimate the actual performance of the optimization approach, we used part of the data for scenario generation and the rest for evaluating the performance of the strategy.

We present the results of a plain out-of-sample test, where the older data are considered as the in-sample data for the algorithm and the newer data are treated as the to-be-realized future. First, we took the 12 monthly returns from December, 1995, to November, 1996, as the initial historical data for constructing the first portfolio to invest in, and we observed the portfolio's realized value by observing the historical prices for December, 1996. Then we added one more month, December, 1996, to the data which were used for scenario generation (12 months of historical data in total) to generate an optimal portfolio and to allocate to investments in January, 1997, and so on. Note that we did not implement the moving window method for out-of-sample testing, where the same number of scenarios (i.e., the most recent historical points) is used for solving the portfolio rebalancing problem. Instead, we accumulated the historical data for portfolio optimization.

First, we perform the out-of-sample runs for each risk measure in constraint (29.5) for different values of risk tolerance level ω (market neutrality constraint (29.6) is inactive). Figures 29.5 and 29.6 illustrate the historical trajectories of the optimal portfolio under different risk constraints. (The portfolio values are given in % relative to the initial portfolio value.) Risk tolerance level ω was set to 0.005, 0.01, 0.03, 0.05, 0.10, 0.12, 0.15, 0.17, and 0.20, but for better reading of figures, we report only results with $\omega = 0.005$, 0.01, 0.05, 0.10, and 0.15. The parameter α, which is the risk confidence level in CDaR and CVaR constraints, was set to $\alpha = 0.90$.

Figures 29.7 and 29.8 show that risk constraint (29.5) has a significant impact on the algorithm's out-of-sample performance. Earlier, we had also observed that this constraint has a significant impact on the in-sample performance. Constraining risk in the in-sample optimization decreases the optimal value of the objective function, and the results reported in the preceding subsection reflect this. The risk constraints force the algorithm to favor less profitable but safer decisions over more profitable but "dangerous" ones. Imposing extra constraints always reduces the feasibility set and consequently leads to lower optimal objective values. However, the situation changes dramatically for an out-of-sample application of the optimization algorithm. The numerical experiments show that constraining risks improves the overall performance of the portfolio rebalancing strategy in out-of-sample runs; tighter in-sample risk constraint may lead to both lower risks and higher out-of-sample returns. For both risk measures, loosening the risk tolerance (i.e., increasing ω values) results in an increased volatility of the out-of-sample portfolio returns and, after exceeding some threshold value, in degradation of the algorithm's performance, especially during the last 13 months (March, 2000–May, 2001). For all risk functions in constraint (29.5), the most attractive portfolio trajectories are obtained for risk tolerance level $\omega = 0.005$, which means that these portfolios have high returns (high final portfolio value), low volatility, and low

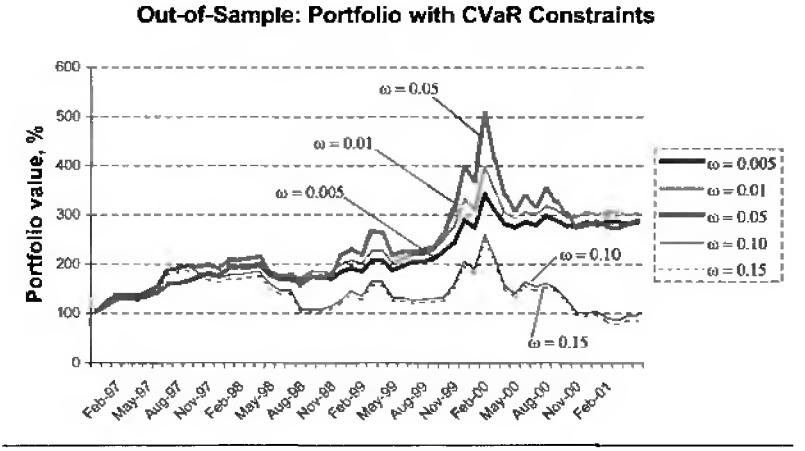


Figure 29.5. *Historical trajectories of optimal portfolio with CVaR constraints.*

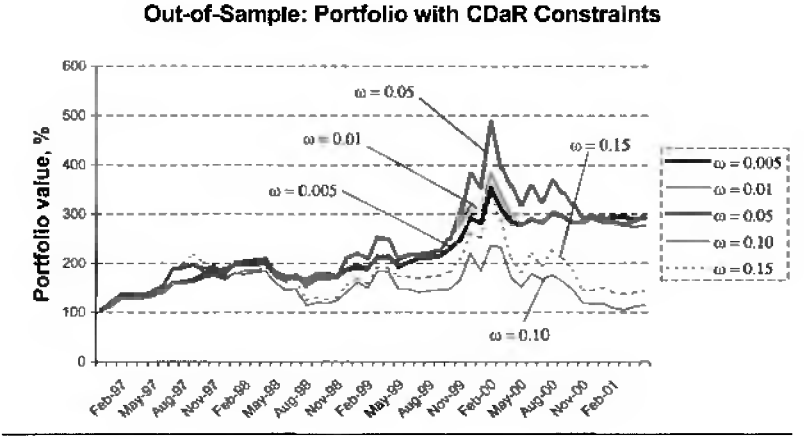


Figure 29.6. *Historical trajectories of optimal portfolio with CDaR constraints.*

drawdowns. Increasing ω to 0.01 leads to a slight increase of the final portfolio value, but it also increases portfolio volatility and drawdowns, especially for the second quarter of 2001. For larger values of ω the portfolio returns deteriorate, and for all risk measures portfolio curves with $\omega = 0.10$ show quite poor performance. Further increasing the risk tolerance to $\omega = 0.15$ in some cases allows for achieving higher returns at the end of 2000, but after this high peak the portfolio suffers severe drawdowns.

Figures 29.7 and 29.8 illustrate the effects of imposing market neutrality constraint (29.6) in addition to risk constraint (29.5). The primary purpose of (29.6) is making the portfolio uncorrelated with the market. The main idea of composing a market neutral portfolio is protecting it from market drawdowns. Figures 29.7 and 29.8 compare the trajectories of market neutral and without-risk-neutrality optimal portfolios. Additional

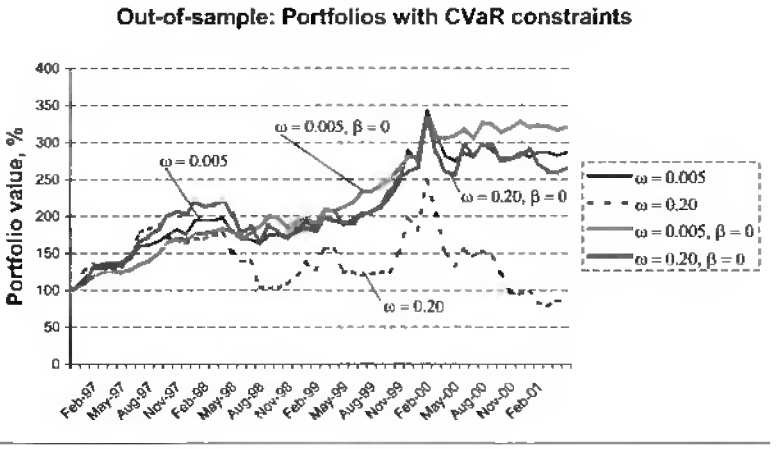


Figure 29.7. *Historical trajectories of optimal portfolio with CVaR constraints. Lines with $\beta = 0$ correspond to portfolios with market neutral constraint.*

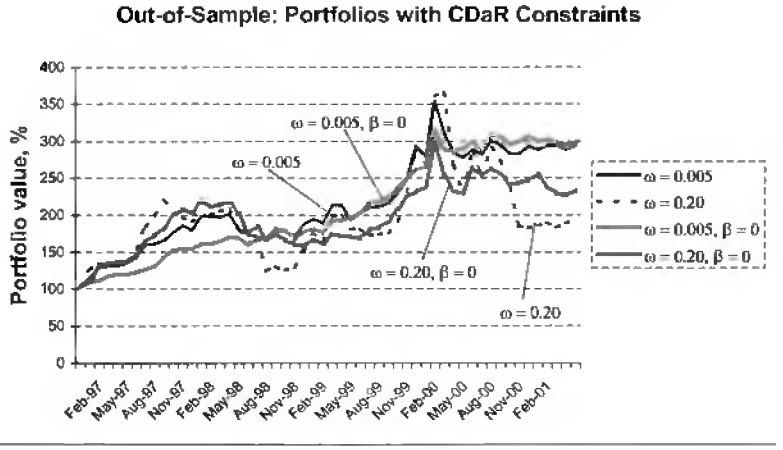


Figure 29.8. *Historical trajectories of optimal portfolio with CDaR constraints. Lines with $\beta = 0$ correspond to portfolios with market neutral constraint.*

constraining resulted in most cases in a further improvement of the portfolio's out-of-sample performance. To clarify how the risk-neutrality condition (29.6) influences the portfolio's performance, we displayed only figures for lowest and highest values of the risk tolerance parameter, namely for $\omega = 0.005$ and $\omega = 0.20$. Coefficient k in (29.6) was set to $k = 0.01$, and instruments' betas β_i were calculated by correlating with the benchmark S&P 500 index. For portfolios with tight risk constraints ($\omega = 0.005$), imposing market neutrality constraint (29.6) straightened their trajectories (reduced volatility and drawdowns), which made the historic curves almost monotone with a positive slope. On top of that, portfolios with the market neutrality constraint had a higher final portfolio value, compared to those

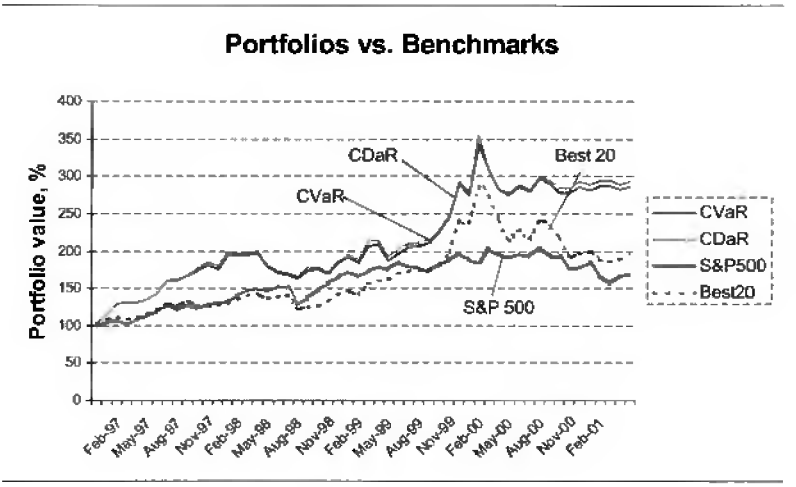


Figure 29.9. *Performance of the optimal portfolios with various risk constraints versus the S&P* 500 *index and a benchmark portfolio combined from the* 20 *best hedge funds. Risk tolerance level* $\omega = 0.005$, *parameter* $\alpha = 0.90$. *Market neutrality constraint is inactive.*

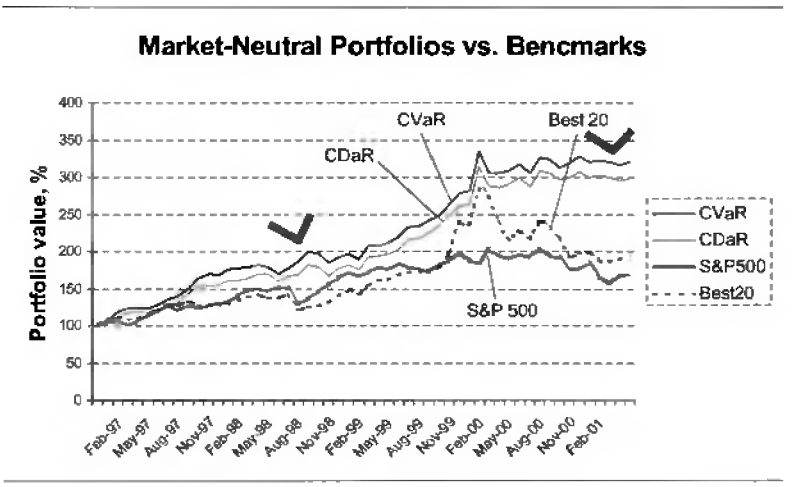


Figure 29.10. *Performance of market neutral optimal portfolio with various risk constraints versus the S&P* 500 *index and a benchmark portfolio combined from the* 20 *best hedge funds. Risk tolerance level is* $\omega = 0.005$, *parameter is* $\alpha = 0.90$.

without market neutrality. Also, for portfolios with loose risk constraints ($\omega = 0.20$), imposing a market neutrality constraint had a positive effect on the form of their trajectories, dramatically reducing volatility and drawdowns.

Finally, Figures 29.9 and 29.10 demonstrate the performance of the optimal portfolios versus two benchmarks: (1) S&P 500 index and (2) Best20, representing a portfolio dis-

tributed equally among the best 20 hedge funds. These 20 hedge funds include funds with the highest expected monthly returns calculated with past historical information. Similar to the optimal portfolios (29.2)–(29.6), the Best20 portfolio was rebalanced monthly (without risk constraints).

According to Figures 29.9 and 29.10, CVaR and CDaR constrained portfolios, both without and with a market neutrality condition, outperform benchmarks, which provides evidence of high efficiency of the risk constrained portfolio management algorithm (29.2)–(29.6). Also, we would like to emphasize the behavior of market neutral portfolios in down market conditions. Two marks on Figure 29.10 indicate the points at which two risk-constrained portfolios gained positive returns while the market was falling. Also, all risk-constrained portfolios seem to withstand the down market in 2000, when the market experienced significant drawdown. This demonstrates the efficiency and appropriateness of risk management approaches considered in this paper.

The Best20 benchmark evidently lacks the solid performance of its competitors. It not only significantly underperforms all the portfolios constructed with algorithm (29.2)–(29.6), but also underperforms the market half of the time. Unlike portfolios (29.2)–(29.6), the Best20 portfolio pronouncedly follows the market drop in the second half of 2000, and, moreover, it suffers much more severe drawdowns than the market does. This indicates that the risk constraints in the algorithm (29.2)–(29.6) play an important role in selecting the funds.

Summarizing, we emphasize the general inference about the role of risk constraints in the out-of-sample and in-sample application of an optimization algorithm, which can be drawn from our experiments: risk constraints decrease the in-sample returns, while out-of-sample performance may be improved by adding risk constraints, and, moreover, stronger risk constraints usually ensure better out-of-sample performance.

29.4 Conclusions

We tested the performance of a portfolio allocation algorithm with different types of risk constraints in an application for managing a portfolio of hedge funds. As the risk measure in the portfolio optimization problem, we used conditional value at risk and conditional drawdown at risk. We combined these risk constraints with the market neutrality (zero-beta) constraint, making the optimal portfolio uncorrelated with the market.

The numerical experiments consist of in-sample and out-of-sample testing. We generated efficient frontiers and compared algorithms with various constraints. The out-of-sample part of the experiments was performed in two setups, which differed in constructing the scenario set for the optimization algorithm.

The results obtained are data set–specific and we cannot make direct recommendations on portfolio allocations based on these results. However, we learned several lessons from this case study. Imposing risk constraints may significantly degrade in-sample expected returns while improving risk characteristics of the portfolio. In-sample experiments showed that for tight risk tolerance levels, all risk constraints produce relatively similar portfolio configurations. Imposing risk constraints may improve the out-of-sample performance of the portfolio-rebalancing algorithms in the sense of risk-return trade-off. Especially promising results can be obtained by combining several types of risk constraints. In particular, we

combined the market neutrality (zero-beta) constraint with CVaR or CDaR constraints. We found that tightening of risk constraints greatly improves portfolio dynamic performance in out-of-sample tests, increasing the overall portfolio return and decreasing both losses and drawdowns. In addition, imposing the market neutrality constraint adds to the stability of the portfolio's return and reduces portfolio drawdowns. Both CDaR and CVaR risk measures demonstrated a solid performance in out-of-sample tests.

Acknowledgment

We thank the Foundation for Managed Derivatives Research for providing the data set for conducting numerical experiments and for partial financial support of this case study.

Bibliography

[1] C. Acerbi, C. Nordio, and C. Sirtori, *Expected Shortfall as a Tool for Financial Risk Management*, 2001; available online from http://www.gloriamundi.org.

[2] C. Acerbi and D. Tasche, *On the Coherence of Expected Shortfall*, 2001; available online from http://www.gloriamundi.org.

[3] C. Ackermann, R. McEnally, and D. Ravenscraft, *The performance of hedge funds: Risk, return and incentives*, J. Finance, 54 (1999), pp. 833–874.

[4] G. Amin and H. Kat, *Hedge fund performance* 1990–2000: *Do the "money machines" really add value?*, J. Financial and Quant. Anal., 38 (2003), pp. 251–274.

[5] F. Andersson, H. Mausser, D. Rosen, and S. Uryasev, *Credit risk optimization with conditional value-at-risk criterion*, Math. Program. Ser. B, 89 (2001), pp. 273–291.

[6] P. Artzner, F. Delbaen, J. M. Elber, and D. Heath, *Thinking coherently*, Risk, 10 (1997), pp. 68–71.

[7] P. Artzner, F. Delbaen, J. M. Elber, and D. Heath, *Coherent measures of risk*, Math. Finance, 9 (1999), pp. 203–228.

[8] BARRA RogersCasey, *Market Neutral Investing*, Barra, Berkeley, CA, 2000.

[9] S. Brown and W. Goetzmann, *Hedge Funds with Style*, Working Paper 00-29, Yale International Center for Finance, New Haven, CT, 2000.

[10] D. R. Cariño and W. T. Ziemba, *Formulation of the Russell Yasuda Kasai financial planning model*, Oper. Res., 46 (1998), pp. 433–449.

[11] D. R. Cariño, D. H. Myers, and W. T. Ziemba, *Concepts, technical issues and uses of the Russell-Yasuda Kasai model*, Oper. Res., 46 (1998), pp. 450–462.

[12] A. Chekhlov, S. Uryasev, and M. Zabarankin, *Portfolio Optimization with Drawdown Constraints*, Research Report 2000-5, ISE Department, University of Florida, Gainesville, FL, 2000.

[13] G. Consigli and M. A. H. Dempster, *Solving dynamic portfolio problems using stochastic programming*, Z. Angew. Math. Mech., 775 (1997), pp. 565–566.

[14] G. Consigli and M. A. H. Dempster, *Dynamic stochastic programming for asset-liability management*, Ann. Oper. Res., 81 (1998), pp. 131–161.

[15] R. Dembo and A. King, *Tracking models and the optimal regret distribution in asset allocation*, Appl. Stoch. Models Data Anal., 8 (1992), pp. 151–157.

[16] A. Duarte, Jr., *Fast computation of efficient portfolios*, J. Risk, 1 (1999), pp. 1–24.

[17] D. Duffie and J. Pan, *An overview of value-at-risk*, J. Derivatives, 4 (1997), pp. 7–49.

[18] P. Embrechts, A. McNeil, and D. Straumann, *Correlation and dependency in risk management: Properties and pitfalls*, in Risk Management: Value at Risk and Beyond, M. Dempster, ed., Cambridge University Press, Cambridge, UK, 2001.

[19] W. Fung and D. Hsieh, *Empirical characteristics of dynamic trading strategies: The case of hedge funds*, Rev. Financial Stud., 10 (1997), pp. 275–302.

[20] W. Fung and D. Hsieh, *Performance characteristics of hedge funds and commodity funds: Natural vs. spurious biases*, J. Financial Quantitative Anal., 35 (2000), pp. 291–307.

[21] W. Fung and D. Hsieh, *The risk in hedge fund strategies: Theory and evidence from trend followers*, Rev. Financial Stud., 14 (2001), pp. 313–341.

[22] W. V. Harlow, *Asset allocation in a downside-risk framework*, Financial Anal. J., Sep/Oct (1991), pp. 28–40.

[23] P. Jorion, *Value-at-Risk: The New Benchmark for Controlling Market Risk*, McGraw–Hill, New York, 1997.

[24] H. Konno and S. Shirakawa, *Equilibrium relations in a capital asset market: A mean absolute deviation approach*, Financial Engrg. Japanese Markets, 1 (1994), pp. 21–35.

[25] H. Konno and A. Wijayanayake, *Mean-absolute deviation portfolio optimization model under transaction costs*, J. Oper. Res. Soc. Japan, 42 (1999), pp. 422–435.

[26] H. Konno and H. Yamazaki, *Mean absolute deviation portfolio optimization model and its application to Tokyo stock market*, Management Sci., 37 (1991), pp. 519–531.

[27] P. Krokhmal, J. Palmquist, and S. Uryasev, *Portfolio optimization with conditional value-at-risk objective and constraints*, J. Risk, 4 (2002), pp. 43–68.

[28] M. I. Kusy and W. T. Ziemba, *A bank asset and liability management model*, Oper. Res., 34 (1986), pp. 356–376.

[29] F.-S. LHABITANT, *Assessing Market Risk for Hedge Funds and Hedge Funds Portfolios*, Research Paper 24, Union Bancaire Privée, Geneva, Switzerland, 2001.

[30] A. LO, *Risk management for hedge funds: Introduction and overview*, Financial Anal. J., 57 (2001), pp. 16–33.

[31] H. M. MARKOWITZ, *Portfolio selection*, J. Finance, 7 (1952), pp. 77–91.

[32] H. M. MARKOWITZ, *Portfolio Selection: Efficient Diversification of Investments*, Blackwell, New York, 1991.

[33] H. MAUSSER AND D. ROSEN, *Beyond VaR: From measuring risk to managing risk*, ALGO Res. Quart., 1 (1998), pp. 5–20.

[34] R. T. ROCKAFELLAR, *Convex Analysis*, Princeton Math. Ser. 28, Princeton University Press, Princeton, NJ, 1970.

[35] R. T. ROCKAFELLAR AND S. URYASEV, *Optimization of conditional value-at-risk*, J. Risk, 2 (2000), pp. 21–41.

[36] R. T. ROCKAFELLAR AND S. URYASEV, *Conditional value-at-risk for general loss distributions*, J. Banking Finance, 26 (2002), pp. 1443–1471.

[37] M. PRITSKER, *Evaluating value-at-risk methodologies*, J. Financial Services Res., 12 (1997), pp. 201–242.

[38] W. OGRYCZAK AND A. RUSZCZYNSKI, *From stochastic dominance to mean-risk model*, Eur. J. Oper. Res., 116 (1999), pp. 33–50.

[39] F. STAUMBAUGH, *Risk and value-at-risk*, Eur. Management J., 14 (1996), pp. 612–621.

[40] C. TESTURI AND S. URYASEV, *On Relation between Expected Regret and Conditional Value-at-Risk*, Research Report 2000-9, ISE Department, University of Florida, Gainesville, FL, 2000.

[41] A. L. TURNER, C. STACY, D. R. CARIÑO, D. H. MYERS, K. WATANABE, M. SYLVANUS, T. KENT, AND W. T. ZIEMBA, *The Russell-Yasuda Kasai model: An asset/liability model for a Japanese insurance company using multistage stochastic programming*, Interfaces, 24 (1994), pp. 29–49.

[42] M. R. YOUNG, *A minimax portfolio selection rule with linear programming solution*, Management Sci., 44 (1998), pp. 673–683.

[43] S. A. ZENIOS, *A model for portfolio management with mortgage-backed securities*, Ann. Oper. Res., 43 (1993), pp. 337–356.

[44] S. A. ZENIOS, *High performance computing for financial planning: The last ten years and the next*, Parallel Comput., 25 (1999), pp. 2149–2175.

[45] W. T. ZIEMBA AND R. G. VICKSON, EDS., *Stochastic Optimization Models in Finance*, Academic Press, New York, 1975.

[46] W. T. Ziemba, ed., *The Stochastic Programming Approach to Asset, Liability and Wealth Management*, AIMR, Charlottesville, VA, 2003.

[47] W. T. Ziemba and M. J. Mulvey, eds., *Worldwide Asset and Liability Modeling*, Cambridge University Press, Cambridge, UK, 1998.

This page intentionally left blank

Chapter 30

Stochastic Unit Commitment in Hydrothermal Power Production Planning

Nicole Gröwe-Kuska and Werner Römisch**

30.1 Introduction

Economic needs and the ongoing liberalization of European electricity markets stimulate the interest of power utilities in developing models and optimization techniques for the generation and trading of electric power under uncertainty. Utilities participating in deregulated markets observe increasing uncertainty in load (i.e., demand for electric power) and prices for fuel and electricity on spot and contract markets. The mismatch between actual and predicted power demand may be supplied by the power system or by trading activities. The competitive environment forces the utilities to rate alternatives within a few minutes.

In this chapter, we describe a mathematical model for optimal short-term operation and trading of a hydrothermal-based electric utility, which is usually called the *unit commitment problem* because of the important role of the commitment or on/off decisions. Furthermore, we present a methodology for modeling the stochastic data process in the form of a scenario tree and a report on a Lagrangian-based decomposition strategy for solving the optimization model. We also provide some numerical experience obtained from test runs on realistic data from the German utility Vereinigte Energiewerke AG (VEAG). The optimization model has emerged from a collaboration with engineers of VEAG. For our tests we use a configuration of the VEAG system consisting of 25 (coal-fired, gas-burning) thermal units and seven pumped storage hydro units. Its total capacity is about 13,000 megawatts (MW), including a hydro capacity of 1700 MW; the peak loads of the system are about 8600 MW. In contrast to other hydrothermal-based utilities the amount of installed pumped storage capacity enables the inclusion of pumped storage plants into the optimization. It is an additional feature of the VEAG system that, for a weekly planning period, inflows to reservoirs are negligible.

*Institute of Mathematics, Humboldt-University Berlin, 10099 Berlin, Germany (nicole@mathematik.hu-berlin.de, romisch@mathematik.hu-berlin.de).

There is a growing number of contributions to stochastic power system optimization with emphasis on modeling aspects and solution methods. For stochastic models including commitment decisions, see [1, 2, 6, 8, 9, 10, 17, 22, 25, 26, 27, 34, 35, 36, 37]. While [1, 17, 25, 37] propose variants of Lagrangian relaxation methods for their solution and present implementations, the work in [2, 22, 34] is directed to a comparison of stochastic and deterministic power system modeling, further engineering aspects, and industrial applications. The solution methods in [1, 17, 37] differ from each other by the nondifferentiable optimization methods, the subproblem solvers, and the Lagrangian heuristics employed as components of the master algorithm. Modeling issues of stochastic data processes in power systems are addressed in [15, 18, 27]. We also refer to the state-of-the-art survey [12] on scenario (tree) generation and to references therein.

This chapter is organized as follows. In section 30.2 we describe the stochastic unit commitment model and its particular features. Section 30.3 contains a brief description of the solution algorithm based on Lagrangian relaxation. Our strategy for generating scenario trees for the electrical load process is presented in section 30.4. It consists of two parts: simulation of load scenarios using a statistical model for the electrical load process and a method for constructing scenario trees out of simulation scenarios. In section 30.5 we report on the performance of the Lagrangian relaxation algorithm and of the scenario tree generation technique.

30.2 Stochastic power system modeling

We consider a power generation system comprising thermal and hydro units and contracts for delivery and purchase, and we address the unit commitment problem in short-term operation planning. This problem concerns the scheduling of start-up/shutdown decisions and of operation levels for all power units and contracts, respectively, such that the operation costs over the time horizon are minimal. Although our outlook is short term, the uncertainty of important system parameters like electrical load, streamflows in hydro units, and prices for fuel or electricity is a major modeling issue.

Let the planning horizon be discretized into T uniform subintervals, and suppose there are sets $\mathcal{I}$ and $\mathcal{J}$ of thermal and hydro units, respectively. The decision variable of thermal unit $i \in \mathcal{I}$ is (u_i, p_i), where the components of u_i are binary variables taking the value 1 if the unit is on at some time period and 0 if off. The components of p_i are the corresponding operation levels. Contracts for delivery and purchase are regarded as special thermal units. The decision variable of hydro unit $j \in \mathcal{J}$ is (v_j, w_j), where the components of v_j and w_j are the generation and pumping levels over time, respectively.

To formulate a unit commitment model that incorporates fluctuations of uncertain system parameters, we use a probabilistic description of uncertainty. Let

$$\xi = \{\xi_t := (d_t, r_t, \gamma_t, a_t, b_t, c_t)\}_{t=1}^{T}$$

be a discrete-time stochastic process on some probability space $(\Omega, \mathcal{F}, \mathcal{P})$, where ξ_1 is deterministic; d_t, r_t, and γ_t represent the load, the spinning reserve, and the hydro inflows in period t; while a_t, b_t, and c_t collect the cost coefficients.

The scheduling decisions for period t are made *after* having learned the realization of the stochastic data for that period. Denote by $\mathcal{F}_t \subseteq \mathcal{F}$ the σ-field generated by $\{\xi_\tau\}_{\tau=1}^{t}$, i.e.,

the events observable until period t. Since the information on ξ_1 is complete, $\mathcal{F}_1 = \{\emptyset, \Omega\}$, i.e., ξ_1 is deterministic. By assuming $\mathcal{F}_T = \mathcal{F}$ we require full information to be available at the end of the planning horizon. The sequence of scheduling decisions $\{(u_t, p_t, v_t, w_t)\}_{t=1}^T$ also forms a stochastic process on $(\Omega, \mathcal{F}, \mathcal{P})$, which is assumed to be adapted to the filtration of σ-fields, i.e., *nonanticipative*. Nonanticipativity means that the decisions (u_t, p_t, v_t, w_t) may depend on the data observable only until period t or, equivalently, that (u_t, p_t, v_t, w_t) is $\mathcal{F}_t$-measurable.

Assume that the data process $\{\xi_t\}_{t=1}^T$ has a *discrete* probability distribution; i.e., its support consists of a finite number of *scenarios* (or realizations). Then there exist finite subsets $\mathcal{E}_t$ of the σ-algebra $\mathcal{F}_t$, $t = 1, \ldots, T$, such that $\mathcal{E}_t$ is a partition of Ω and that the smallest σ-algebra containing $\mathcal{E}_t$ is just $\mathcal{F}_t$. Using conditional expectations with respect to (w.r.t.) $\mathcal{F}_t$, the nonanticipativity conditions may be formulated as linear equality constraints. As $\mathcal{F}_t \subseteq \mathcal{F}_{t+1}$, every element of $\mathcal{E}_t$ can be represented as the union of certain elements of $\mathcal{E}_{t+1}$. Since the numbers of elements of $\mathcal{E}_t$, i.e., $|\mathcal{E}_t|$, and of scenarios of ξ_t coincide, the relations between the elements of $\mathcal{E}_t$ and of $\mathcal{E}_{t+1}$ for $t = 1, \ldots, T-1$ may be represented in the form of a tree, called a *scenario tree*. Let $\mathcal{N} = \{1, \ldots, |\mathcal{N}|\}$ denote the set of nodes of the tree. The *root* node $n = 1$ stands for period $t = 1$. Every other node n has a unique *predecessor* node n_-. Let path(n) be the set $\{1, \ldots, n_-, n\}$ of nodes from the root to node n and $t(n) := |\text{path}(n)|$ denote the time period associated with node n. Then the nodes in $\mathcal{N}_t := \{n : t(n) = t\}$ correspond to the realizations of ξ_t, $t = 1, \ldots, T$. Nodes n belonging to the set $\mathcal{N}_T$ are called *leaves*. A scenario corresponds to a path from the root to some leaf, i.e., to path(n) for some $n \in \mathcal{N}_T$. Furthermore, let $\mathcal{N}_+(n)$ denote the set of *successors* to node n. Hence, $\mathcal{N}_T = \{n : \mathcal{N}_+(n) = \emptyset\}$. With the given scenario probabilities $\{\pi_n\}_{n\in\mathcal{N}_T}$, we associate a probability π_n to each node n by the recursion $\pi_n := \sum_{n_+\in\mathcal{N}_+(n)} \pi_{n_+}$, $n \in \mathcal{N}$. Clearly, $\sum_{n\in\mathcal{N}_t} \pi_n = 1$ holds for each $t = 1, \ldots, T$. Let $\mathcal{N}_{\text{first}} := \cup_{t=1}^{t_1} \mathcal{N}_t$ be the set of *first-stage* nodes, where t_1 is the maximal period such that the data process $\{\xi_t\}_{t=1}^{t_1}$ is deterministic, i.e., the sets $\mathcal{N}_t$, $t = 1, \ldots, t_1$, are singletons. We use the following notation for the sequence of predecessors of any node $n \in \mathcal{N}$: $n_0 := n$, $n_{-1} := n_-$ if $n > 1$, $n_{-(\kappa+1)} := (n_{-\kappa})_-$ if $t(\kappa) > 1$. See Figure 30.1 for a sample scenario tree.

We use the notation $\{\xi^n = (d^n, r^n, \gamma^n, a^n, b^n, c^n)\}_{n\in\mathcal{N}}$ and $\{(u^n, p^n, v^n, w^n)\}_{n\in\mathcal{N}}$ for the scenario trees representing the stochastic data process ξ and the (stochastic) decision process (u, p, v, w), respectively. The decisions (u^n, p^n, v^n, w^n) assigned to nodes n in $\mathcal{N}_t$ are the realizations of the stochastic decisions (u_t, p_t, v_t, w_t) for $t = 1, \ldots, T$. For the commitment decisions u_i we set $u_i^{\text{path}(n)} := (u_i^v)_{v\in\text{path}(n)}$. To handle initial values of the u_i we introduce a starting time $t_{\text{ini}} \le 0$, set $n_\kappa := \kappa - t(n)$ for $\kappa = t(n) + t_{\text{ini}}, \ldots, t(n)$, and assume that initial values $u_i^{1-\kappa}$ for $\kappa = t_{\text{ini}}, \ldots, 0$, $i \in \mathcal{I}$, are given. The *fuel costs* for operating the thermal unit i at node n are

$$C_i^n(p_i^n, u_i^n) := \max_{l=1,\ldots,\bar{l}} \left\{a_{il}^n p_i^n + b_{il}^n u_i^n\right\}$$

with coefficients a_{il}^n and b_{il}^n such that $C_i^n(\cdot, 1)$ is convex and increasing on $\mathbb{R}_+$. The *start-up costs* of unit i at node n depend on its downtime; they may vary from a maximum cold-start value to a much smaller value when the unit is still relatively close to its operating

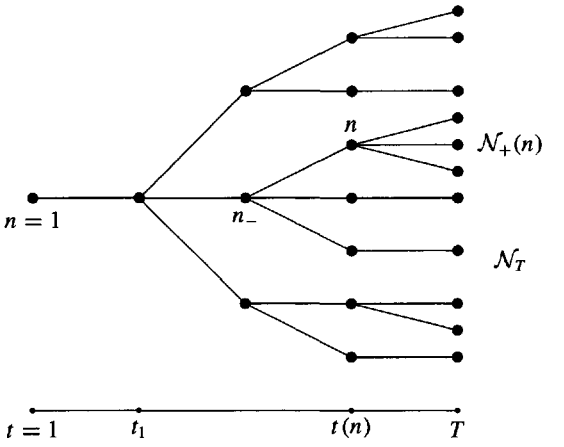


Figure 30.1. *Scenario tree with* $t_1 = 2$, $T = 5$, $|\mathcal{N}| = 23$, *and* 11 *leaves.*

temperature. This is modeled by

$$S_i^n\left(u_i^{\text{path}(n)}\right) := \max_{\tau=0,\dots,t_i^c} c_{i\tau}^n \left(u_i^n - \sum_{\kappa=1}^{\tau} u_i^{n-\kappa}\right),$$

where $0 < c_{i0}^n < \cdots < c_{it_i^c}^n$ are cost coefficients, t_i^c is the cooldown time, and $c_{it_i^c}^n$ the maximum cold-start costs of unit $i \in \mathcal{I}$. Since the operating costs of hydro units are negligible in short-term planning, the expected total system costs are given by the sum of fuel and start-up costs of all thermal units

$$\sum_{n\in\mathcal{N}} \pi_n \sum_{i\in\mathcal{I}} \left[C_i^n(p_i^n, u_i^n) + S_i^n\left(u_i^{\text{path}(n)}\right)\right]. \tag{30.1}$$

The operation of all thermal units is described by certain operating ranges and minimum up/downtime requirements, namely, by the inequalities

$$p_{it(n)}^{\min} u_i^n \le p_i^n \le p_{it(n)}^{\max} u_i^n, \quad u_i^n \in \{0,1\}, \quad n \in \mathcal{N},\ i \in \mathcal{I}, \tag{30.2a}$$

$$u_i^{n-\kappa} - u_i^{n-(\kappa+1)} \le u_i^n, \quad \kappa = 1,\dots,\bar{t}_i - 1,\ n \in \mathcal{N},\ i \in \mathcal{I}, \tag{30.2b}$$

$$u_i^{n-(\kappa+1)} - u_i^{n-\kappa} \le 1 - u_i^n, \quad \kappa = 1,\dots,\underline{t}_i - 1,\ n \in \mathcal{N},\ i \in \mathcal{I}, \tag{30.2c}$$

where $p_{it}^{\min}$ and $p_{it}^{\max}$ are the minimum and maximum capacities of unit i at period t, and (30.2b), (30.2c) mean that unit i must remain on (off) for at least $\bar{t}_i$ (and $\underline{t}_i$, respectively) periods if it is switched on (off). The operating ranges and dynamics of hydro units are described by the constraints

$$0 \le v_j^n \le v_{jt(n)}^{\max},\ 0 \le w_j^n \le w_{jt(n)}^{\max},\ 0 \le l_j^n \le l_{jt(n)}^{\max},\ n \in \mathcal{N},\ j \in \mathcal{J}, \tag{30.3a}$$

$$l_j^n = l_j^{n-} - v_j^n + \eta_j w_j^n + \gamma_j^n, \quad n \in \mathcal{N},\ j \in \mathcal{J}, \tag{30.3b}$$

$$l_j^0 = l_j^{\text{in}}, \quad l_j^n = l_j^{\text{end}}, \quad n \in \mathcal{N}_T,\ j \in \mathcal{J}, \tag{30.3c}$$

where $v_{jt}^{\max}$ and $w_{jt}^{\max}$ are the maximum capacities for the generation and pumping of hydro unit $j \in \mathcal{J}$ at period t, γ_j^n is the water inflow to reservoir j at node n, and l_j^n is the reservoir *storage volume* at the end of period $t(n)$, with upper bound $l_{jt(n)}^{\max}$. By η_j we denote the *pumping efficiency*, and by l_j^{in} and l_j^{end} the initial and final volumes, respectively, of unit j. In our model, we disregard spill and head variation effects. Furthermore, we prefer to prescribe final storage volumes of all hydro units instead of introducing a water value function depending on $\{l_j^n\}_{j\in\mathcal{J},n\in\mathcal{N}_T}$ and of maximizing this water value as part of the objective function (30.1). The basic system requirements are the load and the spinning reserve constraint at each node n

$$\sum_{i\in\mathcal{I}} p_i^n + \sum_{j\in\mathcal{J}} (v_j^n - w_j^n) \geq d^n, \quad n \in \mathcal{N}, \tag{30.4a}$$

$$\sum_{i\in\mathcal{I}} (u_i^n p_{it(n)}^{\max} - p_i^n) \geq r^n, \quad n \in \mathcal{N}, \tag{30.4b}$$

where the constraint (30.4b) means that the total commited capacity at each node n should exceed d^n by a certain amount r^n, e.g., by a fraction of d^n. Figure 30.2 shows a collection of weekly load scenarios. They exhibit typical daily cycles, morning and evening peaks, and night and weekend off-peaks. Constraint (30.4a) implies that the total generation has to follow the load scenario curves.

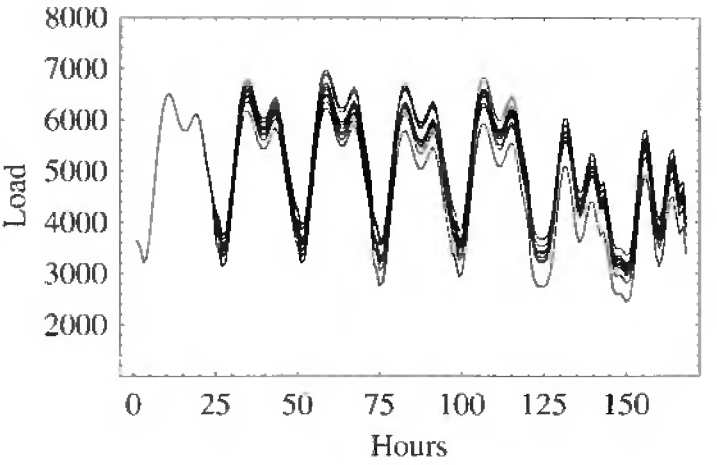


Figure 30.2. *Typical weekly load scenarios.*

The *stochastic unit commitment problem* consists of minimizing the expected costs (30.1) such that the system constraints (30.2), (30.3), and (30.4) are satisfied. This model forms a large-scale linear mixed-integer program involving $|\mathcal{I}||\mathcal{N}|$ binary and $(|\mathcal{I}| + 2|\mathcal{J}|)|\mathcal{N}|$ continuous decision variables and $(2+|\mathcal{J}|)|\mathcal{N}|+|\mathcal{J}||\mathcal{N}_T|$ (in)equality constraints (without taking into account the bounds, the constraints of type (30.2b)–(30.2c), and the objective function). As usual, we call the model (30.1)–(30.4) with fixed binary decisions $\{u^n\}_{n\in\mathcal{N}}$ the *stochastic economic dispatch problem*. We notice that the model (30.1)–(30.4)

is almost separable with respect to the unit index sets $\mathcal{I}$ and $\mathcal{J}$ and only loosely coupled by the $2|\mathcal{N}|$ constraints (30.4). This observation becomes particularly important when recalling that, for a typical weekly time horizon with hourly time steps (i.e., $T = 168$), scenario trees comprise more than $|\mathcal{N}| = 10^3$ nodes and that for midsize generation systems one has $|\mathcal{I}| + |\mathcal{J}| \geq 30$.

30.3 Lagrangian relaxation

Due to the enormous size of the stochastic unit commitment model in the previous section, the use of standard software for mixed-integer linear programs is appropriate for smaller models only. In general, one has to resort to *decomposition* approaches. While the algorithmic realization of *primal* decomposition methods leads to serious obstacles that are impossible to overcome by existing methods (cf. [5]), strategies based on dualizing the model (30.1)–(30.4) appear to be more promising.

Two general *dual* decomposition schemes have been elaborated so far: *scenario* and *nodal* decomposition (see [7, 31]). The first scheme is based on dualizing the nonanticipativity constraints and the second on the dualization of the dynamic constraints. Due to the loose coupling structure of the model, a third scheme comes into play, which is called *geographical* or *component* decomposition in [11, 31]. It is based on assigning Lagrange multipliers to the coupling constraints and on minimizing the corresponding Lagrangian function. The dual problem decomposes into a finite number of (much) smaller stochastic subproblems. The application of all these dualization schemes is justified in convex situations (cf. [30] and [31, section 2.5]).

When comparing dual decomposition approaches for stochastic integer programmming models, two arguments appear to be important: the size of the corresponding duality gaps and the complexity of the dual and of the subproblems. Recent results on a comparison of duality gaps for scenario, nodal, and geographic decomposition indicate that the geographic decomposition has the potential to lead to the smallest duality gaps (see [11]). In the case of the stochastic unit commitment model its subproblems have specific structures that allow for the use of efficient solution algorithms.

We give a brief description of the geographical decomposition or *Lagrangian relaxation* approach and of a Lagrangian-based algorithm for solving (30.1)–(30.4). For a more detailed presentation, see [17]. Let $x := (u, p, v, w)$ denote the decision and $\lambda = (\lambda_1, \lambda_2) := \{(\lambda_1^n, \lambda_2^n)\}_{n\in\mathcal{N}} \in \Lambda := \mathbb{R}_+^{|\mathcal{N}|} \times \mathbb{R}_+^{|\mathcal{N}|}$ the Lagrange multiplier in scenario-tree form to be associated with the coupling constraints (30.4). Then the Lagrangian function is

$$L(x;\lambda) := \sum_{n\in\mathcal{N}} \pi_n \left\{ \sum_{i\in\mathcal{I}} \left[C_i^n(p_i^n, u_i^n) + S_i^n\left(u_i^{\mathrm{path}(n)}\right) \right] \right. \tag{30.5}$$
$$\left. + \lambda_1^n \left[d^n - \sum_{i\in\mathcal{I}} p_i^n - \sum_{j\in\mathcal{J}} (v_j^n - w_j^n) \right] + \lambda_2^n \left[r^n - \sum_{i\in\mathcal{I}} (u_i^n p_{it(n)}^{\max} - p_i^n) \right] \right\},$$

and the dual function and the dual problem are

$$D(\lambda) := \min \{ L(x;\lambda) : x \text{ satisfies the constraints (30.2)–(30.3)} \}, \tag{30.6}$$

$$\max \{D(\lambda) : \lambda \in \Lambda\}. \tag{30.7}$$

Due to the compactness of the constraint sets given by (30.2) and (30.3) for (u, p) and (v, w), respectively, there exists a Lagrangian solution $x(\lambda)$, i.e., a solution of the minimization problem defining $D(\lambda)$ in (30.6) for every $\lambda \in \Lambda$. Under the assumptions made on the fuel costs, the dual function D is concave polyhedral. Hence, the dual (30.7) is solvable if the primal problem (30.1)–(30.4) is feasible. The dual function

$$D(\lambda) = \sum_{i \in \mathcal{I}} D_i(\lambda) + \sum_{j \in \mathcal{J}} \hat{D}_j(\lambda_1) + \sum_{n \in \mathcal{N}} \pi_n \left(\lambda_1^n d^n + \lambda_2^n r^n\right) \tag{30.8}$$

decomposes into the *thermal subproblems*

$$D_i(\lambda) = \min \left\{ \sum_{n \in \mathcal{N}} \pi_n \left[\min_{p_i^n} \left\{ C_i^n(p_i^n, u_i^n) - (\lambda_1^n - \lambda_2^n) p_i^n \right\} \right.\right. \tag{30.9}$$

$$\left.\left. - \lambda_2^n u_i^n p_{it(n)}^{\max} + S_i^n \left(u_i^{\text{path}(n)}\right) \right] : u_i \text{ satisfies (30.2)} \right\}$$

and the *hydro subproblems*

$$\hat{D}_j(\lambda_1) = \min \left\{ \sum_{n \in \mathcal{N}} \pi_n \lambda_1^n (w_j^n - v_j^n) : (v_j, w_j) \text{ satisfies (30.3)} \right\}. \tag{30.10}$$

Both subproblems represent multistage stochastic programming models for the operation of one single unit. While the thermal subproblem (30.9) represents a combinatorial multistage program involving stochastic costs, the hydro subproblem (30.10) is a linear multistage model with stochastic costs and stochastic right-hand sides. The thermal problem (30.9) was solved by (stochastic) dynamic programming. Incorporating the thermal state space into the scenario tree leads to a backward tree recursion for the cost-to-go of all states. This yields the optimal cost-to-go and, by forward tracing the tree, the optimal scheduling decisions $\{(u_i^n(\lambda), p_i^n(\lambda))\}_{n \in \mathcal{N}}$ (see [17, section 3.6] and [24]). For solving the hydro subproblems (30.10) a specialized descent method has been developed. It generates a finite sequence of feasible hydro decisions, where the decision at some step differs from the preceding one only at the nodes of a certain subtree of $\mathcal{N}$. Such a subtree exists for each nonoptimal feasible hydro decision and, hence, the algorithm terminates with an optimal solution $\{(v_j^n(\lambda), w_j^n(\lambda))\}_{n \in \mathcal{N}}$. (See [17, section 3.5], [24], and [25] for details and numerical results.)

The dual problem (30.7) serves as the *master* program. Its iterative solution by a subgradient bundle method leads to a successive decomposition of the primal model (30.1)–(30.4). A subgradient of D at λ is

$$g_D(\lambda) = \left\{ \left(d^n - \sum_{i \in \mathcal{I}} p_i^n(\lambda) - \sum_{j \in \mathcal{J}} (v_j^n(\lambda) - w_j^n(\lambda)), r^n - \sum_{i \in \mathcal{I}} (u_i^n(\lambda) p_{it(n)}^{\max} - p_i^n(\lambda)) \right) \right\}_{n \in \mathcal{N}}.$$

The *proximal bundle method* [14, 20, 21] is used for solving the dual. Starting from an arbitrary point $\lambda^1 = \bar{\lambda}^1 \in \Lambda$, this method generates a sequence $\{\lambda^k\}_{k \in \mathbb{N}}$ in Λ that converges

to some dual solution, and trial points $\bar{\lambda}^k$ for evaluating the solutions $x(\bar{\lambda}^k)$ of (30.6), the subgradients $g_D(\bar{\lambda}^k)$ of D and its linearizations

$$D^k(\cdot) := D(\bar{\lambda}^k) + \langle \cdot - \bar{\lambda}^k, g_D(\bar{\lambda}^k) \rangle \geq D(\cdot),$$

where $\langle \lambda, \mu \rangle := \sum_{n \in \mathcal{N}} \pi_n (\lambda_1^n \mu_1^n + \lambda_2^n \mu_2^n)$ is the dual pairing on Λ. Iteration k uses the polyhedral model $D_k(\cdot) := \min_{l \in N^k} D^l(\cdot)$ with $k \in N^k \subset \{1, \ldots, k\}$ for finding the next trial point $\bar{\lambda}^{k+1}$ as a solution of the quadratic subproblem

$$\max \left\{ D_k(\lambda) - \frac{1}{2} \rho_k |\lambda - \lambda^k|^2 : \lambda \in \Lambda \right\}, \tag{30.11}$$

where the proximity weight $\rho_k > 0$ and the penalty term $|\cdot|^2 := \langle \cdot, \cdot \rangle$ should keep $\bar{\lambda}^{k+1}$ close to the prox-center λ^k. An ascent step to $\lambda^{k+1} = \bar{\lambda}^{k+1}$ occurs if $D(\bar{\lambda}^{k+1}) \geq D(\lambda^k) + \kappa \delta_k$, where $\kappa \in (0, 1)$ is a fixed Armijo-like parameter and $\delta_k := D_k(\bar{\lambda}^{k+1}) - D(\lambda^k) \geq 0$ is the predicted ascent (if $\delta_k = 0$, then λ^k is a solution and the method may stop). Otherwise, a null step $\lambda^{k+1} = \lambda^k$ improves the next model D_{k+1} with the new linearization D^{k+1}. The stopping criterion $\delta_k \leq \text{opt\_tol}(1 + D(\lambda^k))$ and the choices of the weights ρ_k and of the index set N^{k+1}, in particular its upper bound NGRAD, are discussed in [14, 20] (see also [17, section 3.4]).

The optimal value $D(\lambda^*)$ of (30.7) resulting from the bundle method provides a lower bound for the optimal costs of the model (30.1)–(30.4). In general, however, the dual optimal scheduling decisions $x(\lambda^*) = (u(\lambda^*), p(\lambda^*), v(\lambda^*), w(\lambda^*))$ violate the load and reserve constraints (30.4) such that a low-cost primal feasible solution has to be determined by a *Lagrangian heuristic*. Two Lagrangian heuristics (see [17, section 3.7] and [24]) that determine nearly optimal first-stage decisions $\{(u^n, p^n, v^n, w^n)\}_{n \in \mathcal{N}_{\text{first}}}$ starting from the optimal multiplier λ^* and the Lagrangian solution $x(\lambda^*)$ have been developed. The first heuristic LH1 is based on a combination of a water rescheduling procedure and a known thermal heuristic [38] applied to a (deterministic) unit commitment model where the stochastic quantities ξ, λ^*, and $l(\lambda^*)$ are replaced by their mean values. Clearly, this heuristic provides a nearly optimal decision at nodes $n \in \mathcal{N}_{\text{first}}$ only. The second heuristic LH2 starts by finding some $\varepsilon > 0$ such that $x(\lambda^* + \varepsilon \underline{1})$ ($\underline{1}$ being the element in Λ with unit components) is feasible. Taking $u(\lambda^* + \varepsilon \underline{1})$ as a starting point, a finite sequence of binary decisions is constructed such that their components are decreasing. This is done by selecting a node $n \in \mathcal{N}$, where the available reserve capacity $\sum_{i=1}^{I} (u_i^n p_{it(n)}^{\max} - p_i^n) - r^n$ is maximal, and switching some unit i off at node n and at some predecessor and successor nodes, where the unit i and the neighboring nodes of n are detected by stochastic dynamic programming. Next, the corresponding stochastic economic dispatch problem is reformulated as a hydro problem with piecewise linear costs and solved by a modification of the descent method mentioned earlier (see [17, section 3.5], [24], and [25]). This procedure, which generates a sequence of scheduling decisions at all nodes, is continued until infeasibility is detected during economic dispatch and terminates with a nearly optimal solution at each node in $\mathcal{N}$.

The whole Lagrangian relaxation approach is based on the same, but stochastic, ingredients as in the classical deterministic unit commitment situation [16, 32]: a solver for the nondifferentiable dual subproblem solvers and some Lagrangian heuristic. The interaction of these ingredients is illustrated in Figure 30.3.

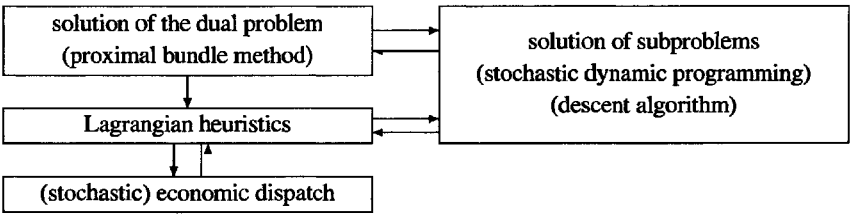


Figure 30.3. *Scheme of the Lagrangian relaxation algorithm.*

30.4 Load scenario trees

30.4.1 A statistical model for the electrical load

The identification of a statistical model for the electric load of the VEAG generation system is based on an hourly load profile for a period of three years (1098 days). A plot of the hourly load data is displayed in Figure 30.4. The historical load records show seasonal variations caused by meteorological factors like temperature, cloud cover, etc. In the weekly and monthly load data there are recurring patterns of length 24 (one day) and of length 168 (one week). The periodic patterns complete themselves within the calendar year and are then repeated on a yearly basis. Interruptions of this regularity are caused by customs like public holidays or the start/end of daylight saving time. Thus, in principle the electric load depends on the category of the day (Monday,..., Sunday, public holiday, etc.) and on the season. Figure 30.4 highlights the periodic components of our historical data.

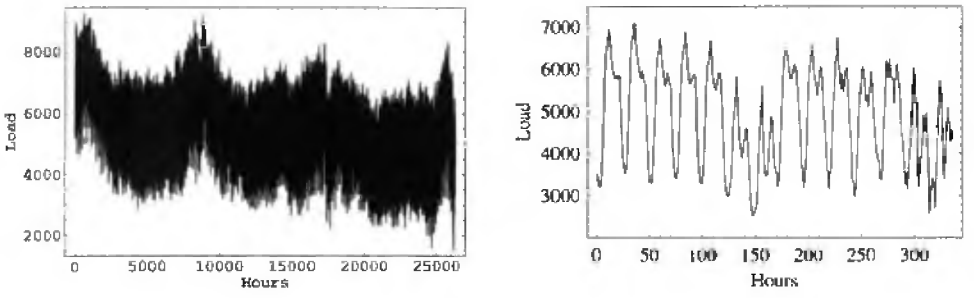


Figure 30.4. *Hourly load data: Three years (left) and two weeks (right).*

In a first step, days of a similar load pattern are identified using daily load records (24 load data of a day). To each such record we assign a day category (1 for a Monday record, ..., 7 for a Sunday, 8 for a public holiday following a working day, 9 for days between holidays and weekends, 10 for a public holiday following a weekend or a holiday). Clustering methods from S-PLUS 4.5 are applied to answer the question of whether the records can be grouped or classified into useful or informative clusters. After eliminating seasonal effects of the load records, clustering and ANOVA tests lead to a classification of the load records into eight categories (see Table 30.1).

The statistical modeling of the load process exploits the decomposition of the load

Table 30.1. *Categories of daily load records.*

Category	Definition
1	Monday or working day after a public holiday
2	working day (Tuesday, Wednesday, Thursday)
3	Friday or working day before a public holiday
4	Saturday
5	Sunday
6	public holiday not following days of the categories 2, 3
7	public holiday following days of the categories 2, 3
8	working day between days of the categories 4–7

process into a *daily mean load process* and a *mean-corrected load series*, which are treated separately. Let $x_{j\tau}$ be the observed load at time period $\tau = 1, \ldots, 24$ of day $j \in \mathrm{J} := \{1, \ldots, 1098\}$ (i.e., record j of the data base), $\overline{d}_j := \frac{1}{24} \sum_{\tau=1}^{24} x_{j\tau}$ the mean load of day j, and $\mathrm{cat}(j)$ the category of day j according to Table 30.1. Then the historical load records are decomposed according to

$$x_{j\tau} = d_{j\tau} + \overline{d}_j \quad (\tau = 1, \ldots, 24;\ j \in \mathrm{J}), \tag{30.12}$$

where $d_{j\tau}$, $\tau = 1, \ldots, 24$, is the mean corrected load record of day $j \in \mathrm{J}$. The daily mean load series versus the day number is plotted in Figure 30.5.

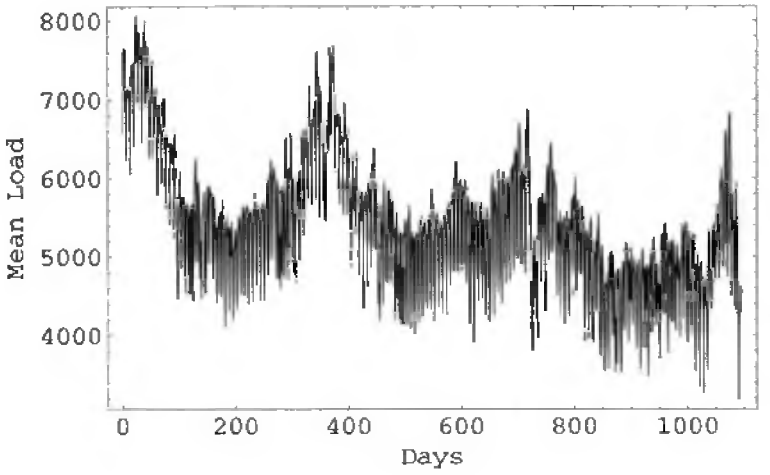


Figure 30.5. *Daily mean load versus the day number.*

Figure 30.6 displays the mean-corrected load records $(d_{j\tau})_{\mathrm{cat}(j)=k}$ for days of category $k = 2$ and $k = 5$.

The mean load depends on the category of the day and on the season. Further, there is an interaction between the mean load and meteorological factors like temperature and cloud cover. The meteorological impact on the daily mean demand could not be modeled

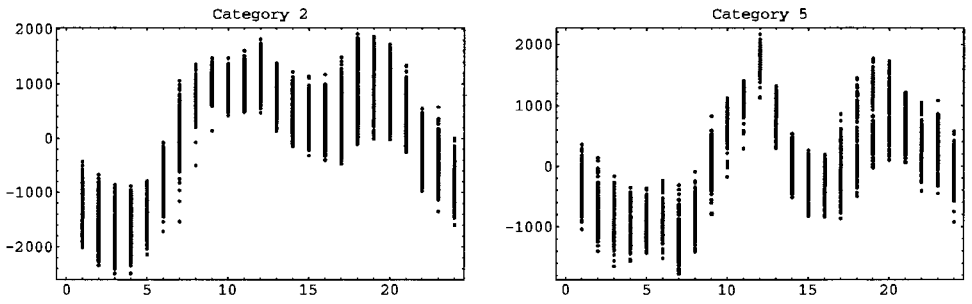


Figure 30.6. *Mean-corrected load records for days of category* 2 *(left) and* 5 *(right).*

because of missing meteorological parameters.

To select an appropriate class of (time series) models for the daily mean load series $\{\overline{d}_j\}_{j\in J}$ with $J \subset \mathbb{Z}$, $\{\overline{d}_j\}_{j\in J}$ is considered as a part of a realization of the stochastic mean load process $\{\overline{\mathbf{d}}_j\}_{j\in\mathbb{Z}}$. Data analysis methods [3] are used to detect any *seasonal* (periodic) or *trend* (nonconstant mean) components, outlying observations, or sharp changes in behavior. Then suitable transformations are applied to the data to obtain a new stationary series (*residuals*) with zero mean and unit variance. The trend and seasonal components may be removed by estimating these components and subtracting them from the data. Another transformation is called *differencing*; it replaces the original process by differences of the process at t and at $t - s$ for some *lag* $s \in \mathbb{N}$ and eliminates a seasonal component of period s. The mean load series $\{\overline{d}_j\}_{j\in J}$ clearly contains a recurring pattern with the seasonal period of 365 (one year). There are further periodic components of length 7 (one week) and change points due to the start/end of daylight saving time. Irregularities of the weekly patterns have been removed from the time series by replacing outlying observations by the value of the nearest day of the same category.

Many approaches for fitting a time series to the deseasonalized data rely on classical linear models. *Autoregressive moving average* (ARMA) models are characterized by finite-order linear difference equations with constant coefficients. A real stochastic process $\{X_t\}_{t\in\mathbb{Z}}$ is called ARMA(p, q) if it is stationary, i.e., if $\mathbb{E}[X_t^2] < \infty$, $\mathbb{E}[X_t]$ is constant and $\mathbb{E}[X_r X_s] = \mathbb{E}[X_{r+t} X_{s+t}]$ $\forall r, s, t \in \mathbb{Z}$, and

$$X_t - \phi_1 X_{t-1} - \cdots - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + \cdots + \theta_q Z_{t-q} \quad \forall t \in \mathbb{Z}, \tag{30.13}$$

where ϕ_k, $k = 1, \ldots, p$, and θ_l, $l = 1, \ldots, q$, are real coefficients and $\{Z_t\}_{t\in\mathbb{Z}}$ is the *white noise* process WN$(0, \sigma^2)$ with zero mean and variance σ^2, i.e., $\mathbb{E}[Z_t] = 0$, $\mathbb{E}[Z_t^2] = \sigma^2$ $\forall t \in \mathbb{Z}$, and $\mathbb{E}[Z_r Z_t] = 0$ if $r \neq t$. Using the *backward shift operator* B defined by $B^\ell X_t := X_{t-\ell}$ for $t, \ell \in \mathbb{Z}$, the ARMA equations (30.13) can be rewritten as

$$\phi(B) X_t = \theta(B) Z_t \quad \forall t \in \mathbb{Z}, \quad \{Z_t\} \sim \text{WN}(0, \sigma^2),$$

where ϕ and θ denote the polynomials $\phi(z) = 1 - \phi_1 z - \cdots - \phi_p z^p$, $\theta(z) = 1 + \theta_1 z + \cdots + \theta_q z^q$. An ARMA$(p, q)$ process $\{X_t\}_{t\in\mathbb{Z}}$ is said to be *causal* if there exists a real sequence $\{\psi_\ell\}$ such that $\sum_{\ell=0}^{\infty} \psi_\ell < \infty$ and $X_t = \sum_{\ell=0}^{\infty} \psi_\ell Z_{t-\ell}$ $\forall t \in \mathbb{Z}$.

Seasonal autoregressive integrated moving average (SARIMA) models are defined as follows. The process $\{X_t\}_{t\in\mathbb{Z}}$ is said to be a SARIMA$(p, d, q) \times (P, D, Q)_S$ *process*

with period S if the differenced process $Y_t := (1-B)^d(1-B^S)^D X_t$ is the causal ARMA process

$$\phi(B)\Phi(B^S)Y_t = \theta(B)\Theta(B^S)Z_t, \quad \{Z_t\} \sim WN(0,\sigma^2),$$

where $\phi(z) = 1 - \cdots - \phi_p z^p$, $\Phi(z) = 1 - \cdots - \Phi_P z^P$, $\theta(z) = 1 + \cdots + \theta_q z^q$, and $\Theta(z) = 1 + \cdots + \Theta_Q z^Q$. Hence, the model for $\{X_t\}_{t\in\mathbb{Z}}$ reads $\phi(B)\Phi(B^S)(1-B)^d(1-B^S)^D X_t = \theta(B)\Theta(B^S)Z_t$.

To identify a suitable SARIMA model for a given time series, the differencing orders d, D, the model orders p, P, q, Q, and the length S of the seasonal component must be identified. They can be discovered by inspecting the empirical autocorrelation function, the empirical counterpart of the autocorrelation function $\mathbb{E}[X_\ell X_0]$, $\ell \in \mathbb{Z}$; see [3]. The model coefficients $\{\phi_\ell\}_{\ell=1}^p$, $\{\Phi_\ell\}_{\ell=1}^P$, $\{\theta_\ell\}_{\ell=1}^q$, $\{\Theta_\ell\}_{\ell=1}^Q$, and the white noise variance σ^2 can be estimated via parameter estimation procedures for ARMA processes. The maximum likelihood method produces the most efficient estimates in the special case of Gaussian time series. Initial values for the model coefficients can be obtained by the Hannan–Rissanen algorithm (cf. [3, section 5]), which solves the problem of order selection and parameter estimation for ARMA processes simultaneously.

In case of the daily mean load process, stationary residuals were obtained after three differencing operations (two lag-364 differencing operations followed by one lag-1 differencing). The residuals were treated as part of a realization of the stochastic process

$$\{Y_j := \overline{\mathbf{d}}_j - \overline{\mathbf{d}}_{j-1} - 2\overline{\mathbf{d}}_{j-364} + 2\overline{\mathbf{d}}_{j-365} + \overline{\mathbf{d}}_{j-728} - \overline{\mathbf{d}}_{j-729}\}.$$

For $\{Y_j\}$ the Hannan–Rissanen algorithm from the Mathematica Time Series Pack selected an ARMA(1,1) model that served as an initial model for the maximum likelihood method. The maximum likelihood estimates for the model coefficients and random noise process led to the time series model

$$Y_j - \hat{\phi}_1 Y_{j-1} = Z_j + \hat{\theta}_1 Z_{j-1}, \quad j \in \mathbb{Z}, \quad \text{where}$$
$$\hat{\phi}_1 = 0.357756, \quad \hat{\theta}_1 = -0.639978, \quad \{Z_j\} \sim N(0, 15533.88), \quad j \in \mathbb{Z}.$$

Accordingly, for the daily mean load process $\{\overline{\mathbf{d}}_j\}_{j\in\mathbb{Z}}$ we obtain the SARIMA$(1,1,1)\times(0,2,0)_{364}$ model

$$(1-B)(1-B^{364})^2(1-\hat{\phi}_1 B)\overline{\mathbf{d}}_j = (1+\hat{\theta}_1 B)Z_j, \tag{30.14}$$

which can be converted into the general ARMA(730, 1) model

$$\begin{aligned}\overline{\mathbf{d}}_j - (1+\hat{\phi}_1)\overline{\mathbf{d}}_{j-1} + \hat{\phi}_1\overline{\mathbf{d}}_{j-2} - 2\overline{\mathbf{d}}_{j-364} + 2(1+\hat{\phi}_1)\overline{\mathbf{d}}_{j-365} - 2\hat{\phi}_1\overline{\mathbf{d}}_{j-366} \\ + \overline{\mathbf{d}}_{j-728} + (\hat{\phi}_1 - 1)\overline{\mathbf{d}}_{j-729} + \hat{\phi}_1\overline{\mathbf{d}}_{j-730} = Z_j + \hat{\theta}_1 Z_{j-1}, \ j \in \mathbb{Z}.\end{aligned} \tag{30.15}$$

For modeling the time dependence of the mean-corrected load records corresponding to days of the same category k, $k = 1, \ldots, 8$, polynomial-based linear regression models of the form

$$\hat{d}_{k\tau} = \sum_{l=0}^{m_k} \beta_{kl}\tau^l + \epsilon_{km_k} \quad (\tau = 1, \ldots, 24) \tag{30.16}$$

have been fitted, where the error term ϵ_{km_k} is normally distributed with zero mean and variance $\sigma^2_{m_k}$. The degree m_k of the polynomials will be fixed later. For model fitting, regression diagnostics, and forecasting we used the statistical package S-PLUS 4.5.

The statistical model for the load is obtained by combining the models for the daily mean load and the mean-corrected load records according to (30.12). The regression models for the mean-corrected load records that correspond to different day categories are included in (30.12) by using day category variables D_{jk}

$$D_{jk} := \begin{cases} 1, & \text{cat}(j) = k, \\ 0 & \text{otherwise}, \end{cases} \qquad (j \in \mathrm{J};\ k = 1, \ldots, 8).$$

With these definitions (30.12) may be rewritten as

$$x_{j\tau} = \sum_{k=1}^{8} D_{jk}\hat{d}_{k\tau} + \overline{d}_j \quad (j \in \mathrm{J};\ \tau = 1, \ldots, 24). \tag{30.17}$$

The different time scales for the historical load records and the load process can be synchronized by an index transformation:

$$d_t = \sum_{k=1}^{8} D_{\lfloor \frac{t}{24} \rfloor, k}\hat{d}_{k,r(t/24)} + \overline{\mathbf{d}}_{\lfloor \frac{t}{24} \rfloor}, \quad t \in \mathbb{Z}, \tag{30.18}$$

where $\lfloor \frac{t}{24} \rfloor$ denotes the lower integer part of $\frac{t}{24}$ and $r(t/24)$ the remainder of t after division by 24. Finally, the statistical model of the load is obtained by inserting (30.15) and (30.16) into (30.18), yielding

$$\begin{aligned} d_t = {} & \sum_{k=1}^{8} D_{\lfloor \frac{t}{24} \rfloor, k} \sum_{l=0}^{m_k} \beta_{kl} \left(r\left(\frac{t}{24}\right) \right)^l \\ & + (1 + \hat{\phi}_1)\overline{\mathbf{d}}_{\lfloor \frac{t}{24} \rfloor - 1} - \hat{\phi}_1 \overline{\mathbf{d}}_{\lfloor \frac{t}{24} \rfloor - 2} + 2\overline{\mathbf{d}}_{\lfloor \frac{t}{24} \rfloor - 364} - 2(1 + \hat{\phi}_1)\overline{\mathbf{d}}_{\lfloor \frac{t}{24} \rfloor - 365} \\ & + 2\hat{\phi}_1 \overline{\mathbf{d}}_{\lfloor \frac{t}{24} \rfloor - 366} - \overline{\mathbf{d}}_{\lfloor \frac{t}{24} \rfloor - 728} - (\hat{\phi}_1 - 1)\overline{\mathbf{d}}_{\lfloor \frac{t}{24} \rfloor - 729} - \hat{\phi}_1 \overline{\mathbf{d}}_{\lfloor \frac{t}{24} \rfloor - 730} \\ & + \sum_{k=1}^{8} D_{\lfloor \frac{t}{24} \rfloor, k}\epsilon_{km_k} + Z_{\lfloor \frac{t}{24} \rfloor} + \hat{\theta}_1 Z_{\lfloor \frac{t}{24} \rfloor - 1} \quad (t \in \mathbb{Z}). \end{aligned} \tag{30.19}$$

To select the degrees m_k of the regression polynomials we measured the squared distance between (30.19) and the historical load data for the third year. The best fit was obtained for $m_k = 10, k = 1, \ldots, 8$.

The stochastic model (30.19) for the electrical load is used to simulate a number of load scenarios for the time horizon $\{1, \ldots, T\}$ by employing a (pseudo) random number generator for the independent normal random variables $\epsilon_{km_k}, k = 1, \ldots, 8$, and $Z_j, j \in \mathrm{J}$. In this way, S load scenarios $\{d^i\}_{i=1}^S$ are generated which have identical probabilities $p_i = \frac{1}{S}$, $i = 1, \ldots, S$, and coincide for $t = 1, \ldots, t_1$. Hence, the nodes at $t = 1, \ldots, t_1$ are the first-stage nodes of a specific scenario tree forming a *fan* of individual scenarios. Clearly, this tree could be used as input of the optimization algorithm described in section 30.3. Such a tree contains a relatively large number of nodes, namely, $|\mathcal{N}| = t_1 + (T - t_1)S$.

30.4.2 Construction of scenario trees

Next we describe a general methodology that successively reduces the number of nodes of a fan $\xi = \{\xi^i\}_{i=1}^S$ of individual scenarios by modifying the tree structure and by bundling similar scenarios. This methodology is based on a successive scenario reduction technique developed in [13, 19]. The idea is to compare the probability distance of original and reduced trees and to delete scenarios if the reduced tree is still close enough to the original one. The probability distance has to be chosen such that the underlying stochastic program behaves stably with respect to this distance when changing the probability distribution. Here, stability means that the optimal costs and solution sets behave continuously with respect to such changes; see [13, 29].

In the context of stochastic power scheduling models, we use the Kantorovich distance D_K of (multivariate) probability distributions (cf. [28, section 5]). For discrete probability distributions with finitely many scenarios the distance D_K is just the optimal value of a linear transportation problem. Let P denote the probability distribution of ξ with scenarios ξ^i and probabilities p_i, $i = 1, \ldots, S$, and Q that with scenarios $\tilde{\xi}^j$ and probabilities q_j, $j = 1, \ldots, \tilde{S}$. Then

$$D_K(P, Q) = \inf \left\{ \sum_{i=1}^{S} \sum_{j=1}^{\tilde{S}} \eta_{ij} c_T(\xi^i, \tilde{\xi}^j) : \eta_{ij} \geq 0, \sum_{i=1}^{S} \eta_{ij} = q_j, \sum_{j=1}^{\tilde{S}} \eta_{ij} = p_i \ \forall i, \ \forall j \right\}, \tag{30.20}$$

where c_t is defined by $c_t(\xi^i, \tilde{\xi}^j) := \sum_{\tau=1}^{t} |\xi_\tau^i - \tilde{\xi}_\tau^j|$ for each $t = 1, \ldots, T$.

Now, let Q be the probability distribution of a reduced tree of ξ; i.e., the support of Q consists of scenarios ξ^j for $j \in \{1, \ldots, S\} \setminus J$ and J denotes some index set of deleted scenarios. For fixed $J \subset \{1, \ldots, S\}$, the scenario tree Q_* based on the scenarios $\{\xi^j\}_{j \notin J}$ having minimal D_K-distance to P may be computed explicitly [13, Theorem 3.1]. The minimal distance is

$$D_K(P, Q_*) = \sum_{i \in J} \pi_i \min_{j \notin J} c_T(\xi^i, \xi^j), \tag{30.21}$$

and the probability q_j^* of scenario ξ^j, $j \notin J$, of Q_* is given by the rule

$$q_j^* := p_j + \sum_{i \in J(j)} p_i, \ J(j) := \{i \in J : j = j(i)\}, \ j(i) \in \arg\min_{j \notin J} c_T(\xi^i, \xi^j) \ \forall i \in J. \tag{30.22}$$

This means that the scenario ξ^j of the reduced tree represents the bundle $\{\xi^i\}_{i \in J(j)}$ of original scenarios, and its probability is given by formula (30.22). Since the solution sets of $\min_{j \notin J} c_T(\xi^i, \xi^j)$ are nonunique in general, the optimally reduced tree is not uniquely determined.

Our approach for constructing a scenario tree ξ_{app} that approximates the original fan ξ of individual scenarios consists of a successive reduction procedure by applying the above reduction argument recursively to (sub)trees on the time horizons $\{1, \ldots, t\}$ for $t = T, \ldots, t_1$. More precisely, given some tolerance $\varepsilon > 0$ and constant $\alpha > 1$, an index set J_t is determined in the $(T - t + 1)$th step such that $|J_{t_1}| = S - 1$ and

$$\sum_{i \in J_t} \pi_i \min_{j \notin J_t} c_t(\xi^i, \xi^j) \leq \frac{\varepsilon(\alpha - 1)}{\alpha^{T-t+1}} \quad (t = T, \ldots, t_1) \tag{30.23}$$

by the simultaneous backward reduction algorithm [19, Algorithm 2.2]. While J_t is the index set of deleted scenarios in the $(T-t+1)$th step, the index set of remaining scenarios is denoted by $I_t, t = t_1, \ldots, T+1$; i.e., it holds that $I_t \cup J_t = I_{t+1}, t = t_1, \ldots, T, |I_{t_1}| = 1$, and $I_{T+1} = \{1, \ldots, S\}$.

The approximate scenario tree ξ_{app} with sets $\mathcal{N}_t$ of nodes at time period t is then defined by setting $|\mathcal{N}_t| := |I_t|$ and $\{\xi^n_{\text{app}}\}_{n\in\mathcal{N}_t} := \{\xi^j_t\}_{j\in I_t}$ for every $t = t_1, \ldots, T$. According to redistribution rule (30.22) the probability π^j_t of ξ^j_t is recursively given by $\pi^i_{T+1} := p_i$, $i = 1, \ldots, S$, and

$$\pi^j_t := \pi^j_{t+1} + \sum_{i\in J(t,j)} \pi^i_{t+1}, \quad J(t,j) = \{i \in J_t : j = j(t,i)\}, \quad j(t,i) \in \arg\min_{j\notin J_t} c_t(\xi^i, \xi^j),$$

for every $j \in I_t$ and $t = T, \ldots, t_1$. Hence, the approximate tree ξ_{app} exhibits the following structure. It holds that $\xi^n = \xi^n_{\text{app}}$ for every $n \in \mathcal{N}_{\text{first}}$ and $|\mathcal{N}_T| = |I_T|$. The cardinality of $\mathcal{N}_+(n)$ is equal to $|\{j\} \cup J(t,j)|$ if the node $n \in \mathcal{N}_t$ corresponds to the index $j \in I_t$. This means that the index sets $\{J(t,j)\}_{j\in I_t}$ characterize the branching degree of ξ_{app} at period t.

30.5 Numerical results

The Lagrangian relaxation algorithm was implemented in C++ except for the proximal bundle method, for which the Fortran package NOA 3.0 [21] was used as a callable library. For numerical tests we used the hydrothermal power system of VEAG (with $T = 168$, $|\mathcal{I}| = 25$, and $|\mathcal{J}| = 7$) under uncertain load (i.e., the remaining data were deterministic). The test runs were performed on an HP 9000 (780/J280) computer with 180 MHz frequency and 768 MB main memory under HP-UX 10.20.

For testing the performance of the optimization algorithm a bunch of load scenario trees was randomly generated.

The upper part of Table 30.2 contains test results of the Lagrangian relaxation algorithm based on the heuristic LH1 with the parameters opt_tol $= 10^{-3}$ and NGRAD $= 50$ for NOA 3.0. In particular, it provides computing times and gaps for different numbers of scenarios (S) and four randomly generated scenario trees, each having a different number of nodes (N). The gap refers to the relative difference

$$\frac{1}{D_*}\left(\sum_{t=1}^{T}\sum_{i\in\mathcal{I}}[C_{it}(p_{it}, u_{it}) + S_{it}(u_i)] - D_*\right)$$

of the costs of the scheduling decision (u, p, v, w) and the optimal value D_* of the dual. In general, this gap does not provide a quality measure for the approximate first-stage solution (it may even become nonpositive). When reading the computing times in Table 30.2, it is worth recalling that $N = 4000$ and $N = 8000$ correspond to 100,000 and 200,000 binary variables in the model (30.1)–(30.4), respectively.

The lower part of Table 30.2 reports computing times and gaps for the Lagrangian relaxation algorithm based on LH2. Here, the parameters for NOA 3.0 are opt_tol $= 10^{-5}$ and NGRAD $= 200$, and the gap refers to the following bound of the relative duality gap:

$$\frac{1}{D_*}\left(\sum_{n\in\mathcal{N}} \pi_n \sum_{i\in\mathcal{I}}\left[C^n_i(p^n_i, u^n_i) + S^n_i\left(u_i^{\text{path}(n)}\right)\right] - D_*\right).$$

Table 30.2. *Test results for randomly generated load trees.*

Lagrangian relaxation algorithm based on heuristic LH1

S	N	time[s]	gap[%]	N	time[s]	gap[%]
20	1982	89	0.15	1627	94	0.10
20	1651	68	0.37	1805	85	0.07
50	4530	475	0.18	4060	274	0.10
50	4041	313	0.10	4457	288	0.43
100	9230	1183	0.11	9224	1072	0.13
100	7727	930	0.09	8867	1234	0.30

Lagrangian relaxation algorithm based on LH2

S	N	NOA time[s]	total time[s]	gap[%]
1	168	10	16	0.20
5	542	65	101	0.19
10	983	128	230	0.71
17	1786	278	733	0.45
21	2098	351	531	0.39
27	2208	380	8349	0.73
32	2173	359	3337	0.66
39	3848	874	4092	0.82

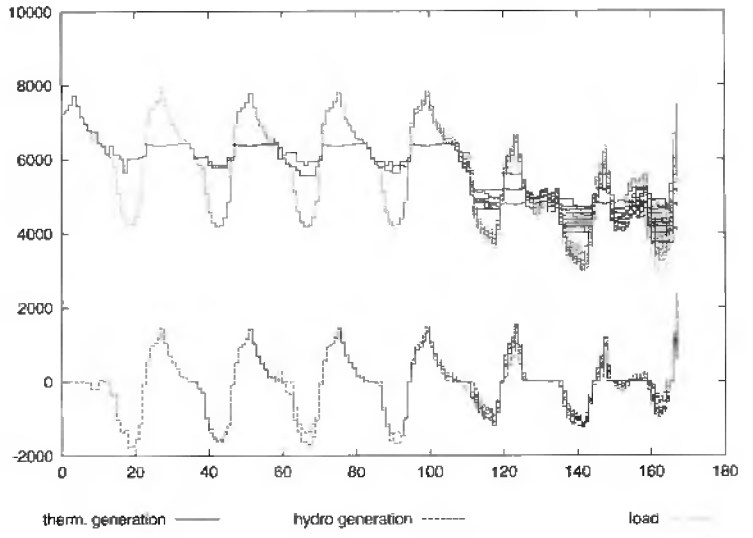


Figure 30.7. *Weekly optimal stochastic solution with* 17 *scenarios.*

This bound provides an accuracy certificate for the approximate primal-feasible solution $\{(u^n, p^n, v^n, w^n))\}_{n\in\mathcal{N}}$.

Figure 30.7 displays the final output of the Lagrangian relaxation algorithm based on LH2 and on a load scenario tree with 17 scenarios and 1786 nodes.

While the deterministic Lagrangian heuristics LH1 requires only short computing times, this becomes quite different for the "stochastic" heuristics LH2. Table 30.2 gives

more insight into the (total) computing times of different test runs. Higher computing times are always due to lots of economic dispatch runs required by LH2. It is worth mentioning here that LH2 is quite sensitive to the accuracy of the dual solution, i.e., to the optimality tolerance of the proximal bundle method. The advantage of using LH1 consists of low running times even for midsize scenario trees, while its drawbacks are that only first-stage solutions are provided with no accuracy bounds. The advantage of LH2 is that it produces a "stochastic" solution together with a guaranteed accuracy bound but at the expense of higher computing times even for scenario trees of smaller size. For further information, see [24].

Another test combined the Lagrangian relaxation algorithm with the load scenario tree generation technique of section 30.4.2. First, we used the statistical model to generate $S = 100$ load scenarios d^i with identical probabilities $p_i = 0.01$ for an hourly discretized time horizon of one week in summer. The accuracy of the load model for the summer season allowed us to choose the first day of the optimization horizon as the first-stage period, i.e., $t_1 := 24$. Hence, the scenario values for the first-stage periods $t = 1, \ldots, 24$ coincide with the load prediction for this period. Given an appropriate number of starting load values, the prediction can be computed from (30.19) by ignoring the realizations of the random noise process $\{Z_t\}$ and of the error terms ϵ_{km_k}. To compute the remaining $T - t_1 = 144$ values of a single scenario from (30.19) we simulated realizations of the random noise process and of the error terms using the random number generators contained in the RANLIBC library [4].

Table 30.3 reports the computing times for solving the stochastic dual (30.7) based on different reduced load scenario trees, each having a different numbers of scenarios (S) and of nodes (N). The trees are constructed by the algorithm in section 30.4.2 for $\alpha := 2$ and different relative tolerances $\varepsilon_{\text{rel}} := \frac{\varepsilon}{\varepsilon_{\max}}$, where $\varepsilon_{\max}$ is the best possible Kantorovich distance D_{K} of the probability distribution $P = 0.01 \sum_{i=1}^{100} \delta_{d^i}$ to one of its scenarios endowed with unit mass. Figure 30.8 shows the improved accuracy of the dual optimum and of the scenario tree structure for different relative tolerances.

Table 30.3. *Test results for solving the stochastic dual based on a reduced load scenario tree of relative tolerance* ε_{rel}.

ε_{rel}	S	N	Variables		Constraints	Nonzeros	time[s]
			Binary	Continuous			
0.6	1	168	4200	7728	16975	44695	7.83
0.1	67	515	12875	23690	52484	137459	17.09
0.05	81	901	22525	41446	91568	240233	37.82
0.01	94	2660	66500	122360	269318	708218	150.14
0.005	96	3811	95275	175306	385583	1014398	291.65
0.001	100	9247	231175	425362	934647	2460402	1176.38

Acknowledgments

This research was supported by the Schwerpunktprogramm Echtzeit-Optimierung grosser Systeme of the Deutsche Forschungsgemeinschaft and by the BMBF-Programm Neue math-

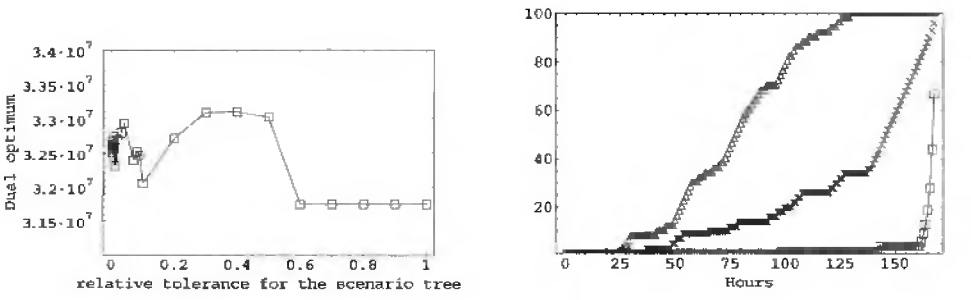


Figure 30.8. *Dual optimum (left) and* $|I_t|$ $(t = 1, \ldots, T)$ *for scenario trees with relative tolerance* $\varepsilon_{\text{rel}} = 0.001$ (△), 0.005 (×), 0.01 (□) *(right).*

ematische Verfahren in Industrie und Dienstleistungen. The authors are grateful to their colleagues D. Dentcheva, M. P. Nowak, and I. Wegner (formerly Humboldt-University Berlin) and to G. Schwarzbach, J. Thomas, and J. Krause (VEAG Vereinigte Energiewerke AG, Berlin) for outstanding cooperation over many years. Further thanks are due to K. C. Kiwiel (Polish Academy of Sciences, Warsaw) for his invaluable contributions to this project during his visits to Berlin as a Fellow of the Alexander von Humboldt Foundation and for the permission to use his NOA 3.0 package, and to J. Dupačová (Charles University Prague) for her collaboration in modeling load profiles.

Bibliography

[1] L. Bacaud, C. Lemaréchal, A. Renaud, and C. Sagastizábal, *Bundle methods in stochastic optimal power management: A disaggregated approach using preconditioners*, Comput. Optim. Appl., 20 (2001), pp. 227–244.

[2] J. H. Bogensperger, *Wochenplanung in Stromhandel und Erzeugung*, in Optimierung in der Energieversorgung III, VDI-Berichte 1508, VDI-Verlag, Düsseldorf, 1999, pp. 183–189.

[3] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*, Springer, New York, 1996.

[4] B. Brown and J. Lovato, *RANLIBC Library of C Routines for Random Number Generation*, Department of Biomathematics, University of Texas, Houston, 1989; available online from http://lib.stat.cmu.edu/general/Utexas/.

[5] C. C. Carøe, *Decomposition in Stochastic Integer Programming*, Ph.D. thesis, Institute of Mathematical Sciences, University of Copenhagen, Copenhagen, 1998.

[6] C. C. Carøe and R. Schultz, *A Two-Stage Stochastic Program for Unit Commitment under Uncertainty in a Hydro-Thermal System*, Preprint SC 98-11, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 1998; available online from http://www.zib.de/pub/pw/index.en.html.

[7] C. C. Carøe and R. Schultz, *Dual decomposition in stochastic integer programming*, Oper. Res. Lett., 24 (1999), pp. 37–45.

[8] P. Carpentier, G. Cohen, J.-C. Culioli, and A. Renaud, *Stochastic optimization of unit commitment: A new decomposition framework*, IEEE Trans. Power Syst., 11 (1996), pp. 1067–1073.

[9] A. J. Conejo, J. M. Arroyo, N. Jiménez Redondo, and F. J. Prieto, *Lagrangian relaxation applications to electric power operations and planning problems*, in Modern Optimisation Techniques in Power Systems, Y. H. Song, ed., Kluwer Academic Publishers, Dordrecht, The Netherlands, 1999, pp. 173–203.

[10] D. Dentcheva and W. Römisch, *Optimal power generation under uncertainty via stochastic programming*, in Stochastic Programming Methods and Technical Applications, K. Marti and P. Kall, eds., Lecture Notes in Econ. and Math. Systems 458, Springer, Berlin, 1998, pp. 22–56.

[11] D. Dentcheva and W. Römisch, *Duality Gaps in Nonconvex Stochastic Optimization*, Preprint 02-5, Institute of Mathematics, Humboldt-University, Berlin, 2002; available online from http://www.mathematik.hu-berlin.de/publ/publ.html.

[12] J. Dupačová, G. Consigli, and S. W. Wallace, *Scenarios for multistage stochastic programs*, Ann. Oper. Res., 100 (2000), pp. 25–53.

[13] J. Dupačová, N. Gröwe-Kuska, and W. Römisch, *Scenario reduction in stochastic programming: An approach using probability metrics*, Math. Program., 95 (2003), pp. 493–511.

[14] S. Feltenmark and K. C. Kiwiel, *Dual applications of proximal bundle methods, including Lagrangian relaxation of nonconvex problems*, SIAM J. Optim., 10 (2000), pp. 697–721.

[15] S.-E. Fleten, S. W. Wallace, and W. T. Ziemba, *Hedging electricity portfolios via stochastic programming*, in Decision Making under Uncertainty: Energy and Power, C. Greengard and A. Ruszczyński, eds., IMA Vol. Math. Appl. 128, Springer, New York, 2002, pp. 71–94.

[16] R. Gollmer, M. P. Nowak, W. Römisch, and R. Schultz, *Unit commitment in power generation—A basic model and some extensions*, Ann. Oper. Res., 96 (2000), pp. 167–189.

[17] N. Gröwe-Kuska, K. C. Kiwiel, M. P. Nowak, W. Römisch, and I. Wegner, *Power management in a hydro-thermal system under uncertainty by Lagrangian relaxation*, in Decision Making under Uncertainty: Energy and Power, C. Greengard and A. Ruszczyński, eds., IMA Vol. Math. Appl. 128, Springer, New York, 2002, pp. 39–70.

[18] N. Gröwe-Kuska, M. P. Nowak, and I. Wegner, *Modelling of uncertainty for the real-time management of power systems*, in Online Optimization of Large Scale Systems, M. Grötschel, S. O. Krumke, and J. Rambau, eds., Springer, Berlin, 2001, pp. 621–645.

[19] H. HEITSCH AND W. RÖMISCH, *Scenario reduction algorithms in stochastic programming*, Comput. Optim. Appl., 24 (2003), pp. 187–206.

[20] K. C. KIWIEL, *Proximity control in bundle methods for convex nondifferentiable minimization*, Math. Program., 46 (1990), pp. 105–122.

[21] K. C. KIWIEL, *User's Guide for NOA* 3.0: *A Fortran Package for Convex Nondifferentiable Optimization*, Polish Academy of Science, System Research Institute, Warsaw, 1994.

[22] B. KRASENBRINK, *Integrierte Jahresplanung von Elektrizitätserzeugung und -handel*, Aachener Beiträge zur Energieversorgung 81, H.-J. Haubrich, ed., Aachen, Germany, 2002.

[23] *Mathematica Time Series Pack: Reference and User's Guide*, Wolfram Research, Champaign, IL, 1995.

[24] M. P. NOWAK, *Stochastic Lagrangian Relaxation in Power Scheduling of a Hydro-Thermal System under Uncertainty*, Ph.D. thesis, Institute of Mathematics, Humboldt-University, Berlin, 2000; available online from http://dochost.rz.hu-berlin.de/dissertationen/.

[25] M. P. NOWAK AND W. RÖMISCH, *Stochastic Lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty*, Ann. Oper. Res., 100 (2000), pp. 251–272.

[26] R. NÜRNBERG AND W. RÖMISCH, *A two-stage planning model for power scheduling in a hydro-thermal system under uncertainty*, Optim. Engrg., 3 (2002), pp. 355–378.

[27] A. B. PHILPOTT, M. CRADDOCK, AND H. WATERER, *Hydro-electric unit commitment subject to uncertain demand*, Eur. J. Oper. Res., 125 (2000), pp. 410–424.

[28] S. T. RACHEV, *Probability Metrics and the Stability of Stochastic Models*, John Wiley, Chichester, UK, 1991.

[29] S. T. RACHEV AND W. RÖMISCH, *Quantitative stability in stochastic programming: The method of probability metrics*, Math. Oper. Res., 27 (2002), pp. 792–818.

[30] R.T. ROCKAFELLAR AND R. J.-B. WETS, *The optimal recourse problem in discrete time: L^1-multipliers for inequality constraints*, SIAM J. Control Optim., 16 (1978), pp. 16–36.

[31] W. RÖMISCH AND R. SCHULTZ, *Multistage stochastic integer programs: An introduction*, in Online Optimization of Large Scale Systems, M. Grötschel, S. O. Krumke, and J. Rambau, eds., Springer, Berlin, 2001, pp. 579–598.

[32] G. B. SHEBLE AND G. N. FAHD, *Unit commitment literature synopsis*, IEEE Trans. Power Syst., 9 (1994), pp. 128–135.

[33] *S-PLUS User's Guide Version* 4.5, Insightful Corporation, Seattle, 1998.

[34] B. STERN, *Kraftwerkseinsatz und Stromhandel unter Berücksichtigung von Planungsunsicherheiten*, Aachener Beiträge zur Energieversorgung 78, H.-J. Haubrich, ed., Aachen, Germany, 2001.

[35] S. TAKRITI, J. R. BIRGE, AND E. LONG, *A stochastic model for the unit commitment problem*, IEEE Trans. Power Syst., 11 (1996), pp. 1497–1508.

[36] S. TAKRITI AND J. R. BIRGE, *Lagrangian solution techniques and bounds for loosely coupled mixed-integer stochastic programs*, Oper. Res., 48 (2000), pp. 91–98.

[37] S. TAKRITI, B. KRASENBRINK, AND L. S.-Y. WU, *Incorporating fuel constraints and electricity spot prices into the stochastic unit commitment problem*, Oper. Res., 48 (2000), pp. 268–280.

[38] F. ZHUANG AND F. D. GALIANA, *Towards a more rigorous and practical unit commitment by Lagrangian relaxation*, IEEE Trans. Power Syst., 3 (1988), pp. 763–773.

This page intentionally left blank

Chapter 31

Valuation of Electricity Generation Capacity

*Shi-Jie Deng** *and Shmuel S. Oren*†

31.1 Introduction

The emerging restructuring of the power industry in the United States and abroad has resulted in change of ownership on a massive scale of electric generation assets through divestiture, merger, and acquisition of physical plants or long-term entitlement to the plants' output. Such ownership transfers are typically done through public auctions. Establishing the market value of generation assets has become an important problem for utility commissions and private organizations buying or selling such assets. The generation capacity of a typical power plant is measured in hundreds of megawatts (MW) and the selling price runs into hundreds of millions of dollars. Hence, even a few percentage points of improvement in the valuation accuracy can have substantial financial consequences. The public interest in such valuation stems from the fact that in most jurisdictions the proceeds from the sales offset ratepayer's liability for stranded costs of uneconomic investments that were made by regulated utilities. Private entities bidding for these assets are obviously interested in establishing their market value, which will guide their bids. Market-based valuation of generation asset is also important for investors in new generation capacity and for financial institutions that are financing such investments.

The uncertain energy prices that prevail in the new competitive electricity markets make the generation asset valuation problem challenging as compared to what it used to be under the old regulated regime, where electricity prices were set by regulators based on a fixed rate of return on investment. Under the rate of return regulation investment decisions in generation capacity were typically based on a discounted cash flow (DCF) method that

*School of Industrial and Systems Engineering, Georgia Institute of Technology, 765 Ferst Drive, Atlanta, GA 30332 (deng@isye.gatech.edu).

†Industrial Engineering and Operations Research, University of California at Berkeley, 4135 Etcheverry Hall, Berkeley, CA 94720 (oren@ieor.berkeley.edu).

was used to evaluate the expected future cash flow associated with the generation capacity under consideration. This paradigm is being changed by the restructuring of the electricity supply industries and the transition to market-based prices.

It has been recognized in the literature (e.g., [6]) that in the presence of price uncertainty the traditional DCF approach tends to undervalue assets by ignoring the optionality available to the asset owner. In a well-developed financial and physical market for electricity, the payoffs of an electric power plant can be approximated by a series of financial instruments on electricity, and thus financial methods can be applied to value a power plant via valuing the appropriate set of financial instruments. Such an approach is employed in [3] for the valuation of power generation assets. In particular, the authors construct a "spark spread option"–based valuation model for fossil-fuel power plants. They demonstrate that the option-based approach better explains the observed market valuation than does the DCF-based approach. In fact the DCF valuations underestimate, by nearly a factor of four, the sale prices of several power plants divested by a southern California utility. However, the pure option pricing approach tends to oversimplify the valuation problem by ignoring operational costs and constraints on a power plant, such as startup costs, ramp-up constraints, and operating-level-dependent heat rate.

While it is imperative to recognize the embedded optionality to properly value generation capacity, it is of equal importance to recognize that physical operating characteristics of a real asset often impose restrictions on exercising the embedded options. It is therefore important when applying financial option pricing methodology to examine the impact of operational constraints on the capacity valuation. We explicitly incorporate several operating characteristics of a power plant into its valuation and illustrate by way of a numerical example the significance of accounting for operating constraints and costs.

In section 31.2, we highlight several key operating characteristics of a fossil-fuel power generation asset and describe the valuation problem of the power plant in a competitive power market environment. We construct a discrete-time mean-reverting trinomial lattice for the electricity and the generating fuel prices in section 31.3. We then formulate a stochastic dynamic programming (SDP) model based on the lattice price processes for our valuation problem incorporating operational constraints and outline the solution procedure. In section 31.4, we present results from numerical experiments to illustrate the significance of the impact by each of the operating characteristics on the valuation at different operating efficiency levels. Finally, we conclude with observations and remarks.

31.2 Problem description

Real option valuation methods that model generation assets in terms of financial options are becoming increasingly popular. A key concept employed by these approaches is the *heat rate*, which measures the conversion rate from a generating fuel into electricity. In some rough sense, heat rate represents the number of units of the fuel needed for generating one unit of electricity. The economic value of a generation plant of given capacity and known heat rate can then be roughly represented in terms of a spark spread call option, which is an option that yields its holder the positive part of electricity price less the "strike" heat rate-adjusted fuel cost at the option's maturity time. The analogy between a spark spread option and a generation plant stems from the fact that an owner of a merchant power plant

(i.e., a power plant that can sell its output into at least one spot market) has the right, but not the obligation, to generate electricity by burning fuel at any point in time during the lifetime of the power plant. Suppose that the owner exercises such operational rights economically over time[1] and there are no operating constraints/lead times in running the power plant; then she would receive the spot price of electricity less the heat rate-adjusted generating fuel cost by selling/purchasing electricity/fuel, respectively, at spot market prices. Thus, the payoff obtainable by a rational merchant power plant owner at time t is the same as that of a spark spread call option with strike heat rate being set to the operating heat rate level of the power plant. The market value of a fossil-fuel power plant can then be obtained by summing up the values of the corresponding set of spark spread call options with an appropriate strike heat rate and the maturity time spanning the lifetime of the plant. Deng, Johnson, and Sogomonian [3] demonstrate that such a spark spread option–based valuation provides a much better approximation to empirically observed market valuations than does a DCF valuation.

However, the financial option valuation approach overlooks the differences between a physical asset and a financial asset. The optionality associated with operating a physical asset at different time epochs is often constrained by specific operating characteristics of the physical asset. These operational constraints may impose significant transaction costs on the exercise of the operational options either directly through setup costs or indirectly through operational lead times. Thus the financial option pricing formula tends to overestimate the option value of a real fossil-fuel generator.

The following implicit assumptions underlie a financial option–based valuation model: (a) a power plant can be instantly turned on or shut down; (b) there are no fixed operating costs but only variable production costs; and (c) the operating efficiency of a power plant is constant. Unfortunately, these assumptions are not very realistic. First, fixed costs are usually incurred whenever a power plant is turned on from the off state. For a steam generating unit, for instance, water in the boiler needs to be boiled before the unit can generate electricity, and the amount of fuel required to boil the water often depends on how long the unit has been shut down. That is, *startup costs* are involved in the process of turning a power generating unit on and the costs could be time-dependent. Second, the output level of a generator cannot be increased instantaneously to the full generation capacity upon turning on a power plant. A certain time period (e.g., the time for the water in the boiler to reach boiling temperature) is needed for a power plant to transit from the off state to the fully operational state. This time lag is often called the ramp-up time. Third, concerning the heat efficiency of a power plant, the rate at which a power plant converts the generating fuel into electricity varies with output levels. Specifically, a power plant is more efficient when it is operated at the rated capacity level than at a low output level.

Our objective is to explicitly incorporate the operating characteristics of a fossil-fuel power plant into the valuation model and illustrate the effects of these constraints on the valuation. In principle, one can formulate the operation of a power plant while incorporating operational characteristics in great detail as a full-fledged dynamic programming problem. Such an approach was employed by [9] for short-term generation asset valuation by solving a stochastic unit commitment problem with constraints on startup and shutdown costs,

[1]To "exercise a right economically" means that a rational power plant owner would exercise an operational right at time t only when the electricity price less generating fuel cost is positive at that time.

minimum run time, and maximum ramp rate. However, the computational complexity makes this approach prohibitively difficult to implement for a long time horizon. The difficulty arises from the fact that operational characteristics affect daily or even hourly decisions, whereas the lifetime of a plant over which its value is determined is of the order of 15 years or more.

Our focus in this paper is on long-term asset valuation. Our primary objective is to demonstrate a computationally feasible approximation method that will capture the essence of the operational constraints in a stochastic dynamic programming framework with realistic stochastic price models. A second objective is to assess the magnitude of the error introduced by ignoring the various operational characteristics of a generation asset and how this error varies with the heat rate of a plant. Our approach compromises on modeling the operational details by using a rather simplistic representation of some key operational aspects. Specifically, we represent the startup cost, ramp-up time, and output-dependent operating heat rate as described below and solve a stochastic dynamic program under the assumption of a discrete-time mean-reverting trinomial stochastic price model for electricity and fuel. A similar approach employing a more elaborate (and more precise) characterization of the underlying price processes is discussed in [5].

Startup cost We assume that there is a constant fixed cost c_{start} associated with the action of turning on a power plant from the off state because the water in the boiler of the generator must be heated before the generator can generate power. In general, the cost for starting up a generating unit depends on how long the unit has been turned off. The longer the unit is off, the more heat is dissipated from its boiler, and thus the higher the cost incurred when reheating the water. Nevertheless, we simplify this effect by assuming that the startup cost, c_{start}, is a constant.

Ramp-up time We approximate the ramp-up time (which introduces a lag in the exercise of the real on/off option) by assuming that whenever a power plant is turned on from the off state, there is a fixed delay time of length D before electricity can be generated. Once a power plant is turned on, it always takes a short period of time (i.e., ramp-up time) for its generating unit to reach certain operating output levels. Similar to the case of startup cost, the length of the ramp-up time also depends on how long the power plant has been off. To reflect this aspect to first order, we assume that there is a constant time lag between the time point at which a generating unit is turned on and the time point at which the generating unit reaches its full output capacity.

Output-dependent operating heat rate While it is known that the dependency between the power output level and the operating heat rate follows a nonlinear functional form (see [10]), we make a simplifying assumption by considering only two possible output levels for a plant: the rated capacity level $\overline{Q}$ per unit of time, called maximum output level, with an operating heat rate of $\overline{Hr}$; and the minimum capacity level $\underline{Q}$ ($\underline{Q} < \overline{Q}$) per unit of time, that is, the minimum output level allowable in order to keep a power plant operational, with a corresponding heat rate of $\underline{Hr}$. The constraint $0 < \overline{Hr} \leq \underline{Hr}$ reflects the fact that a fossil-fuel power plant is more efficient when operated at a high output level than at a low

output level. We also assume that the switching between the maximum capacity level and the minimum capacity level is instantaneous and costless.

31.3 A stochastic dynamic programming approach to capacity valuation

In valuing the power generation capacity, price models for electricity and the generating fuel are key ingredients. We employ a mean-reverting stochastic process to model the prices. Mean reversion has been demonstrated to be a common feature in almost all commodity prices, including energy prices (see [8]). In particular, we construct a discrete-time lattice price process that approximates the continuous-time mean-reverting electricity price model described in [3] and value a power plant based on the price lattice. The lattice (binomial tree) approach to option pricing was rigorously developed by [2]. Our approach is related to [1] and [7], which deal with pricing options on a multinomial lattice when there are multiple state variables.

Consider a finite time horizon of $[0, T]$ for the capacity valuation problem. To set up a discrete-time framework, we divide the interval $[0, T]$ into N subintervals, $[0, t_1]$, $(t_1, t_2]$, $\ldots$, $(t_{N-1}, t_N \equiv T]$ of equal length $\Delta t \equiv T/N$. Let the natural logarithm of the prices of electricity and the fuel be the state variables at time t, denoted by (X_t, Y_t). We assume that the state of the price processes changes value only at t_i $(i = 1, 2, \ldots, N)$ and the state vector (X_t, Y_t) takes on a finite set of values. With the understanding that (X_i, Y_i) denotes (X_{t_i}, Y_{t_i}) $(i = 0, 1, 2, \ldots, N)$, we rewrite the vector process $\{(X_t, Y_t) : t = t_0, t_1, \ldots, t_N\}$ as $\{(X_i, Y_i) : i = 0, 1, \ldots, N\}$. We start with the construction of the discrete-time price processes and then present the valuation model formulation.

31.3.1 A discrete-time mean-reverting price process

From here on, the generating fuel is specified to be natural gas. The following continuous-time mean-reversion models are employed in [3] for modeling the returns of electricity price s_t^e and natural gas price s_t^g:

$$\begin{aligned} dX_t &= \kappa_e(\theta_e - X_t)dt + \sigma_e dB_t^1, \\ dY_t &= \kappa_g(\theta_g - Y_t)dt + \rho\sigma_g dB_t^1 + \sqrt{1-\rho^2}\sigma_g dB_t^2, \end{aligned} \tag{31.1}$$

where $X_t = \ln S_t^e$ and $Y_t = \ln S_t^g$; θ_e and θ_g are the long-term means of X_t and Y_t, respectively; κ_e and κ_g are two positive mean-reverting coefficients indicating the rates at which the electricity price and the natural gas price revert to their respective long-term means; σ_e and σ_g are the instantaneous price volatilities of electricity and natural gas, respectively; ρ is the instantaneous correlation coefficient between the electricity and the natural gas price returns; and B_t^1 and B_t^2 are two independent standard Brownian motion processes.

Using the same set of parameters $(\kappa_i, \theta_i, \sigma_i, \rho)$ as those in (31.1), we construct two discrete-time mean-reverting price models for electricity and natural gas on a recombining trinomial lattice as follows. We choose a state space following [7]. Starting from each log-price state vector (X_t, Y_t) at time t $(t = 0, 1, 2, \ldots, N-1)$, there are three possible states (X_{t+1}^i, Y_{t+1}^i) $(i = 1, 2, 3)$ to reach at time $(t+1)$, as illustrated in Figure 31.1.

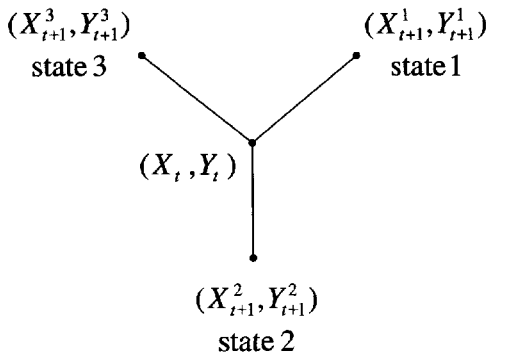


Figure 31.1. *Construction of a trinomial lattice.*

Let the price state movements on the trinomial lattice be

$$X_{n+1} = \begin{cases} X_n + \sigma_e\sqrt{\frac{3}{2}}\sqrt{\Delta t} & \text{(state 1)}, \\ X_n & \text{(state 2)}, \\ X_n - \sigma_e\sqrt{\frac{3}{2}}\sqrt{\Delta t} & \text{(state 3)}, \end{cases}$$

$$Y_{n+1} = \begin{cases} Y_n + \rho\sigma_g\sqrt{\frac{3}{2}}\sqrt{\Delta t} + \sigma_g\sqrt{1-\rho^2}\sqrt{\frac{1}{2}}\sqrt{\Delta t} & \text{(state 1)}, \\ Y_n - \sigma_g\sqrt{1-\rho^2}\frac{2}{\sqrt{2}}\sqrt{\Delta t} & \text{(state 2)}, \\ Y_n - \rho\sigma_g\sqrt{\frac{3}{2}}\sqrt{\Delta t} + \sigma_g\sqrt{1-\rho^2}\sqrt{\frac{1}{2}}\sqrt{\Delta t} & \text{(state 3)}, \end{cases} \tag{31.2}$$

where $\Delta t = \frac{T}{N}$, $n = 0, 1, \ldots, n-1$. Define a set of transition probabilities on the trinomial lattice so that the resulting price models for electricity and natural gas are mean-reverting. Let p_i denote the transition probability moving from a given state (X_n, Y_n) to state (X^i_{n+1}, Y^i_{n+1}) $(i = 1, 2, 3)$ and let $P \equiv (p_1, p_2, p_3)'$. If the conditions

$$\begin{cases} -\frac{1}{3} < \left[\frac{\kappa_e(\theta_e - X_n)}{\sqrt{6}\sigma_e} + \frac{\kappa_g(\theta_g - Y_n)}{2\sqrt{6}\sigma_g}\right]\sqrt{\Delta t} < \frac{2}{3}, \\ -\frac{2}{3} < \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t} < \frac{1}{3}, \\ -\frac{2}{3} < \left[\frac{\kappa_e(\theta_e - X_n)}{\sqrt{6}\sigma_e} - \frac{\kappa_g(\theta_g - Y_n)}{2\sqrt{6}\sigma_g}\right]\sqrt{\Delta t} < \frac{1}{3} \end{cases} \tag{31.3}$$

hold, then

$$P = \begin{cases} p_1 : & \frac{1}{3} + \left[\frac{\kappa_e(\theta_e - X_n)}{\sqrt{6}\sigma_e} + \frac{\kappa_g(\theta_g - Y_n)}{2\sqrt{6}\sigma_g}\right]\sqrt{\Delta t}, \\ p_2 : & \frac{1}{3} - \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t}, \\ p_3 : & \frac{1}{3} - \left[\frac{\kappa_e(\theta_e - X_n)}{\sqrt{6}\sigma_e} - \frac{\kappa_g(\theta_g - Y_n)}{2\sqrt{6}\sigma_g}\right]\sqrt{\Delta t}. \end{cases} \tag{31.4}$$

The conditions in (31.3) are for ensuring that the probabilities p_i $(i = 1, 2, 3)$ are between 0 and 1. If any of the conditions in (31.3) is not satisfied, then P is defined as

$$\text{If } \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t} \geq \frac{1}{3}, \quad P = \begin{cases} p_1: & \frac{1}{2}, \\ p_2: & 0, \\ p_3: & \frac{1}{2}. \end{cases}$$

$$\text{If } \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t} \leq -\frac{2}{3}, \quad P = \begin{cases} p_1: & 0, \\ p_2: & 1, \\ p_3: & 0. \end{cases} \tag{31.5}$$

$$\text{If } \begin{cases} -\frac{2}{3} < \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t} < \frac{1}{3}, \\ \left[\frac{\kappa_e(\theta_e - X_n)}{\sqrt{6}\sigma_e} + \frac{\kappa_g(\theta_g - Y_n)}{2\sqrt{6}\sigma_g}\right]\sqrt{\Delta t} \leq -\frac{1}{3}, \end{cases} \quad P = \begin{cases} p_1: & 0, \\ p_2: & \frac{1}{3} - \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t}, \\ p_3: & \frac{2}{3} + \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t}. \end{cases}$$

$$\text{If } \begin{cases} -\frac{2}{3} < \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t} < \frac{1}{3}, \\ \left[\frac{\kappa_e(\theta_e - X_n)}{\sqrt{6}\sigma_e} + \frac{\kappa_g(\theta_g - Y_n)}{2\sqrt{6}\sigma_g}\right]\sqrt{\Delta t} \geq \frac{2}{3}, \end{cases} \quad P = \begin{cases} p_1: & \frac{2}{3} + \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t}, \\ p_2: & \frac{1}{3} - \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t}, \\ p_3: & 0. \end{cases} \tag{31.6}$$

$$\text{If } \begin{cases} -\frac{1}{3} < \left[\frac{\kappa_e(\theta_e - X_n)}{\sqrt{6}\sigma_e} + \frac{\kappa_g(\theta_g - Y_n)}{2\sqrt{6}\sigma_g}\right]\sqrt{\Delta t} < \frac{2}{3}, \\ -\frac{2}{3} < \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t} < \frac{1}{3}, \\ \left[\frac{\kappa_e(\theta_e - X_n)}{\sqrt{6}\sigma_e} - \frac{\kappa_g(\theta_g - Y_n)}{2\sqrt{6}\sigma_g}\right]\sqrt{\Delta t} \leq -\frac{2}{3}, \end{cases} \quad P = \begin{cases} p_1: & 0, \\ p_2: & \frac{1}{3} - \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t}, \\ p_3: & \frac{2}{3} + \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t}. \end{cases}$$

$$\text{If } \begin{cases} -\frac{1}{3} < \left[\frac{\kappa_e(\theta_e - X_n)}{\sqrt{6}\sigma_e} + \frac{\kappa_g(\theta_g - Y_n)}{2\sqrt{6}\sigma_g}\right]\sqrt{\Delta t} < \frac{2}{3}, \\ -\frac{2}{3} < \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t} < \frac{1}{3}, \\ \left[\frac{\kappa_e(\theta_e - X_n)}{\sqrt{6}\sigma_e} - \frac{\kappa_g(\theta_g - Y_n)}{2\sqrt{6}\sigma_g}\right]\sqrt{\Delta t} \geq \frac{1}{3}, \end{cases} \quad P = \begin{cases} p_1: & \frac{2}{3} + \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t}, \\ p_2: & \frac{1}{3} - \frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}\sqrt{\Delta t}, \\ p_3: & 0. \end{cases} \tag{31.7}$$

The transition probabilities defined in (31.4)–(31.7) yield two mean-reverting price processes for electricity and natural gas. For instance, suppose that the conditions in (31.3) are satisfied in the current state (X_n, Y_n); thus (p_1, p_2, p_3) are given by (31.4). In (31.4), if the current Y_n is greater than θ_g, meaning that the current natural gas price is above its long-term mean, then $\frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}$ is a negative number. Therefore from (31.4) we see that p_2, which is the probability of moving toward a decreasing level of Y_{n+1}, is increased. On the other hand, if the current Y_n is a number less than θ_g, meaning that the current natural gas price is below its long-term mean, then $\frac{\kappa_g(\theta_g - Y_n)}{\sqrt{6}\sigma_g}$ becomes a positive number. Therefore from (31.4) we see that p_1 and p_3, which are the probabilities of moving toward increasing levels of Y_{n+1}, are increased. Similarly, when the current X_n is greater/less than its long-term mean θ_e, the probability of moving toward a decreasing/increasing level of X_{n+1} is increased. The transition probabilities defined in (31.5)–(31.7) also have such mean-reverting property with respect to the price state vector (X_n, Y_n).[2]

[2]This discrete-time model may not converge in distribution to the previously mentioned continuous-time mean-reversion model (see also [4]). The parameters need to be estimated directly from the discrete model structure. To guarantee the convergence to the continuous-time price model, a quadnomial rather than a trinomial lattice is needed, as shown in [5].

31.3.2 Valuation of a merchant power plant with operational constraints

Suppose that the log-prices of electricity and natural gas evolve according to the lattice process $\{(X_t, Y_t) : t = 0, 1, \ldots, N\}$ constructed in section 31.3.1. Moreover, we make the following assumptions regarding the operational characteristics of a natural gas–fired merchant electric power plant.

Assumption 1 The power plant of interest is subject only to the three operating characteristics described in section 31.2.

Assumption 2 When running the power plant, the operator takes one of the three possible actions at discrete time points. The three possible actions are to shut down the plant, to run the power plant at its minimum capacity level (turn on the plant first if it is currently off), and to run the plant at its maximum capacity level (turn on the plant first if it is currently off). We denote these three actions by **off**, **on_min**, and **on_max**, respectively.

The operator of the merchant power plant seeks to maximize the expected total profit of the power plant with respect to the random price vector (S_t^e, S_t^g) over the operating time horizon by making optimal decisions regarding whether to turn on or shut down the generating unit as well as how to operate the unit. Under the risk-neutral probabilities, the expected total profit of a power plant over its operating time horizon yields the value of the power plant during that time period.

Before getting into the formulation of the valuation problem, we introduce some additional notation. The operating time horizon T is divided into N periods. The power plant operator makes the operational decisions at the beginning of every m periods, i.e., the operator takes action only in periods $0, m, 2m, 3m, \ldots, km, \ldots$, etc.

n index for periods $(1, 2, \ldots, N)$;

X_n state variable indicating the logarithm of the electricity price in period n;

Y_n state variable indicating the logarithm of the natural gas price in period n;

w state variable indicating whether the power plant is currently on or off; $w = 0$ means that the power plant is currently off; $w = 1$ means that the plant is currently on;

β discount factor over one time period;

a_n action taken by the power plant operator in period n;

Φ the action space, and $\Phi \equiv \{\text{off, on\_min, on\_max}\}$.

c_{start}, $\underline{Q}$, $\underline{Hr}$, $\overline{Q}$, and $\overline{Hr}$ are defined in section 31.3.1. The ramp-up time is assumed to be one period's delay for simplicity. Let the value function $V_n(X_n, Y_n, w)$ be the expected total profit of the power plant over time periods $n, n+1, n+2, \ldots, N$, given the current price state vector (X_n, Y_n) and the current operating state w of the power plant. Then $V_n(X_n, Y_n, w)$ is obtained by solving the following three recursive equations with proper boundary conditions.

If $n \neq k \cdot m$, where $k = 0, 1, 2, \ldots$, then the operator takes no action in period n. The value of a power plant is equal to the discounted expected future value of the power plant.

$$V_n(X_n, Y_n, w) = \beta \cdot E_n[V_{n+1}(X_{n+1}, Y_{n+1}, w)], \tag{31.8}$$

where $E_n[\cdot]$ denotes the conditional expectation given information available in period n.

If $n = k \cdot m$, where $k = 0, 1, 2, \ldots$ and $w = 0$, then

$$V_n(X_n, Y_n, 0) = \max_{a_n \in \Phi} \begin{cases} \text{on\_max}: & -c_{\text{start}} + \beta \cdot E_n[V_{n+1}(X_{n+1}, Y_{n+1}, 1)], \\ \text{on\_min}: & -c_{\text{start}} + \beta \cdot E_n[V_{n+1}(X_{n+1}, Y_{n+1}, 1)], \\ \text{off}: & \beta \cdot E_n[V_{n+1}(X_{n+1}, Y_{n+1}, 0)]. \end{cases} \tag{31.9}$$

If $n = k \cdot m$, where $k = 0, 1, 2, \ldots$ and $w = 1$, then

$$V_n(X_n, Y_n, 1) = \max_{a_n \in \Phi} \begin{cases} \text{on\_max}: & \overline{Q} \cdot [\exp(X_n) - \overline{Hr} \cdot \exp(Y_n)] \\ & \quad + \beta \cdot E_n[V_{n+1}(X_{n+1}, Y_{n+1}, 1)], \\ \text{on\_min}: & \underline{Q} \cdot [\exp(X_n) - \underline{Hr} \cdot \exp(Y_n)] \\ & \quad + \beta \cdot E_n[V_{n+1}(X_{n+1}, Y_{n+1}, 1)], \\ \text{off}: & \beta \cdot E_t[V_{t+1}(X_{t+1}, Y_{t+1}, 0)]. \end{cases} \tag{31.10}$$

The boundary conditions are

$$V_{N+1}(x, y, w) \equiv 0 \quad \forall (x, y) \in R^2, w = 0, 1. \tag{31.11}$$

31.3.3 The solution of the SDP

With the trinomial price model constructed in section 31.3.1, the optimal policies of the SDP have a barrier control form. There exists a "no-action" band on the plane of the natural gas price (plotted on the horizontal axis) and the electricity price (plotted on the vertical axis). If the price vector (S_t^g, S_t^e) consisting of the market prices of natural gas and electricity is inside this band, then it is optimal for the plant operator to maintain the status quo. If the price vector is above the upper boundary of the band, then, depending on the state of the power plant, the operator should increase the output level of the plant from off to on or from minimum capacity to full capacity. If the price vector is below the lower boundary of the band, then it is optimal for the operator to reduce the output level of the power plant to off or minimum capacity depending on the state of the plant.

31.4 Numerical experiments

We have implemented this proposed methodology for valuing a natural gas–fired power plant to examine the impact of operational characteristics on the capacity valuation. We report some numerical results for a hypothetical 100 MW gas-fired power plant over a 720-day period. We assume that the gas power plant incurs a startup cost whenever turned on and that it takes 1 day to ramp up the power plant from the off state to a desired output state but there is no delay in increasing/decreasing output level once the power plant is on. For startup cost, we examine two possible values, \$6000/start and \$12,000/start. The maximum and the minimum capacity levels are assumed to be 100 MW and 50 MW, respectively. Moreover,

Table 31.1. *Parameters for mean-reversion price models.*

κ_1	3	κ_2	2.25
θ_1	3.15	θ_2	0.87
σ_1	0.75	σ_2	0.6
ρ	21.7		

Table 31.2. *Value of a natural gas–fired power plant with/without physical characteristics.*

Heat Rate ($HR_{\max}$ MMBtu/MWh)	8000	9500	12000	14000
Cap. Value (nophy. constr.)	5.211 mill.	3.236 mill.	1.381 mill.	0.679 mill.
Cap. Value (3 phy. constr./stup=\$6k)	5.153 mill.	3.176 mill.	1.335 mill.	0.648 mill.
Pctg. Val. Overstate. (ignoring 3 phy./stup=\$6k)	1.13%	1.88%	3.43%	4.83%
Cap. Value (2 phy. constr./stup=\$0)	5.207 mill.	3.230 mill.	1.378 mill.	0.677 mill.
Pctg. Val. Overstate. (ignoring stup only/stup=\$6k)	1.06%	1.70%	3.21%	4.51%
Cap. Value (3 phy. constr./stup=\$12k)	5.121 mill.	3.144 mill.	1.312 mill.	0.632 mill.
Pctg. Val. Overstate. (ignoring 3 phy./stup=\$12k)	1.75%	2.93%	5.28%	7.45%
Cap. Value (2 phy. constr./stup=\$0)	5.207 mill.	3.230 mill.	1.378 mill.	0.677 mill.
Pctg. Val. Overstate. (ignoring stuponly/stup=\$12k)	1.68%	2.74%	5.05%	7.12%

the ratio between the operating heat rates at the minimum and the maximum capacity levels of the power plant is assumed to be $1.38 : 1$. Under the mean-reversion price assumption for electricity and natural gas, the trinomial lattice is built with Δt being 1 day. The operator of the power plant makes operating decisions at all nodes of the lattice, i.e., $m = 1$. The initial prices of electricity and natural gas are assumed to be \$21.70 and \$3.16, respectively, which are sampled from the historical market prices.

The parameters used to construct the mean-reverting trinomial lattice are given in Table 31.1.

The value of the underlying power plant is calculated for each of the three cases: considering all three physical operating characteristics, ignoring the three operating characteristics, and ignoring the startup cost only. The numerical results are presented in Table 31.2.

We plot the value of the power plant accounting for all three operating characteristics for different heat rates in Figure 31.2. The x-axis represents different heat rates. The solid curves with crosses and circles plot the capacity value per year with startup cost being \$6000/startup and \$12,000/startup, respectively. The capacity value ignoring the three operating characteristics for different heat rates is plotted by the plain solid curve. All three curves are plotted against the capacity value axis on the left. The dashed curves

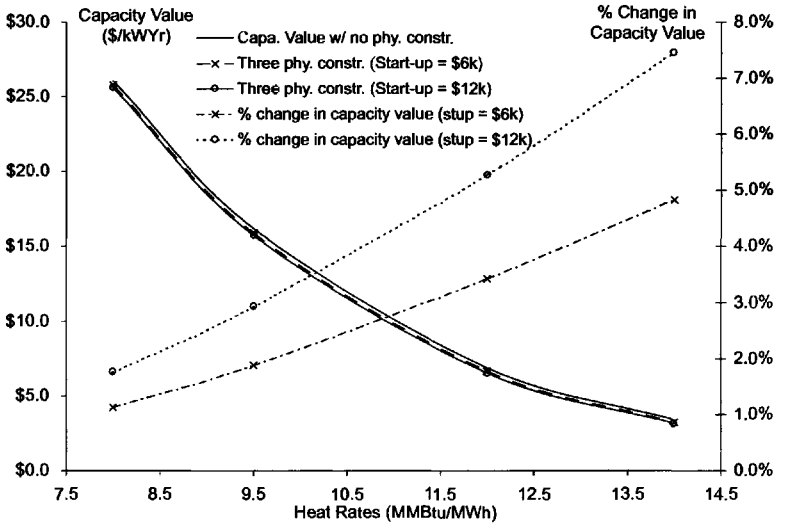


Figure 31.2. *Valuation of a power plant with/without physical characteristics.*

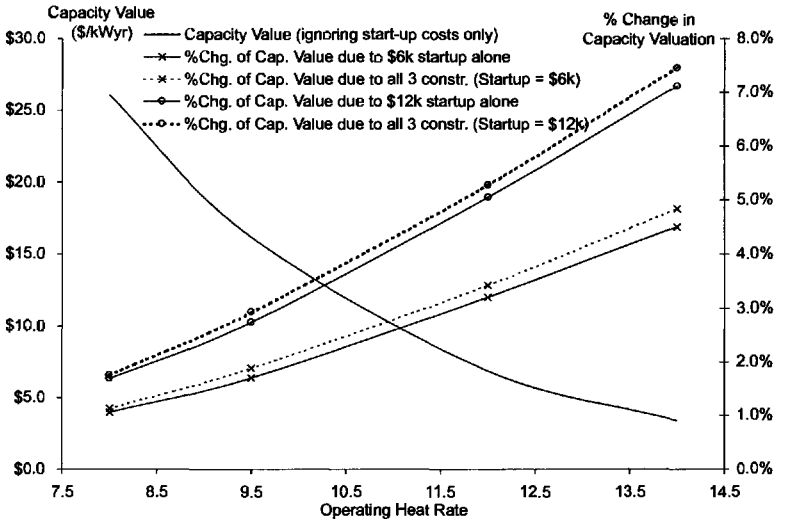


Figure 31.3. *Valuation of a power plant with/without the startup cost.*

with circles and crosses plot the percentage by which the capacity value is overstated due to ignoring the physical operating characteristics with the startup cost being $6000 and $12,000, respectively. The percentage for which the capacity value is overstated due to ignoring the operating characteristics ranges from 1.13% for the most efficient plant with a low startup cost to 7.45% for the least efficient plant with a high startup cost.

Figure 31.3 plots the values of the power plant with and without the startup cost only.

The impact of the startup cost on capacity valuation is very significant. Ignoring the startup cost while considering the other aspects accounts for more than 90% of the overstated capacity value of the underlying power plant.

31.5 Conclusion

We conclude from the numerical results that the operational characteristics affect the valuation of a merchant power plant to different extents depending on the operating efficiency of the power plant and the assumptions about the electricity and the generating fuel prices. In general, the impact of physical operating characteristics on power plant valuation can be very significant under the mean-reversion price models. Moreover, the more efficient a power plant is, the less affected its valuation is by the operational constraints and vice versa. The impact on capacity valuation ranges from 1.13% overvaluation for the most efficient plant with a low startup cost to 7.45% overvaluation for the least efficient plant with a high startup cost. Among the three operating characteristics of a power plant which we consider here, startup cost affects the capacity valuation the most. The reason is twofold. The first-order effect of the startup cost on capacity valuation is that it directly imposes a transaction cost on exercising the embedded spark spread options in a fossil-fuel power plant when the electricity price is greater than the fuel cost. The second-order effect of the startup cost is that it forces the power plant to keep operating at a loss or to forego a profit when the startup cost cannot be justified by the expected loss-saving or the expected profit that would result from turning the power plant off or on. In other words, the startup cost reduces the option value of a power plant. Our sensitivity analysis reveals that, under the mean-reversion price models, ignoring the startup cost alone can explain more than 90% of the overstated capacity value of a power plant (as compared to the overstated value when all three operational characteristics are ignored).

Acknowledgments

This article is mainly based on a previous work by the authors [4] presented at the EPRI Workshop on Applications of Planning under Uncertainty in the summer of 1999. The programming assistance of Shiming Deng is gratefully acknowledged.

Bibliography

[1] P. P. Boyle, *A lattice framework for option pricing with two state variables*, J. Financial Quant. Anal., 23 (1988), pp. 1–12.

[2] J. Cox, S. Ross, and M. Rubinstein, *Option pricing: A simplified approach*, J. Financial Econ., 7 (1979), pp. 229–263.

[3] S. J. Deng, B. Johnson, and A. Sogomonian, *Exotic electricity options and the valuation of electricity generation and transmission assets*, Decision Support Syst., 30 (2001), pp. 383–392.

[4] S. J. Deng and S. S. Oren, *Option-Based Valuation of a Power Plant Incorporating Physical Constraints*, Working Paper, University of California, Berkeley, CA, 1998.

[5] S. J. Deng and S. S. Oren, *Incorporating operational characteristics and startup costs in option-based valuation of power generation capacity*, Probab. Engrg. Inform. Sci., 17 (2003), pp. 155–181.

[6] A. K. Dixit and R. S. Pindyck, *Investment under Uncertainty*, Princeton University Press, Princeton, NJ, 1994.

[7] H. He, *Convergence from discrete- to continuous-time contingent claims prices*, Rev. Financial Stud., 3 (1990), pp. 523–546.

[8] E. S. Schwartz, *The stochastic behavior of commodity prices: Implications for valuation and hedging*, J. Finance, 52 (1997), pp. 923–973.

[9] C.-L. Tseng and G. Barz, *Short-term generation asset valuation: A real options approach*, Oper. Res., 50 (2002), pp. 297–310.

[10] A. Wood and B. Wollenberg, *Power Generation, Operation and Control*, John Wiley, New York, 1984.

This page intentionally left blank

Chapter 32

Stochastic Optimization Problems in Telecommunications

*Alexei A. Gaivoronski**

32.1 Introduction

Telecommunications have a long tradition of application of advanced mathematical modeling methods. Besides being a consumer of mathematical modeling, telecommunications provided a motivation for development of areas of applied mathematics. Important chapters of the theory of random processes have their roots in the work of telecommunication engineers. This mutual influence has been limited mainly to queueing theory and the theory of Markov processes, but now new decision problems arise which require the application of optimization methods. The recent trends in telecommunications have led to considerable increase in the level of uncertainty, which has become persistent and multifaceted. The decision support methodologies which provide adequate treatment of uncertainty are becoming particularly relevant for telecommunications. Stochastic optimization is the methodology of choice for optimal decision support under uncertainty; see [5, 13, 19]. This chapter provides a survey of applications of stochastic programming for solving design and decision problems in telecommunications.

Stochastic optimization is important for a large variety of such problems. We start by defining a classification which will serve as a roadmap for the exposition. This classification is made using the scale of the decision relative to the whole telecommunication environment which defines the nature of decision itself. Different types of uncertainty come into play at different levels. We distinguish three scale levels: technological, network, and enterprise. The technological level corresponds to the smallest scale and the enterprise level to the largest and the most aggregated scale.

The technological level deals with design of different elements of telecommunication

*Department of Industrial Economics and Technology Management, Norwegian University of Science and Technology, Alfred Getz vei 1, N-7491, Trondheim, Norway (alexei.gaivoronski@iot.ntnu.no).

networks, including switches, routers, and multiplexers. Uncertainty on this level is a salient feature of communication requests and flows in the network, and can arise due to equipment failures. The key decisions are the engineering decisions which define the design for blueprints of these elements. Such blueprints depend on a number of parameters which should be chosen from the point of view of performance and quality of service. Traditionally, performance evaluation of the elements of telecommunication networks was the domain of queueing theory [23]. To be successful the methods of this theory require a specific probabilistic description of the stochastic processes which govern the behavior of communication flows. Usually such a description is lacking for the new data services, and when it exists it does not satisfy the requirements of queueing theory. Stochastic optimization may help to obtain the performance estimates in the cases when more traditional methods are difficult to apply. See section 32.2 for one such example.

Network-level problems deal with design and planning of different kinds of networks which may differ by scale and by technology involved. These can be access networks, local area networks, or fixed or mobile networks. The decisions involve the placement of processing and link capacities provided by a given technology in a given geographic area with the aim to satisfy aggregated demand for telecommunication services from different user groups. Decisions are often of dynamic nature and include several time periods. The main uncertainty here is due to the demand for telecommunication services. Due to quantitative and qualitative explosion of such services, this kind of uncertainty has increased considerably during the last decade. There are important additional sources of uncertainty connected with possible network failures and future technology development. Stochastic programming methods provide an added value of identifying the robust network design which within reasonable bounds will accommodate future demand variations. This is particularly true for stochastic programming problems with recourse and multiperiod stochastic programming problems which provide intelligent means for mediation between different and often conflicting scenarios of the future. While the traditional design approach is centered around minimization of the network costs under technological and quality-of-service constraints, systematic application of stochastic programming techniques includes incorporation of modern tools from corporate finance like evaluation of real options. Comprehensive models which include pricing decisions and binary variables provide a motivation for further development of this methodology. Section 32.3 contains several examples of stochastic optimization models for design problems at the network level. For related examples, see [2, 7, 10, 14, 25, 26, 29].

Finally, the enterprise level is the highest level of aggregation and looks at the telecommunication enterprise as a member of a larger industrial environment, which includes other industrial actors and different consumer types. Decisions involve selection of the range of services which the enterprise will provide to the market, strategic investment decisions, and pricing policy. Market acceptance of services, innovation process, and actions of competition constitute the sources of uncertainty which are not present at the lower levels. The telecommunications and, more generally, the information industry differ in important ways from traditional industries due to the rapid pace of innovation. This leads to the absence of perfect markets and to fundamental nonstationarity, which makes it difficult to apply traditional microeconomic approaches based on equilibrium. Stochastic programming models enriched with selected notions of game theory can provide more adequate decision recommendations here. We outline one such model in section 32.4.

There is no rigid boundary between the three scale levels since decisions taken at each level influence decisions on other levels. Further exposition is organized along the lines of this classification. Each section contains examples of stochastic optimization models which illustrate the typical problems of each level. A summary concludes.

32.2 Technological level

The central problem is to find the design parameters of a piece of telecommunication equipment which will ensure a given level of performance for a specified class of traffic patterns. We describe this problem on a general level and then provide a specific example which deals with access design of a high speed data network.

In the vast majority of practically interesting cases the performance is measured by the functional

$$F(x, H) = \mathbb{E}_H f(x, \xi) = \int f(x, \xi) dH(\xi), \tag{32.1}$$

where x is the vector of design parameters, ξ denotes the values of a stochastic process defined on an appropriate probability space which describes the interaction of traffic with device, and H is the stationary distribution of this process. The function $f(x, \xi)$ describes the performance of the equipment for a given traffic value ξ and a given value of design parameters x. $\mathbb{E}_H$ is the expected value with respect to H. The values of the performance measure F should belong to the set Φ of admissible values which describes requirements for the grade of service. This set is usually defined by bounds ϕ^- and ϕ^+. The parameters x should satisfy

$$\phi^- \leq F(x, H) \leq \phi^+. \tag{32.2}$$

Usually equipment should work satisfactorily for a sufficiently wide range of admissible traffic patterns which are defined by a set of traffic parameters. Therefore instead of a single distribution H in (32.1) we have a whole set of distributions G, which is defined indirectly by traffic parameters of interest.

Design problem Find the values of design parameters x for which the values of the performance functional $F(x, H)$ belong to an admissible set Φ for all $H \in G$.

The final design is selected from among solutions of this problem by considering additional criteria which may have an economic or a manufacturing nature.

The function $f(x, \xi)$ from (32.1) is usually known and has a simple analytical structure. The major difficulty here is constituted by the distribution H. The reason is that the traffic is described in terms of characteristics of individual nonhomogeneous traffic sources. Even if description of a single source is relatively simple, the composite traffic consisting of a large number of such sources can be complex. Complexity increases due to nontrivial interaction of traffic with a device. Even the description of a single source can be difficult to obtain, especially in the case of new services which generate traffic with partly unknown properties. Therefore a characterization of the distribution H and set G can be extremely difficult.

Traditionally, performance analysis developed two complementary approaches for confronting this difficulty: analytical analysis and simulation. The analytical approach

requires that the traffic sources be described by a Markov chain. This is a serious limitation because the number of states in such a Markov chain can be high for realistic sources. After this the whole system consisting of the traffic and device is described by a Markov chain, and its stationary distribution H is computed numerically. This enables a computation of the performance functional from (32.1) by direct integration which reduces to summation. The main problem of this approach is that the resulting Markov chain is very often so large that the computation of its stationary distribution is far beyond the reach of modern computers. Approximations are necessary for a majority of the problems of interest, which may undermine the relevance of results. The analytic approach is often supplemented by a simulation approach. It consists of direct simulation of the interaction between the traffic and device and allows a considerably more realistic representation of the whole system. The problem with this approach is that the simulation times necessary for obtaining the estimates of stationary performance can be prohibitively long. This is especially true for the case when design requirements are expressed in terms of packet loss, which is a popular performance measure for modern data networks. Both these techniques have the common drawback that they are applicable for a given traffic pattern, while design should be valid for the whole range of traffic parameters.

Stochastic optimization can enhance both analytic and simulation approaches to performance analysis by addressing the problem of development of guaranteed estimates for performance of telecommunication systems. Such estimates involve computation of $F^+(x)$ and $F^-(x)$ such that

$$F^-(x) \le F(x, H) \le F^+(x) \tag{32.3}$$

for all $H \in G$. When performance requirements are described by (32.2), one can select x from

$$\phi^- \le F^-(x), \qquad F^+(x) \le \phi^+,$$

which will guarantee satisfaction of performance requirements for all traffic patterns of interest. The bounds $F^+(x)$ and $F^-(x)$ can be obtained from the solution of

$$F^-(x) = \inf_{H \in G} F(x, H), \qquad F^+(x) = \sup_{H \in G} F(x, H). \tag{32.4}$$

These problems can be classified as belonging to a special class of stochastic optimization problems, namely, optimization problems in the space of probability measures; see [11, 12, 16, 20, 21]. One may object that solving such problems should be even more difficult than computing the value of $F(x, H)$ for a given H, a difficult problem by itself as argued above. In many cases this is not true. First, the function $F(x, H)$ often can be approximated by simpler functions $F^+(x, H)$ and $F^-(x, H)$ satisfying

$$F(x, H) \le F^+(x, H), \qquad F^-(x, H) \le F(x, H)$$

for all $H \in G$. These functions can be used in (32.4) instead of $F(x, H)$, which will make these problems simpler. Even more important, the measures which solve the problems (32.4) often have a very special structure which can be obtained from analysis of function $F(x, H)$ and set G without solving the problem itself [12, 28]. Numerical methods have been developed which exploit this structure [16] and simplify the solution of (32.4).

32.2.1 Access engineering of a broadband multiservice network

We illustrate this general approach using a specific example of access engineering of a broadband multiservice network taken from [8]. We consider a high-speed data network with data packets of fixed length, for example, a network based on asynchronous transfer mode (ATM) architecture; see [9]. The central part of an access network consists of a server (multiplexer) with one output link and L input links. Data packets arrive from input links and are sent by the server to the network, perhaps staying some time in the buffer of limited capacity x_0. If some packet finds the buffer full on arrival, it is discarded. Since all the packets are of the same length, so are the service times. Therefore it is natural to consider the system as operating in discrete time, with the time interval being equal to the service time of one packet. Denoting by $l(t)$ the buffer contents at time t, by $\xi_i(t)$ the number of packets which arrive from the source i at time t, and by $\xi(t)$ the total number of packets arrived at time t, we have the following relation which describes the dynamics of the buffer contents:

$$l(t+1) = \max\{0, \min\{x_0, l(t) + \xi(t) - 1\}\}, \quad \xi(t) = \sum_{i=1}^{L} \xi_i(t),$$

where $\xi_i(t)$ is 0 or 1. Each source i at the input of a server generates a packet arrival process with distribution H_i. This distribution defines the probability that a packet arrives at time t conditioned on the history of packet arrivals and is described by a vector of parameters $a : H_i = H_i(a)$. Many distributions may correspond to a given value of the vector a. Each source belongs to one of N classes where each class corresponds to a given service. In terms of arrival distribution H_i each class is characterized by a set A_j in the space of parameters such that if the source i belongs to the traffic class j, it means that $H_i \in G_j$, where

$$G_j = \{H(a) | a \in A_j\}.$$

Examples of traffic classes and corresponding parameter sets will be given later. Denote by x_j the number of sources which belong to class j; then $\sum_{j=1}^{N} x_j = L$. I_j is a subset of $\{1, \ldots, L\}$, which indexes the sources belonging to the traffic class G_j. The maximal admissible number of sources of each class x_j, $j = 1 : N$ together with the buffer length x_0 constitute the vector x of design parameters which should be chosen so that the access system satisfies given performance requirements expressed in terms of the admissible upper bound ϕ^+ on the packet loss probability $F(x, H)$:

$$F(x, H) = \frac{\mathbb{E}\max\{0, \xi(t) - l(t) - 1\}}{\mathbb{E}\xi(t)} \leq \phi^+, \tag{32.5}$$

where $H = H(z; x) = \mathbb{P}\{\xi(t) \leq z\}$ denotes the stationary distribution of $\xi(t)$. Usually this bound is chosen between 10^{-10} and 10^{-9}.

This and similar systems constitute an important part of telecommunication networks and considerable effort was dedicated to its study; see, for example, [22]. The Markov chain analysis of this system starts by assuming that the packet arrivals from a single source are independent. This is a very serious assumption because the packets in ATM networks are generated by splitting larger amounts of data into small packets of the standard

size. Therefore the traffic from a single source consists of bursts which correspond to each communication request. Even then, the resulting Markov chain contains at least $\prod_{i=0}^{N}(x_i+1)$ states, which can be a very large number. The number of states explodes further if more realistic assumptions about the traffic are taken. A simulation approach also encounters difficulties because estimation of probabilities of the order of 10^{-10} requires long simulation times.

This problem can be treated by optimization over probability measures. We start by deriving a bound on the packet loss probability from (32.5).

Proposition 32.1 (see [8]). *Suppose that $\xi_i(t)$ are the stationary ergodic stochastic processes for all i and that the length of the buffer x_0 is not smaller than the total number of sources L. Then*

$$F(x, H) \le F^+(x, H) = \frac{\int \max\{0, z-1\} dH(z; x)}{\int z dH(z; x)} = \frac{\mathbb{E}_H \max\{0, \zeta - 1\}}{\mathbb{E}_H \zeta}, \tag{32.6}$$

where ζ is a random variable distributed according to H and $\mathbb{E}_H$ is expectation with respect to H.

The arrival processes from each source usually are assumed to be independent. Then the upper bound $F^+(x)$ on the cell loss probability is a solution of

$$\begin{aligned} F^+(x) &= \max_{H_i \in G_j \ \forall i \in I_j} F^+(x, H) \\ &= \max_{H_i \in G_j \ \forall i \in I_j} \frac{1}{B(x, H)} \int \cdots \int \max\left\{0, \sum_{i=1}^{L} z_i - 1\right\} \prod_{i=1}^{L} dH_i(z_i), \end{aligned} \tag{32.7}$$

where

$$B(x, H) = \sum_{i=1}^{L} \int z_i dH_i(z_i). \tag{32.8}$$

To advance further it is necessary to specify the sets Gj which define the traffic classes. One common way of doing so is to put bounds on the moments of the distributions belonging to this set together with the bounds on the support of these distributions. In this case

$$G_j = G_j(a_j) = \left\{ H | \int_0^{a_{0j}} dH(z) = 1, \int_0^{a_{0j}} \psi_{rj}(z) dH(z) \le a_{rj}, r = 1 : R \right\}, \tag{32.9}$$

where $a_j = (a_{0j}, a_{1j}, \ldots, a_{Rj})$ and $\psi_{rj}(z)$ are known functions. From the point of view of access design, this definition has an important meaning. The bound on support a_{0j} is the peak bandwidth of the source from the class j measured in fractions of the output bandwidth of the server. Let $\psi_{1j}(z) = z$; then a_{1j} will be the average bandwidth of a source from the class j. The properties of (32.7) with the sets G_j defined by (32.9) are well understood. Its solution has a special structure which was exploited for development of numerical methods in [12, 16]. Sometimes it is possible to obtain an explicit solution as in the important case when $G_j(a_j)$ is defined by the values of peak and average bandwidth only.

Theorem 32.2 (see [8]). *Suppose that the traffic classes G_j are defined by the peak bandwidth a_{0j} and the average bandwidth a_{1j}. Then among the sources which yield the largest packet loss probability always exist those with cell arrival distribution concentrated in two points $(0, a_{0j})$ with weights $(1 - p_j, p_j)$, $p_j = a_{1j}/a_{0j}$. The tight upper bound for the cell loss probability is*

$$F^+(x) = \frac{1}{\sum_{j=1}^N a_{1j}x_j} \sum_{\substack{0 \le k_j \le x_j \\ 1 \le j \le N}} \max\left\{0, \sum_{j=1}^N a_{0j}k_j - 1\right\} \prod_{j=1}^N \frac{x_j!}{k_j!(x_j - k_j)!} p_j^{k_j}(1 - p_j)^{x_j - k_j}. \tag{32.10}$$

A design decision can be taken by finding feasible solutions of

$$F^+(x) \le \phi^+,$$

where ϕ^+ is in Theorem 32.2. A design obtained by stochastic optimization has the following advantage compared to designs obtained by Markov chain modeling or simulations. It will ensure the required quality of service for all sources with specified average and peak bandwidth. In contrast, a design obtained through Markov chain modeling will be valid only for a much narrower class of sources which in addition generate packets with independently distributed arrival times. Simulations can ensure required quality of service only for a finite set of sources which were selected for simulation experiments.

Stochastic optimization can be applied to other design problems. Often it is important to exploit carefully the special structure of each particular case to obtain approximations of performance measures similar to (32.6). It is also possible to address the problem of this approximation from a more general point of view by developing approximations of complex random processes by simpler ones. For the case of Markov chains such approximations, which yield guaranteed bounds for performance measures, were developed in [6].

32.3 Network level

The objective of the network level is to develop a design of the telecommunication network with a given capability to provide a set of services to a population of end users. Results of technological design are used as the inputs to the network design. This design should serve different and often conflicting purposes, e.g., satisfaction of demand, maintenance of a given service quality, or cost effectiveness. Important decisions to make at this level include placement and dimensioning of processing nodes and transmission links. These decisions are affected by service pricing because it affects the quantity of demand to be satisfied. There is considerable literature dedicated to the optimal design of networks in a deterministic case [1, 4, 15, 24].

In the age of big state monopolies, largely immutable services, and a highly predictable environment the prevailing paradigm was the minimization of network costs under constraints on quality of service and demand satisfaction. Although this paradigm remains important, it is clearly insufficient for the highly mutable and uncertain environment of today. A robust network which can accommodate within reasonable bounds future market changes is more valuable than a possibly cheaper network designed for current conditions

and perhaps for one specific future scenario. For this reason profit, service pricing, and evaluation of investment opportunities under uncertainty become increasingly important in the network design models. This is where stochastic programming models have a competitive edge compared to deterministic models because the capability to mediate between scenarios of an uncertain future is explicitly embedded into them. This capability is very important because uncertainty of different kinds is one of the defining features of modern telecommunications. On the network level the main source of uncertainty is unpredictable user response to introduction of new services, which results in highly uncertain demand variations. To this one can add uncertainty due to technological innovation and uncertainty related to possible failures.

Another important feature of network design is the dynamic character of decisions. Network development projects have a time dimension, and an important decision is how to distribute the investment over time. New information about the market will become available, and the possibility to react to this information should be included in the decision models. Stochastic programming with recourse provides adequate tools for doing so. It also facilitates the incorporation of adaptation policies specifically designed to allow the network to react flexibly to the changing environment. Such policies are essential for robust network design. Another important issue is the correct evaluation of flexibilities present in the network investment projects. Examples of such flexibilities or real options are the option to expand, the option to upgrade technology, and the option to alter usage. Consideration of these options can drastically change the overall evaluation of the network expansion project. For example, a project which is unprofitable at first glance can reveal hidden profit opportunities.

Further exposition is organized in the form of examples which illustrate the general ideas. Section 32.3.1 presents a series of decision models for planning of an Internet-based information service starting from a simple traditional deterministic cost minimization model to a two-period stochastic programming model for profit maximization which can be used for evaluation of real options embedded in this project. The problem of design of access network described in section 32.2 is considered in section 32.3.2 on the level of network design. This is an interesting example which shows interplay between both levels. Sections 32.3.3 and 32.3.4 show examples of how technology influences network design models in the case of backbone networks. Network design which takes into account possible failures is considered in section 32.3.5. Finally, section 32.3.6 is dedicated to design of mobile networks in the situation of leveling out of demand.

32.3.1 Planning of an Internet-based information service

We consider the problem of deployment of an Internet-based information service on some territory, which can be a country or a region. We assume that the network itself exists already and the decision consists of deployment of servers at the nodes of this network and assignment of demand generated in different geographical locations to these servers. The service provider on behalf of which the problem is solved can be the network owner but can also be a virtual service provider which does not possess its own network and leases a network from some network owner. Decisions to consider include phased introduction of service, which can take the shape of Phase 1 deployment followed by Phase 2 deployment

contingent on the market reaction. In addition, decisions include pricing of service.

Among various aspects of the problem one should consider geographical dimension; uncertainty of demand and costs; cost structure, which includes fixed and variable costs; competition and substitution between services; and relations between different market actors, e.g., network providers and service providers. The decision to go ahead with the project depends on the project profitability, which in its turn depends on various options embedded in it, e.g., the option to expand, to abandon, or to upgrade technology. It is advisable to start the model development from the simple case which includes only some of the relevant features and to expand the model stepwise. We present here three such model development steps.

Step 1: Single-period deterministic cost minimization model

We start by considering only one decision period and full knowledge about demand and other uncertainties. Although these assumptions are highly unrealistic, the resulting model sets the stage for more realistic models. In this setting we assume that the deployment program has to satisfy the known demand fully. Since the service price is fixed, the revenue becomes fixed. For this reason the only way one can influence the profit is by minimizing the costs.

Notation

$i = 1 : n$ index for regions which constitute a territory. User population exists in each region which generates demand;

$j = 1 : m$ index for possible server locations;

y_j binary variable which takes the value 1 if decision to place a server at location j is taken, and 0 otherwise;

x_{ij} amount of demand from region i served by server placed in location j;

f_j fixed cost for setting up a server in location j;

c_{ij} cost for serving of one unit of demand from region i by server at location j;

d_i demand generated at region i;

g_j capacity of server placed at location j.

Model Find the server deployment program $y = (y_1, \ldots, y_m)$ and assignment of user groups to servers $x = \{x_{ij}\}, i = 1 : n, j = 1 : m$ as a solution of

$$\min_{x,y} \sum_{j=1}^{m} f_j y_j + \sum_{j=1}^{m} \sum_{i=1}^{n} c_{ij} x_{ij}, \tag{32.11}$$

$$\sum_{j=1}^{m} x_{ij} \geq d_i, \quad i = 1 : n, \tag{32.12}$$

$$\sum_{i=1}^{n} x_{ij} \leq g_j y_j, \quad j = 1 : m, \tag{32.13}$$

where $y_j \in \{0, 1\}$ and $x_{ij} \geq 0$. Here the first term in (32.11) represents the fixed cost of deployment of servers, while the second term represents the variable costs for serving demand. The constraints (32.12) are imposed to obtain full demand satisfaction, while constraints (32.13) are the capacity constraints. This is a well known facility location model, and it will serve as a starting point for development of a stochastic programming model with different scenarios of the future demand and a larger number of deployment phases.

Step 2: Two period stochastic cost minimization model

Two deployment phases are considered: the present Phase 1 with known demand and the future Phase 2 with uncertain demand which is described by a finite number of scenarios. Each scenario is described by demand values in different regions during Phase 2 and the probability of this scenario. The Phase 2 decisions include additional deployment of servers and reassignment of demand to servers in response to the demand which becomes known. The model follows the framework of stochastic programming with recourse.

Additional notation

$r = 1 : R$ index for demand scenarios;

d_i^r demand generated by region i under scenario r;

p^r probability of scenario r;

z_j^r binary variable which takes the value 1 if under scenario r the decision to place a server at location j is taken, and 0 otherwise;

x_{ij}^r amount of demand from region i served by a server placed in location j under scenario r;

α coefficient for discounting of the Phase 2 costs to the present.

Each scenario is characterized by a pair (d^r, p^r), where $d^r = (d_1^r, \ldots, d_n^r)$.

Model Find the Phase 1 server deployment program $y = (y_1, \ldots, y_m)$ and assignment of user groups to servers $x = \{x_{ij}\}$, $i = 1 : n$, $j = 1 : m$, as a solution of

$$\min_{x,y} \sum_{j=1}^{m} f_j y_j + \sum_{j=1}^{m} \sum_{i=1}^{n} c_{ij} x_{ij} + \alpha \sum_{r=1}^{R} p^r Q(r, y) \tag{32.14}$$

subject to (32.12)–(32.13). The third term in (32.14) represents discounted costs of the Phase 2 deployment averaged over scenarios. The cost associated with scenario r is $Q(r, y)$

and it depends on the Phase 1 deployment decision y. These costs are obtained from the solution of the recourse problem for each scenario r

$$Q(r, y) = \min_{x^r, z^r} \sum_{j=1}^{m} f_j z_j^r + \sum_{j=1}^{m} \sum_{i=1}^{n} c_{ij} x_{ij}^r, \tag{32.15}$$

$$\sum_{j=1}^{m} x_{ij}^r \geq d_i^r, \quad i = 1 : n, \tag{32.16}$$

$$\sum_{i=1}^{n} x_{ij}^r \leq g_j(y_j + z_j^r), \quad j = 1 : m, \tag{32.17}$$

which is similar to (32.11)–(32.13) and chooses the Phase 2 deployment $z^r = (z_1^r, \ldots, z_m^r)$ and new assignment of user groups to servers $x^r = \{x_{ij}^r\}, i = 1 : n, j = 1 : m$ from minimization of fixed deployment costs and variable service costs for a given scenario r.

This can be a numerically challenging problem because it contains binary variables. Still, modern optimization technology permits solving it for nontrivial and practically important cases. For example, we solved its deterministic equivalent to optimality using the MPL modeling system powered by CPLEX and XPRESS solvers with $R = 5$, $n = m = 20$ in approximately 8 minutes on a 1133 MHz Pentium III laptop. The deterministic equivalent for this case has 120 binary and 2400 continuous variables with 240 constraints. The time has grown to 1 hour for $R = 6$, $n = m = 25$ with the deterministic equivalent having 175 binary and 4375 continuous variables and 350 constraints. Utilization of decomposition is essential for solving the problems of larger dimensions.

Step 3: Two-period stochastic profit maximization model with pricing

In a competitive deregulated environment, profit maximization is a more appropriate objective than the minimization of network costs. It becomes fundamentally different from plain cost minimization when the pricing decisions are considered simultaneously with deployment decisions. The models become more complicated because pricing affects demand and this dependence introduces nonlinearities. Still, meaningful analysis is feasible also in this case. We start by defining the linear demand model extending the scenario framework explained earlier.

Additional notation

h_0 reference price for service during Phase 1;

d_{i0} reference demand at region i during Phase 1 which corresponds to reference price h_0;

w_i demand elasticity at region i during Phase 1;

h the price increment relative to the reference price during Phase 1;

h_0^r reference price for service during Phase 2 under scenario r;

d_{i0}^r reference demand at region i during Phase 2 which corresponds to reference price h_0^r under scenario r;

w_i^r demand elasticity at region i during Phase 2 under scenario r;

h^r the price increment relative to the reference price during Phase 2 under scenario r.

Demand model This is the crucial piece of the profit model. Consider the Phase 1 deployment. The service price equals $h_0 + h$, and the price decision consists of selecting the price increment h, which may be positive or negative. Assume that the demand d_i in region i during this phase depends on the price of the service according to some function $d_i = f_i(h_0 + h)$, and in the vicinity of the point h_0 this dependence can be linearized via

$$d_i = d_{i0} - w_i h. \tag{32.18}$$

Similar relations describe the demand behavior during Phase 2 for each of the scenarios $r = 1 : R$. Each scenario is defined in this case by a tuple $(d_{i0}^r, h_0^r, w_i^r, p^r)$ which defines the dependence of demand on price according to relation (32.18) for a given scenario r.

Decision model Find the Phase 1 increment for the service price h, server deployment program $y = (y_1, \ldots, y_m)$, and assignment of user groups to servers $x = \{x_{ij}\}$, $i = 1 : n$, $j = 1 : m$ as a solution of

$$\max_{h,x,y} W(h) - C(y, x) + \alpha \sum_{r=1}^{R} p^r P(r, y) \tag{32.19}$$

subject to

$$w_i h + \sum_{j=1}^{m} x_{ij} \geq d_{i0}, \ i = 1 : n, \tag{32.20}$$

and constraint (32.13). Here $W(h)$ is the revenue during Phase 1,

$$W(h) = \sum_{i=1}^{n} (h + h_i)(d_0 - w_i h), \tag{32.21}$$

and $C(y, x)$ are the costs during Phase 1 defined according to (32.11). The third term in (32.19) represents the profits during Phase 2 averaged over scenarios and discounted to the present where $P(r, y)$ is the profit during Phase 2 under scenario r. It is taken as the optimal value of the following recourse problem:

$$P(r, y) = \max_{h^r, x^r, z^r} W(r, h^r) - C(r, z^r, x^r), \tag{32.22}$$

$$w_i^r h^r + \sum_{j=1}^{m} x_{ij}^r \geq d_{i0}^r, \ i = 1 : n, \tag{32.23}$$

subject to additional constraints (32.17). Here $W(r, h^r)$ is the revenue during Phase 2 under scenario r obtained similarly to (32.21), and $C(r, z^r, x^r)$ are the costs during Phase 2 under scenario r taken from (32.15).

There is one important feature of this model which was not present in the models (32.11)–(32.13) and (32.14)–(32.17). While (32.14)–(32.17) can be transformed to a mixed integer linear program by considering the deterministic equivalent, no such transformation is possible for the model (32.19)–(32.23). This is because the revenues $W(r, h)$ and $W(r, h^r)$ depend nonlinearly on the decision variables h and h^r. Even in the simplest case of the linear demand model (32.18) this dependence is quadratic. Therefore specialized numerical techniques should be employed in this case, with decomposition approaches being the most promising.

Evaluation of investment opportunities, real options. One of the most important utilizations of model (32.19)–(32.23) is the evaluation of profitability of the investment project which consists of the deployment of the new service. Recent developments in corporate finance showed the importance of evaluation of real options for correct evaluation of industrial projects [27]. While for more traditional industries direct evaluation techniques can be similar to evaluation of financial options, for innovative industries with unique projects such approaches are difficult to apply. Stochastic programming models can represent a valid alternative for real option evaluation. Let us utilize model (32.19)–(32.23) for this purpose. In particular, let us evaluate options to expand, to upgrade technology, to abandon or to convert a part of the infrastructure.

Option to expand (wait and see option). This option is already embedded in the model (32.19)–(32.23), which contains the possibility to add additional servers during Phase 2 contingent on market reaction. The value of this option can be computed as follows. Denote by P^* the optimal value of the model (32.19)–(32.23). This is the value of the project with an option to expand. The value $\hat{P}$ of the same project without the option to expand is obtained by solving the same model with binary variables z^r fixed to zero for all scenarios. Clearly, $\hat{P} \leq P^*$. The value of the option is the difference $P^* - \hat{P}$.

Option to upgrade technology. This is a valuable option because it can dramatically change the project evaluation. The best-known example is the GSM mobile network, whose development began before the technology to make very small mobile phones was available. To evaluate this option it is necessary to have a closer look at the ways technology development can affect various components of the model (32.19)–(32.23). For example, technology development can lead to decreased fixed costs for server installation and/or increase in the possible server capacities during Phase 2. In this case it is necessary to introduce these features into the definition of scenarios; f_j^r is the fixed cost for setting up a server in location j under scenario r, and g_i^r is the capacity of a server placed at location i under scenario r.

The model changes as follows. Part (32.19)–(32.20) remains the same because it describes Phase 1 to be implemented with known technology. Part (32.22)–(32.23) has a modified capacity constraint which substitutes (32.17)

$$\sum_{i=1}^{n} x_{ij}^r \leq g_j y_j + g_j^r z_j^r, \quad j = 1 : m. \tag{32.24}$$

The cost term $C(r, z^r, x^r)$ from (32.22) is

$$C(r, z^r, x^r) = \min_{x^r, z^r} \sum_{j=1}^{m} f_j^r z_j^r + \sum_{j=1}^{m} \sum_{i=1}^{n} c_{ij} x_{ij}^r. \tag{32.25}$$

The model (32.19)–(32.23) is solved with modification (32.24)–(32.25), which will give the value P^{**} of the project with an option to upgrade technology. This value is compared with the value of the project P^* without such an option, and the difference $P^{**} - P^*$ will give the option value.

Option to abandon. This is a valuable option when the market reaction is uncertain. If demand does not catch up, it is reasonable to cut maintenance costs in the regions where demand is weak and possibly recover part of the fixed costs by selling or leasing the server infrastructure. b_j^r is the maintenance costs for a server at location j during Phase 2 under scenario r, β_j^r is the fraction of fixed costs which can be recovered by abandonment of a server at location j under scenario r, and u_j^r is the binary variable which equals 1 if a server at location j is abandoned during scenario r.

The model changes as follows. Again, part (32.19)–(32.20) which refers to Phase 1 remains the same. Part (32.22)–(32.23) has a modified capacity constraint which substitutes for (32.17)

$$\sum_{i=1}^{n} x_{ij}^r \le g_j(y_j + z_j^r - u_j^r), \quad j = 1:m, \quad r = 1:R, \tag{32.26}$$

and additional abandonment constraints

$$u_j^r \le y_j, \quad j = 1:m, \quad r = 1:R. \tag{32.27}$$

The revenue term $W(r, h^r)$ and the cost term $C(r, z^r, x^r)$ from (32.19) are

$$W(r, h^r) = \sum_{i=1}^{n} (h_0^r + h^r)(d_{i0}^r - w_i^r h^r) + \sum_{j=1}^{m} \beta_j^r f_j u_j^r, \tag{32.28}$$

$$C(r, z^r, x^r) = \sum_{j=1}^{m} \left(f_j z_j^r + b_j^r (y_j + z_j^r - u_j^r) + \sum_{i=1}^{n} c_{ij} x_{ij}^r \right). \tag{32.29}$$

The model (32.19)–(32.23) is solved with the modification (32.26)–(32.29), which will yield the value P^{++} of the project with an option to abandon infrastructure. This value is compared with the value of the project P^+ obtained by solving the same model with variables u_j^r set to zero, which corresponds to evaluation without the option to abandon. The difference $P^{++} - P^+$ is the option value.

We now provide an example of such an option evaluation in Figure 32.1. This figure shows the dependence of the project value on the service price $h_0 + h$ for the case when the Phase 2 service prices were fixed to the Phase 1 prices, i.e., $h_0^r \equiv h_0, h^r \equiv h$. Three alternatives are shown in this figure. The first alternative is depicted by a thin line and describes the dependence of project value on price for the case when no option to expand and no option to upgrade technology are considered during Phase 2. The second alternative allows an option to expand but not an option to upgrade technology and is depicted by a dotted line. The third alternative shown with a thick line allows both options during Phase 2.

First, one notices the jumps on the curves, which are due to the discrete character of the decisions. The objective in all three cases is full demand satisfaction. A small increase

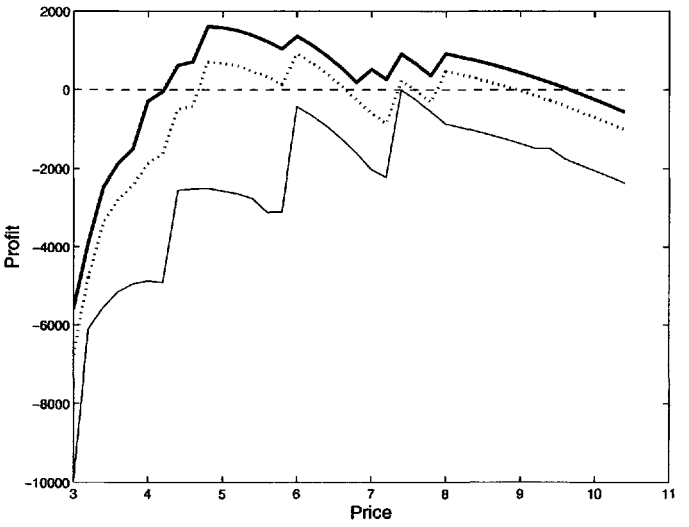


Figure 32.1. *Evaluation of real options in the case of service introduction.*

in price leads to a small decrease in demand, which can make a given server redundant with a corresponding stepwise decrease in fixed costs. Another observation confirms the added value of flexibility which options provide. The value of the project without option is not positive even for the best choice of service price. The project becomes profitable when the option to expand is allowed. There are two regions of profitability with respect to the service price. The first corresponds to an aggressively low service price designed to stimulate large demand, and the second corresponds to a less aggressive behavior with high prices and smaller demand. These profitability regions expand when an additional option to upgrade technology is considered. In the absence of options the model recommends defensive behavior with high pricing, while flexibility embedded in options allows more aggressive stimulation of demand with lower prices.

32.3.2 Design of access network

We consider the topic of section 32.2.1 from the network level. The model from the technological level allowed us to understand which user population can have a guaranteed quality of service from an access server with given technological characteristics. On the basis of this information the network-level design should answer the following questions. Given the present demand and future projections for a given region, what type of servers and how many should a network operator choose? Given a description of the typical user populations with given demand patterns, what kind of access servers should the equipment manufacturer produce? In this section we describe a stochastic programming model which helps to answer these questions [7].

We focus on the core of the network access design, which is the design of the first statistical multiplexer stage. The model will support decisions on the number of multiplex-

ers, the bandwidth they carry, and a recommended composition of the traffic they serve. Such a model with two time periods—the present and the future—incorporates some robustness against unexpected traffic evolution; otherwise the design can become obsolete less or more rapidly, and considerable problems due to the system reconfigurations can arise. The present demand is relatively well known. The information available at present about the future demand is available in the form of several demand scenarios. The decision about access design should be made at the present only on the basis of scenario probabilities, without knowledge about which of the demand scenarios will materialize. When the demand will become known in the future, some corrective action will be taken with the aim to ensure the required grade of service. The objective is to select a design which provides a cost-effective solution from the point of view of both the present costs (design implementation) and the future costs (design correction).

Evolution of traffic patterns

- Traffic generated by different users is statistically independent.
- Traffic generated by any of the users belongs to one of the well-defined traffic classes whose characteristics are known. These classes are defined by simple parameters like maximal and average bandwidth. Initially there are N traffic classes ν_i, $i = 1 : N$; in the future there are M traffic classes μ_j, $j = 1 : M$. Some future classes may coincide with the past classes, while some others may be completely new.
- Users presently belonging to class ν_i can pass in the future to class μ_j; new users can appear in class μ_j.
- Present traffic classes are known; for the future traffic classes there are K different scenarios ϑ_k, $k = 1 : K$, described as triples

$$\vartheta_k = \{X_0^k, \alpha_{ij}^k, p^k\}, k = 1 : K, i = 0 : N, j = 1 : M, \sum_{j=1}^{M} \alpha_{ij}^k = 1, \sum_{j=1}^{M} p^k = 1,$$

  where X_0^k is the number of new users under scenario ϑ_k, $\alpha_{ij}^k \geq 0$ for $i = 1 : N$ is the portion of the users of class ν_i which pass to class μ_j under scenario ϑ_k and for $i = 0$ is the portion of the new users which belong to the class μ_j, and $p^k \geq 0$ is the probability of scenario ϑ_k.

Denoting the number of users which belong to the present traffic class ν_i by X_i, $i = 1 : N$, and the number of users which belong to the future traffic class μ_j under scenario ϑ_k by Y_j^k, we obtain the following relation between these two quantities:

$$\sum_{i=1}^{N} \alpha_{ij}^k X_i + \alpha_{0j}^k X_0^k = Y_j^k, \ j = 1 : M. \tag{32.30}$$

Decision variables We consider the case when all multiplexers from the first statistical multiplexing stage handle a fixed mixture of sources. The more general case which includes multiplexers of different types can be handled similarly. Thus, we have to define the initial number n_1 of multiplexers to install, the number of multiplexers n_2^k to add later in the future when the demand scenario ϑ_k becomes known, the bandwidth a of one multiplexer, and the mix of sources served by an arbitrary multiplexer at present and in the future. x_i is the number of users of class ν_i supported by one multiplexer at present, and y_j^k is the number of users of class μ_j supported by one multiplexer in the future under scenario ϑ_k. The vectors $x = (x_1, \ldots, x_N)$ and $y^k = (y_1^k, \ldots, y_M^k)$ describe the source mix. These decision parameters satisfy the quality constraints of two types. The first one states that all users should be covered, now and in the future

$$n_1 x_i \geq X_i, \ i = 1 : N, \tag{32.31}$$

$$(n_1 + n_2^k) y_j^k \geq Y_j^k, \ j = 1 : M, \ k = 1 : K, \tag{32.32}$$

where Y_j^k is defined in (32.30). The second group of constraints should ensure that each multiplexer provides the required grade of service. The quality of service is characterized by the known function $f(a, z)$, where a is the multiplexer bandwidth and z is the source mix supported by a given multiplexer. There is an admissible bound γ on the grade of service. This yields the following representation for the quality of service constraints for the present and the future:

$$f(a, x) \leq \gamma, \tag{32.33}$$

$$f(a, y^r) \leq \gamma, \ r = 1 : R. \tag{32.34}$$

The gather expression for the function $f(a, z)$ or the algorithm for its computation is provided by the technological-level design; an example was described in section 32.2.1. If we take the packet loss as the measure of the quality of service, we can use expression (32.10) from that section.

Costs We take into account the costs of initial installation, additional installation, connection of new users, and reconnection of old users:

- Cost of initial installation and connection of users

$$n_1 C_{11} + C_{21} \sum_{i=1}^{N} X_i,$$

where C_{11} is the fixed cost for initial installation of one multiplexer and C_{21} is the initial cost for connecting one user;

- cost of additional installation and connection of users

$$n_2 C_{12} + C_{22} X_0^k,$$

where C_{12} is the fixed cost for additional installation of one multiplexer in the future and C_{21} is the cost for connecting one user in the future;

- cost of reconnecting the users in the future,

$$C_3 n_1 \sum_{j=1}^{M} \max \left\{ 0, \sum_{i=1}^{N} \alpha_{ij}^k x_i - y_i^k \right\},$$

where C_3 is the cost of reconnecting one user in the future. The future costs C_{12}, C_{22}, and C_3 may be only partially known and may differ between different scenarios.

Decision hierarchy Different decision variables are defined by two coordinated optimization problems corresponding to different time scales and different levels of knowledge about demand. Initial installed capacity n_1, multiplexer bandwidth a, and initial source mix x are decided at the present, when the actual future demand scenario is not known. This is the *long-term planning problem.* Additional installed capacity n_2^k and the final source mix y^k supported by one multiplexer are decided in the future when the demand scenario is known. This is the decision correction problem or *recourse problem* in stochastic programming terminology. The optimal value of this problem enters in the expression for the total costs of the long-term planning problem. This recourse problem is defined as follows.

For scenario k, multiplexer bandwidth a, initial installed capacity n_1, and initial traffic mix x find (n_2^k, y^k), which solves

$$Q(k, a, n_1, x) = \min_{(n_2^k, y^k)} C_{12} n_2 + C_{22} X_0^k + C_3 n_1 \sum_{j=1}^{M} \max \left\{ 0, \sum_{i=1}^{N} \alpha_{ij}^k x_i - y_i^k \right\} \quad (32.35)$$

subject to constraints (32.32), (32.34).

The stochastic programming problem with recourse for long-term planning is to find (a, n_1, x), which minimize the present costs and discounted future costs averaged among demand scenarios

$$\min_{a, n_1, x} C_{11} n_1 + \alpha \sum_{k=1}^{K} p^k Q(k, a, n_1, x) \quad (32.36)$$

subject to constraints (32.31), (32.33), where α is a discount coefficient.

It possesses features which place it apart from the vast majority of such problems found in the literature. First, constraints (32.33), (32.34) are nonlinear. Moreover, while variables a, x, y^k can be considered to be continuous, the variables n_1 and n_2 are substantially discrete. Therefore the usual solution approaches based on application of large-scale linear programming to the deterministic equivalent or even Benders decomposition are not applicable here. A version of stochastic random search worked well on this problem.

32.3.3 Design of a backbone connection-oriented network

The objective of the network design is to make a decision about the placement of the network elements in a given geographical area. The network is represented as the collection of nodes with processing capabilities which are connected by links with transmission capacities. Design involves making decisions concerning the placement of nodes of different type and processing capability and placement of links with different capacities. The objective of

design is the satisfaction of communication demand between different nodes under given requirements about the quality of service and taking into account profit, costs, and other considerations. Telecommunication networks have a hierarchical structure, and backbone networks are the top level of this hierarchy responsible for carrying large demand quantities between geographically distributed nodes where demand is collected with the help of local networks.

The design of a backbone network involves considerable investment, and the resulting network should be robust enough to accommodate unpredictable demand variations during the time horizon of a few years. The flexibility required for the adequate reaction to the changing demand patterns is provided by the network management policies. For connection-oriented networks such policies may include construction of a logical network on top of the physical network by reserving transportation capacity between different nodes for different virtual paths in the network. While the change in the physical network requires considerable investment and time, the change in the logical network can be performed relatively quickly and cheaply following a change of the demand pattern. The design of the physical network should take into account this possibility, and the stochastic programming approach provides the necessary modeling tools for doing so. We describe one such model for the case of connection-oriented networks where the transportation and processing capacity for connection between any given pair of nodes should be reserved before the actual communication can take place. Examples of such networks are traditional telephone networks, broadband ATM networks [9], and mobile networks.

Topology of the physical network

n number of nodes in the network; the nodes are indexed by integer numbers $i = 1 : n$;

x_{ij} link capacity between nodes i and j;

u_i processing capacity of node i;

x vector of all link capacities;

u vector of all processing capacities;

x_{ij}^- lower bound on the transmission capacity between nodes i and j; nonzero x_{ij}^- means that the link between nodes i and j exists already and the objective of design is to identify the necessary network expansion;

x_{ij}^+ upper bound on the transmission capacity between nodes i and j;

u_i^+ upper bound on processing capacity at node i;

u_i^- lower bound on processing capacity at node i.

Demand scenarios

$q = 1, \ldots, m$ index for demand scenarios;

d_{ij}^q communication demand between nodes i and j under scenario q;

z_{ij}^q amount of demand between nodes i and j which is not served under scenario q;

g^q amount of the link capacity required for transportation of one demand unit by an arbitrary link under scenario q;

h^q amount of communication flow through a node served by one unit of processing capacity at this node under scenario q; the communication flow is measured by the amount of transmission capacity necessary to carry this flow.

p_q probability of scenario q.

The dependence of g^q and h^q on the demand scenario allows different service development possibilities.

Topology of the logical network Capacities of the links in the logical network depend on the specific demand scenario because this network should be adapted to demand.

I_{ij} the set of admissible paths between nodes i and j;

m_{ij} number of different paths in the set I_{ij};

b_{ijr} path number r from the set I_{ij}, $r = 1 : m_{ij}$: it is represented as a sequence of pairs of nodes where each pair identifies the link which belongs to the path b_{ijr},

$$b_{ijr} = ((i, j_{r1}), (j_{r1}, j_{r2}), \ldots, (j_{rt_r}, j)),$$

where t_r is the number of hops in the path b_{ijr};

y_{ijr}^q capacity of path b_{ijr} between nodes i and j under scenario q.

Costs The decision paradigm of the minimization of the network costs under constraints on quality of service and demand satisfaction is adopted here. Different objectives like maximization of revenue or maximization of profit can be considered.

c_{ij}^v variable cost of installation of one unit of link capacity between nodes i and j;

c_{ij}^f fixed cost of installation of link capacity between nodes i and j;

c_i^v variable cost of installation of one unit of processing capacity in node i;

c_i^f fixed cost of installation of processing capacity in node i;

e_{ij}^q opportunity cost of not meeting one unit of demand between nodes i and j under scenario q.

Decision structure There are two coordinated decision problems here which fit the paradigm of stochastic programming problems with recourse. The decision to construct the physical network is taken now. Besides decisions about the specific values of transmission capacities x_{ij} and processing capacities u_i, this decision involves also the logical decisions to install capacities or not. Due to the presence of transaction costs these decisions should be modeled by binary variables. v_{ij} is the binary variable which equals 1 if the link capacity between nodes i and j is increased above the already existing level x_{ij}^- and zero otherwise. w_i is the binary variable which equals 1 if the processing capacity at node i is increased above the already existing level u_i^- and zero otherwise.

The design of the physical network is taken before the actual demand patterns become known only on the basis of the information about demand scenarios. Therefore the objective is to minimize the current costs of network installation and discounted future costs of the network adaptation to demand, averaged among demand scenarios. The design problem is to find v_{ij}, w_i, x_{ij}, u_i for all i, j that solve

$$\min_{\substack{v_{ij},w_i,\\ x_{ij},u_i}} \sum_{(i,j)} (c_{ij}^f v_{ij} + c_{ij}^v x_{ij}) + \sum_i (c_i^f w_i + c_i^v u_i) + \alpha \sum_{q=1}^m p_q Q(q,x,u), \tag{32.37}$$

$$x_{ij}^- \le x_{ij} \le x_{ij}^- + (x_{ij}^+ - x_{ij}^-) v_{ij} \ \forall (i,j), \tag{32.38}$$

$$u_i^- \le u_i \le u_i^- + (u_i^+ - u_i^-) w_i, \ i = 1:n. \tag{32.39}$$

The objective function (32.37) includes fixed and variable costs for installation of processing capacities at nodes, transmission capacities at links, and averaged costs of the network adaptation to demand discounted with the discount coefficient α. Constraints (32.38), (32.39) impose bounds on capacities and connect logical and continuous decision variables.

The cost $Q(q,x,u)$ of the network adaptation to a given demand pattern q is obtained by solving the recourse problem, which is the design problem of the logical network for this demand pattern. This problem will be solved repeatedly during the lifetime of the physical network as a new demand scenario emerges.

Given capacities x_{ij}, u_i of the physical network and demand scenario q, find the link capacities y_{ijr}^q of the logical network by solving

$$Q(q,x,u) = \min_{y_{ijr}^q, z_{ij}^q} \sum_{i,j=1}^n e_{ij}^q z_{ij}^q, \tag{32.40}$$

$$\sum_{r \in I_{ij}} y_{ijr}^q + g^q z_{ij}^q = g^q d_{ij}^q \ \forall (i,j), \tag{32.41}$$

$$\sum_{\substack{i,j \\ r \in I_{ij}, (k,l) \in b_{ijr}}} y_{ijr}^q \le x_{kl} \ \forall (k,l), \tag{32.42}$$

$$\sum_{\substack{i,j \\ r \in I_{ij}, l, (k,l) \in b_{ijr}, (l,k) \in b_{ijr}}} y_{ijr}^q \le h^q u_k, \ k = 1:n, \tag{32.43}$$

$$z_{ij}^q \ge 0, \ y_{ijr}^q \ge 0. \tag{32.44}$$

The objective function (32.40) represents adaptation costs which consist of the opportunity costs of not meeting demand. The costs of reconfiguring the logical network are assumed to

be either negligible or not dependent on the decision variables. Constraint (32.41) connects the link capacities of the logical network with the served demand. On the left-hand side of constraint (32.42) we have the total communication capacity required by the logical network from the link between nodes k and l, which should not exceed the physical capacity x_{kl}. The left-hand side of constraint (32.43) represents the sum of all ingoing and outgoing communication flow at node k measured by reserved transmission capacity. It is assumed that the required processing capacity at node k is proportional to this flow.

We solve the problem (32.37)–(32.39) via formulating its deterministic equivalent and solving the resulting mixed integer linear program. Decomposition techniques may be obligatory to process problems of realistic dimensions. Possible simplifications include approximation of fixed costs by variable costs which allows one to dispense with binary variables. When the bottleneck is represented by either processing capacities or transmission capacities, the problem can be simplified by considering only the bottleneck capacities. Different variants of problem (32.37)–(32.39) were considered in [14], where the results of numerical experiments were also reported.

32.3.4 Design of a backbone connectionless network

Connectionless networks represent an important class of telecom networks where no capacity reservation is needed to establish communication between different nodes. An important example is the Internet network. The communication between nodes consists of the flow of data packets, which are routed with the help of routing tables and routing algorithms implemented at network nodes. Therefore the logical network can be represented in the form of multicommodity network flow where each commodity corresponds to a given pair of nodes. The design objectives and the design paradigm remain the same as in the case of the connection oriented network, the main difference being in the formulation of the network adaptation problem (32.40)–(32.44).

The topology of the logical network is represented by y^q_{ijlk}, which is the communication flow between nodes i and j which passes through a link between nodes l and k under scenario q. The problem of design of the physical network (32.37)–(32.39) remains the same, while in the network adaptation problem (32.40)–(32.44) constraints (32.41)–(32.44) are substituted by the following constraints:

$$\sum_{l:l\neq k} y^q_{ijkl} - \sum_{l:l\neq k} y^q_{ijlk} = 0 \;\forall (i,j), \forall k:\; k \neq i, j, \tag{32.45}$$

$$\sum_{l:l\neq i} y^q_{ijil} - \sum_{l:l\neq i} y^q_{ijli} + g^q z^q_{ij} = g^q d^q_{ij} \;\forall (i,j), \tag{32.46}$$

$$\sum_{l:l\neq j} y^q_{ijjl} - \sum_{l:l\neq j} y^q_{ijlj} - g^q z^q_{ij} = -g^q d^q_{ij} \;\forall (i,j), \tag{32.47}$$

$$\sum_{(i,j)} y^q_{ijkl} + \sum_{(i,j)} y^q_{ijlk} \leq x_{kl} \;\forall (k,l), \tag{32.48}$$

$$\sum_{(i,j),l:l\neq k} y^q_{ijkl} + \sum_{(i,j),l:l\neq k} y^q_{ijlk} \leq h^q u_k,\; k = 1:n, \tag{32.49}$$

$$z^q_{ij} \geq 0,\; y^q_{ijkl} \geq 0. \tag{32.50}$$

Constraints (32.45)–(32.47) are the network flow continuity constraints. In the left-hand side of constraint (32.48) we have the total communication capacity required by the logical network from the link between nodes k and l, which should not exceed the physical capacity x_{kl}. The left-hand side of constraint (32.49) represents the sum of all ingoing and outgoing communication flow at node k. It is assumed that the required processing capacity at node k is proportional to this flow.

32.3.5 Incorporating reliability considerations

An important source of uncertainty inherent in the telecommunication networks is presented by possible failures of links and nodes, which can occur due to a variety of reasons; see [18]. Therefore reliability and dependability of networks is a serious design issue. We show that the reliability considerations can be naturally incorporated into the stochastic programming modeling approach. We show how to extend the models of sections 32.3.3 and 32.3.4 to obtain a reliable network design.

The key is to represent the possible link and node failures in the form of failure scenarios which are similar to demand scenarios described in the previous sections. Such scenarios can be independent from demand scenarios or can be combined with them. Demand scenario q is described by the following quantities:

- α_{ij}^q is a portion of the link capacity between nodes i and j which remains operational under failure scenario q, $0 \le \alpha_{ij}^q \le 1$;
- β_k^q is a portion of the processing capacity at node k which remains operational under failure scenario q, $0 \le \beta_k^q \le 1$;
- p_q is a probability of scenario q.

Among all the failure scenarios there will always be one scenario $q = 1$ corresponding to normal operation when $\alpha_{ij}^q = 1$, $\beta_k^q = 1$ for all links (i, j) and all nodes k. All other scenarios will correspond usually to the failure of one given link or node because in the vast majority of practical situations the simultaneous failure of several links or nodes is unlikely.

The models from sections 32.3.3 and 32.3.4 remain the same, the only difference being a slight modification of capacity constraints. For a connectionless network constraints (32.48), (32.49) are substituted by

$$\sum_{(i,j)} y_{ijkl}^q + \sum_{(i,j)} y_{ijlk}^q \le \alpha_{kl}^q x_{kl} \; \forall (k, l), \tag{32.51}$$

$$\sum_{(i,j),l:l\neq k} y_{ijkl}^q + \sum_{(i,j),l:l\neq k} y_{ijlk}^q \le h^q \beta_k^q u_k, \; k = 1 : n. \tag{32.52}$$

For connection-oriented networks similar modifications should be made to constraints (32.42), (32.43).

Figure 32.2 shows dependence of the optimal network costs on the opportunity costs of not meeting demand. The specific model for which these costs were calculated was the design of a connectionless reliable network. The opportunity costs $e_{ij}^q \equiv e$ are the same for all links and all scenarios.

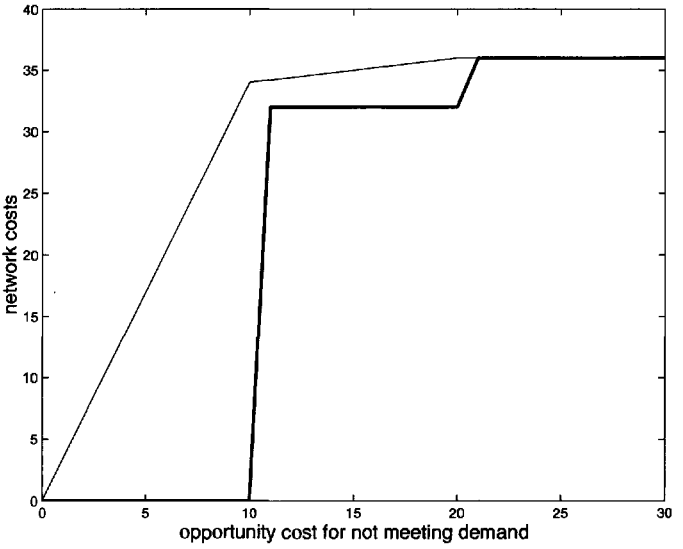


Figure 32.2. *Dependence of the optimal network costs on the opportunity costs of not meeting demand.*

The thin line in Figure 32.2 represents the dependence of the total network costs on the unit opportunity costs e. These costs equal the optimal solution of the problem (32.37)–(32.39), where $Q(q, x, u)$ is the optimal solution of the problem (32.40) with constraints (32.45)–(32.47), (32.50), (32.51), (32.52). These costs grow linearly for low-opportunity costs e; after some threshold they start to grow more slowly and after another threshold they stop growing, which means that all demand is satisfied after that threshold. The dependence of the total costs on opportunity costs is concave. The thick line in the figure represents the corresponding dependence of the pure network component of costs, namely, the sum of the first two terms in (32.37). It shows some interesting phenomena, which can be observed also in the common practice of network development. When the unit opportunity costs of not meeting demand are low, no network is built and all the costs consist of penalties for not meeting demand. After the unit opportunity costs reach some threshold, a network is built which satisfies a large portion of demand, and this network remains unchanged with further growth of opportunity costs until another threshold is reached. After that second threshold a network is built which satisfies all demand. An important observation is that the network is upgraded not incrementally but in a few large steps which corresponds to the industrial practice of telecom network deployment. Stochastic programming models coupled with the estimation of opportunity costs can help with the timing of decisions to upgrade the network.

32.3.6 Planning capacity expansion of a mobile network

The past few years were characterized by exponential growth of mobile networks driven by exponential growth of demand. The market was capable of absorbing practically any

capacity which mobile operators could offer. Things are changing now with the market for voice services being close to saturation in many countries while the market for future broadband wireless services is uncertain. Mobile operators are becoming more attentive to optimal network planning. The objective is to avoid potential losses which may be caused from one side by deployment of excessive capacity and from the other side by deterioration of quality of service and the consequent loss of customers who may turn to the competition. Stochastic programming models can provide network designers with balanced and robust network expansion alternatives which take into account uncertainties in demand development.

In an example, the objective is to develop a plan for the expansion of a mobile network based on the demand forecast. The time horizon of the network expansion is 1 to 2 years, while the plan itself is revised periodically, for example each quarter. The expansion plan consists of two components: establishing new channels (TRX) in existing cells and establishing new cells. Establishing new channels is a relatively minor matter compared to establishing a new cell, from the point of view of both expenditure and time. Establishing a cell is a complex process which may take up to 6 months and considerable resources. It consists of the following steps:

- Preparatory period during which the necessary agreements are made; this takes approximately half the time and a small percentage of the resources.
- Building of a cell, which takes one-third of the time and two-fifths of the resources.
- Putting the cell into operation, which takes one-sixth of the time and three-fifths of the resources.

It is possible to suspend the establishment of a cell after each of these steps and resume it again after some time. This contains a source of additional flexibility which is possible to exploit to diminish the reaction time to demand changes and avoid unnecessary investment. In particular, it is possible to have pools of semifinished cells on different levels of preparedness and to invest into further establishment only when demand requires this. The model which we present below helps to exploit such a possibility. The network expansion process is extended over several time periods $t = 1, 2, \ldots, T$, which in this setting can have the length of 1 month, while T may take values between 6 and 18.

Demand description It is the key input in the network expansion model. As in the previous sections, demand forecasts are represented in the form of scenarios. It is assumed that from times $t = 1$ to t_1 demand d_t^0 is known. After this, demand is uncertain, and this uncertainty is described via scenarios $k = 1 : K$, which describe both demand development uncertainty and seasonal variations and variations due to holidays, special events, etc.

d_t^k demand forecast at time t under scenario k, where $k = 0$ for $t = 1 : t_1$ and $k = 1 : K$ for $t = t_1 + 1, \ldots, T$;

v_t^k demand which will not be satisfied at time t under scenario k;

c_v opportunity cost of not meeting one unit of demand;

p_k probability/frequency of scenario k, $k = 1 : K$.

Network description

a_t amount of capacity necessary to satisfy one unit of demand at time t, which can be variable due to introduction of new services;

C capacity of one cell;

D capacity of one TRX;

Δ_z time necessary for preparation of cell agreement;

c_z cost of preparation of cell agreement;

Δ_y time necessary for building of a cell;

c_y cost of building of a cell;

Δ_x time necessary for activation of a cell;

c_x cost of activation of a cell;

Δ_w time necessary for setting up of a new TRX;

c_w cost of setting up of a new TRX;

α_t coefficient used to discount costs at time t to the present.

During time horizon $t = 1, \ldots, t_1$, *network dimension and dimensioning decisions* are described by the following quantities, which refer to the beginning of period t and which depend on demand scenario k, where $k = 0$ for $t = 1 : t_1$ and $k = 1 : K$ for $t = t_1+1, \ldots, T$.

X_t^k number of cells in the working condition;

Y_t^k number of built but not activated cells;

Z_t^k number of cell agreements;

W_t^k number of TRX;

x_t^k number of new cells to be activated; for $t < 1$ this is a model input resulting from previous decisions;

y_t^k of new cells to be built;

z_t^k number of new cell agreements to be prepared;

w_t^k number of new TRX to be set up;

$Z_{\max}$ maximal number of cell agreements to initiate at any given time period.

Development of the network capacity during time. The following equations describe the development of fully operational cells, built cells, cell agreements, and TRX over time:

$$X_t^k = X_{t-1}^k + x_{t-\Delta_x}^k, \tag{32.53}$$

$$0 \le x_t^k \le Y_t^k, \tag{32.54}$$

$$Y_t^k = Y_{t-1}^k + y_{t-\Delta_y}^k - x_{t-1}^k, \tag{32.55}$$

$$0 \le y_t^k \le Z_t^k, \tag{32.56}$$

$$Z_t^k = Z_{t-1}^k + z_{t-\Delta_z}^k - y_{t-1}^k, \tag{32.57}$$

$$0 \le z_t^k \le Z_{\max}, \tag{32.58}$$

$$W_t^k = W_{t-1}^k + w_{t-\Delta_w}^k, \tag{32.59}$$

$$w_t^k \ge 0, \tag{32.60}$$

where $k = 0$ for the time horizon $t = 1 : t_1$ when demand is assumed to be known and $k = 1 : K$ for the time horizon $t = t_1 + 1 : T$ when demand is uncertain. The following relations connect these time horizons:

$$X_{t_1}^k = X_{t_1}^0,\ Y_{t_1}^k = Y_{t_1}^0,\ Z_{t_1}^k = Z_{t_1}^0,\ W_{t_1}^k = W_{t_1}^0,\ k = 1 : K, \tag{32.61}$$

$$x_{t-\Delta_x}^k = x_{t-\Delta_x}^0, t_1 \le t \le t_1 + \Delta_x, k = 1 : K, \tag{32.62}$$

$$y_{t-\Delta_y}^k = y_{t-\Delta_y}^0, t_1 \le t \le t_1 + \Delta_y, k = 1 : K, \tag{32.63}$$

$$z_{t-\Delta_z}^k = z_{t-\Delta_z}^0, t_1 \le t \le t_1 + \Delta_z, k = 1 : K, \tag{32.64}$$

$$w_{t-\Delta_w}^k = w_{t-\Delta_w}^0, t_1 \le t \le t_1 + \Delta_w, k = 1 : K. \tag{32.65}$$

The relation between capacity and satisfied demand is

$$DW_t^k + a_t v_t^k \ge a_t d_t^k,\ v_t^k \ge 0, \tag{32.66}$$

$$D(W_t^k + w_t^k) \le CX_t^k. \tag{32.67}$$

Two types of costs are considered: the costs of network expansion and the opportunity costs of not meeting demand. Additional costs could be considered, e.g., maintenance of cells and cell agreements.

The capacity expansion decision is taken at the beginning of time period $t = 1$ from the point of view of minimization of total costs, which include costs of network expansion and opportunity costs of not meeting demand during time horizon $t = 1 : t_1$ plus costs of further network expansion contingent on demand scenarios during horizon $t = t_1 + 1 : T$ averaged over scenarios and discounted to time $t = 1$. More formally, find $x_t^0, y_t^0, z_t^0, w_t^0, X_t^0, Y_t^0, Z_t^0, W_t^0, v_t^0, t = 1 : t_1$, from the solution of

$$\min_{\substack{x_t^0, y_t^0, z_t^0, w_t^0, \\ X_t^0, Y_t^0, Z_t^0, W_t^0, v_t^0}} \sum_{t=1}^{t_1} \alpha_t (c_x x_t^0 + c_y y_t^0 + c_z z_t^0 + c_w w_t^0) + \sum_{t=1}^{t_1} \alpha_t c_v v_t^0$$

$$+ \sum_{k=1}^{K} p_k Q^k(x^0, y^0, z^0, w^0, X^0, Y^0, Z^0, W^0) \tag{32.68}$$

subject to (32.53)–(32.60), (32.66), (32.67), where $k = 0$. The arguments of the function $Q^k(\cdot)$ represent the vectors with components indexed by time, for example, $x^0 = (x_1^0, x_2^0, \ldots, x_{t_1}^0)$. The function $Q^k(\cdot)$ represents the optimal costs of further network expansion during time horizon $t = t_1 + 1 : T$ for a given demand scenario k. In stochastic programming terminology this is a recourse problem

$$Q^k(x^0, y^0, z^0, w^0, X^0, Y^0, Z^0, W^0)$$

$$= \min_{\substack{x_t^k, y_t^k, z_t^k, w_t^k, \\ X_t^k, Y_t^k, Z_t^k, W_t^k, v_t^k}} \sum_{t=t_1+1}^{T} \alpha_t (c_x x_t^k + c_y y_t^k + c_z z_t^k + c_w w_t^k) + \sum_{t=t_1+1}^{T} \alpha_t c_v v_t^k \qquad (32.69)$$

subject to constraints (32.53)–(32.67). Problems (32.68)–(32.69) can be transformed into their deterministic equivalents and solved by linear programming software. We utilized the capabilities of the Excel spreadsheet for our solution.

The purpose of the model described above is to provide advice about an aggregated decision about investment into network expansion. It includes aggregated network characteristics and capacity constraints (32.67) which describe the capacity of the whole network. The inherently integer variables, like the number of cells or the number of TRX, were substituted by their continuous approximations. In a more detailed model, constraints (32.67) should be considered for a group of similar cells or even for a given cell.

32.4 Enterprise level

The solutions of the network design problems considered in the previous section depend on a number of external parameters which were considered to be fixed. Some of these parameters derive from decisions which can be taken only on the strategic level through consideration of the whole environment in which the enterprise operates. Examples of such parameters are pricing of services, the total amount of resources allocated for investment, and other quantities which characterize the strategy of the enterprise. The importance of this enterprise decision level has grown considerably during recent years due to the increase in complexity of the telecommunications environment due to the convergence process with computer industry and content provision, deregulation, and technological development. It is characterized by a growing uncertainty whose main sources are unpredictable market response to the new technologies and services, decreasing life cycles of products, and actions of competition. Stochastic programming models represent a natural methodology for decision support on this level. However, they should be enhanced by selected ideas from game theory in order to be capable of reflecting uncertainty which stems from actions of other decision makers.

32.4.1 Network operators and virtual providers: Service pricing

The model developed refers to the situation represented in Figure 32.3. The environment consists of the network operator (NO), virtual service provider (VNO), and a population of customers. Both NO and VNO provide a service to customers, who can decide to subscribe to this service, change provider, or discontinue use of the service altogether.

Service providers decide the price for their service. The NO possesses the network which is necessary for the service provision. The VNO does not have a network, and to provide service it has to lease the necessary network capacity from the NO. Therefore the VNO has to decide how much capacity to lease, and the NO decides the price to charge for the network capacity. Regulatory bodies may impose bounds on the leasing prices. The VNO may provide additional value to a service which may lead to a market expansion which, in its turn, may make the existing network capacity inadequate. Therefore the NO may face the necessity of investing in the network expansion.

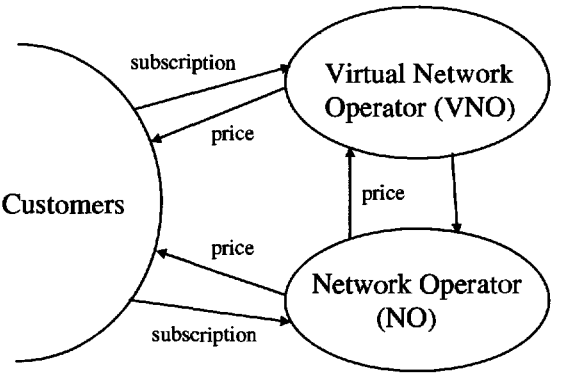


Figure 32.3. *Competition between network operator and virtual network operator.*

The important question is what should be the bounds imposed by regulatory authorities on the price for leasing of the network capacity. From one side they should not be too high in order to permit the VNO to compete with the NO on the service provision. From the other side they should not be too low in order to permit the NO to recover investment expenditures. We develop a stochastic optimization model which helps to answer this question. We take the point of view of the NO which pursues the objective of maximizing its profit by making decisions about service prices, leasing prices, and investment in the network. Its decision model should contain the submodels which describe reactions of customers and the VNO to its policies.

Consider two decision periods $t = 1, 2$. Decisions are taken and implemented at the beginning of each period. During the remaining periods the market and competition reactions are observed which influence revenues and profits.

Decisions of NO and VNO are denoted by vectors y_t and z_t, respectively, $t = 1, 2$, where $y_1 = (y_{11}, y_{12}, y_{13})$, $y_2 = (y_{21}, \ldots, y_{25})$, $z_t = (z_{t1}, z_{t2})$. Components of these vectors have the following meaning:

- y_{t1}, z_{t1} = prices charged for one unit of service by NO and VNO, respectively, during Period t;
- y_{t2} = price charged by NO for one unit of leased capacity;
- y_{t3} = maximal amount of capacity that NO is willing to lease;

- y_{24} = amount of new capacity added to the network at the beginning of Period 2;
- y_{25} = binary variable which equals 1 if decision is taken to expand the network capacity, and zero otherwise; and
- z_{t2} = amount of capacity that VNO decides to lease.

In the profit model of the NO for Period 1, the NO makes the decision about prices at the beginning of the Period 1 without knowing precisely the reaction of customers and competition. The NO's objective is to maximize the average profit, taking into account the possibilities of profit and investment during the second period. The profit of NO is the difference between average revenue and costs during Period 1 with added discounted averaged profit from Period 2. To describe it formally we need the following notations, where we shall associate NO with index $i = 1$ and VNO with index $i = 2$:

- ω_t = vector of parameters which are uncertain for NO at the beginning of Period t. These parameters describe the market and competition reaction and will be defined more precisely when the market and competition models are defined. It is enough that they have a known probabilistic description either in the form of continuous probability distributions or in the form of scenarios with given probabilities.
- d_{ti} = demand for service provided by operator i during Period t measured by the network capacity required for its satisfaction.
- a = network capacity owned by NO and available for service provision at the beginning of Period 1.
- e_{ti} = cost of serving a unit of demand for operator i during Period t.
- g_{ti} = opportunity cost of not meeting one unit of demand for operator i during Period t.
- h_{ti} = cost of maintenance of one unit of capacity for operator i during Period t. It is owned capacity for NO and leased capacity for VNO.
- b_f = fixed costs associated with network expansion.
- b_v = variable costs per unit of capacity associated with network expansion.
- α = coefficient for discounting the second period profit to the beginning of Period 1.

The profit of the NO is

$$F_{11}(y_1, z_1, d_{11}) = (y_{11} - e_{11}) \min\{d_{11}, a - z_{12}\} + y_{12} z_{12} - g_{11} \max\{0, d_{11} - a + z_{12}\} - a h_{11} + \alpha \mathbb{E}_{\omega_1, \omega_2} Q(y_1, z_1, \omega_1, \omega_2), \tag{32.70}$$

where $Q(y_1, z_1, \omega_1, \omega_2)$ is the profit of the NO during Period 2; this depends on the decisions of both NO and VNO during Period 1 and on uncertain parameters ω_1, ω_2. The profit of the NO during Period 1 is described by the first four terms in (32.70). The first term represents the profit due to provision of service to customers. The second term represents the profit due to leasing of capacity to the VNO. The third term represents the opportunity costs of

not meeting the demand for service provision, while the fourth term does not depend on decision variables and represents the variable network maintenance cost.

An important feature of this profit model which distinguishes it from the models of the previous sections is the presence of two unknown components: decisions z_1 of the VNO and service demand d_{11}. To make this model useful for decision making the NO has to predict both these quantities. Such predictions are the scope of the *market model* and *competition model* which the NO should have. The market model provides the prediction of demand $d_{1i} = d_{1i}(y_1, z_1, \omega_1)$ for the service provided by both operators as a function of their pricing decisions and uncertain parameters. The competition model provides the prediction of decisions $z_1 = z_1(y_1, \omega_1)$ of the VNO as a function of decisions of the NO and uncertain parameters. Demand predictions $d_{12} = d_{12}(y_1, z_1, \omega_1)$ for the service provided by the VNO are used for making this prediction. Substituting these predictions into (32.70), we obtain the profit expression which depends only on the decisions of the NO and on uncertain parameters ω_1 with known probabilistic description. The policy recommendation y_1 for the NO can be obtained by finding the values of y_1 which yield the highest mean profit. This leads to the stochastic optimization problem

$$\max_{y_1} \mathbb{E}_{\omega_1} F_{11}(y_1, z_1(y_1, \omega_1), d_{11}(y_1, z_1(y_1, \omega_1), \omega_1)), \tag{32.71}$$

$$y_{11}^- \le y_{11} \le y_{11}^+, \tag{32.72}$$

$$y_{12}^- \le y_{12} \le y_{12}^+, \tag{32.73}$$

$$0 \le y_{13} \le a, \tag{32.74}$$

where y_{11}^-, y_{11}^+, y_{12}^-, y_{12}^+ are price constraints imposed by regulatory and other considerations. We now give examples of the market model and the competition model.

The simplest market model ignores dependence of demand on prices and considers a finite number of demand scenarios with given probabilities. This is, however, an inadequate description of demand. A step toward more realistic demand representation is in the linear autoregressive model

$$d_{t1} = \max\{0, d_{t-1,1} + d_{t1}^0 + r_{t1}(y_{t-1,1} - y_{t1}) + q_t(z_{t1} - y_{t1})\}, \tag{32.75}$$

$$d_{t2} = \max\{0, d_{t-1,2} + d_{t2}^0 + r_{t2}(z_{t-1,1} - z_{t1}) + q_t(y_{t1} - z_{t1})\}, \tag{32.76}$$

where $t = 1, 2$. Here d_{0i} is demand for the service of operator i prior to the beginning of Period 1, which is assumed to be known; y_{01} and z_{01} are some initial reference service prices for both operators; d_{ti}^0 is the component of the demand change for operator i during Period t which is not related to the price changes; r_{ti} is an additional demand obtained/lost due to a unit change of the price of operator i; and q_t is the flow of demand between the operators caused by a unit price difference between them. Parameters d_{ti}^0, r_{ti}, q_t are not known with certainty to the NO and constitute part of the vector ω_t of unknown parameters. We have used the linear model due to its simplicity, but nonlinear models are also possible.

The competition model summarizes the knowledge which the NO has about the objectives of the VNO. In the simplest case it is assumed that the VNO at time period t wants to maximize its expected current profit $F_{t2}(y_t, z_t, d_{t2})$, which has a structure similar to the profit of NO

$$F_{t2}(y_t, z_t, d_{t2}) = (z_{t1} - e_{t2}) \min\{d_{t2}, z_{t2}\} - (y_{t2} + h_{t2}) z_{t2} - g_{t2} \max\{0, d_{t2} - z_{t2}\}, \tag{32.77}$$

where the first term represents the profit derived from service provision, the second term reflects expenditures due to network leasing, and the third reflects opportunity costs for not meeting demand. Parameters e_{t2}, h_{t2}, and g_{t2} are uncertain for the NO and represent another part of the components of the vector of uncertain parameters ω_t. The dependence of demand d_{t2} on prices of both operators is obtained from the market model (32.75)–(32.76) and substituted into (32.77). After this the prediction $z_t(y_t, \omega_t)$ of response of the VNO to decisions of the NO is obtained as a solution of

$$\min_{z_t} F_{t2}(y_t, z_t, d_{t2}(y_t, z_t, \omega_t)), \tag{32.78}$$

$$z_{t1}^- \le z_{t1} \le z_{t1}^+, \tag{32.79}$$

$$0 \le z_{t2} \le y_{t3}, \tag{32.80}$$

where z_{t1}^-, z_{t1}^+ are price bounds imposed by regulation and other considerations.

The profit model of NO for Period 2 reflects the future profits $Q(y_1, z_1, \omega_1, \omega_2)$ in the total profit of the NO. This will make the first-period decision more forward-looking and able to facilitate an eventual adaptation to changing market circumstances by investment in the network expansion. The profit of the NO during Period 2 is

$$\begin{aligned} F_{21}(y_2, z_2, d_{21}, \omega_2) &= (y_{21} - e_{21}) \min\{d_{21}, a - z_{22} + y_{24}\} + y_{22} z_{22} \\ &- g_{21} \max\{0, d_{21} - a + z_{22} - y_{24}\} - h_{21}(a + y_{24}) - b_v y_{24} - b_f y_{25}, \end{aligned} \tag{32.81}$$

where the network expansion occurs at the beginning of the Period 2 and the total network capacity available for service provision during Period 2 is $a + y_{24}$. This expression for the profit is similar to (32.70) and contains two new last terms which reflect variable and fixed costs related to the expansion of the network. Parameters e_{21}, g_{21}, h_{21}, b_v, b_f can be uncertain for the NO at the beginning of Period 1 and constitute additional components of the vector ω_2, others being the components similar to those of ω_1. Prediction $d_{21} = d_{21}(y_2, z_2, \omega_2)$ of demand is obtained from the demand model (32.75)–(32.76), and prediction $z_2(y_2, \omega_2)$ of competition response is obtained from the competition model (32.78)–(32.80). These predictions depend also on y_1, z_1, ω_1 through autoregressive relations (32.75)–(32.76), but we omitted this dependence to simplify notation. The value of the future profit is obtained by solving

$$Q(y_1, z_1, \omega_1, \omega_2) = \min_{y_2} F_{21}(y_2, z_2(y_2, \omega_2), d_{21}(y_2, z_2, \omega_2), \omega_2), \tag{32.82}$$

$$y_{21}^- \le y_{21} \le y_{21}^+, \tag{32.83}$$

$$y_{22}^- \le y_{22} \le y_{22}^+, \tag{32.84}$$

$$z_{12} \le y_{23} \le a + y_{24}, \tag{32.85}$$

$$0 \le y_{24} \le M y_{25}, \tag{32.86}$$

where (32.83)–(32.85) are similar to constraints (32.72)–(32.74). Constraint (32.85) reflects the assumption that the NO cannot offer to the VNO less capacity during Period 2 than the amount offered during Period 1. Constraint (32.86) will force the amount of the network extension to zero if the decision not to expand the network was taken. Otherwise it will limit the network expansion to the maximal admissible level M.

The problem (32.71)–(32.74) is an extension of the classical stochastic programming problem with recourse to the case when part of the uncertainty is due to the actions of other decision makers. This adds a new level of complexity in the form of prediction problem (32.78)–(32.80). The problem (32.82)–(32.86) is an extension of the classical recourse problem with the new feature added by prediction problem (32.78)–(32.80). This additional complexity makes the problem far more challenging than traditional linear stochastic problems with recourse. The linearity structure is never present here. However, some general approaches still can be used, notably the transformation of the problem into its deterministic equivalent by representing the uncertain parameters ω_1, ω_2 through a finite number of scenarios. Nonlinear programming software can be used for solution of such a deterministic equivalent. Another promising approach is the stochastic quasi-gradient methods [17].

We finish this section by presenting in Figure 32.4 a typical example of a computation of the profit function $F_1(y_1) = \mathbb{E}_{\omega_1} F_{11}(y_1, z_1(y_1, \omega_1), d_{11}(y_1, z_1(y_1, \omega_1), \omega_1))$ of the NO performed by MATLAB 6.1 with Optimization Toolbox.

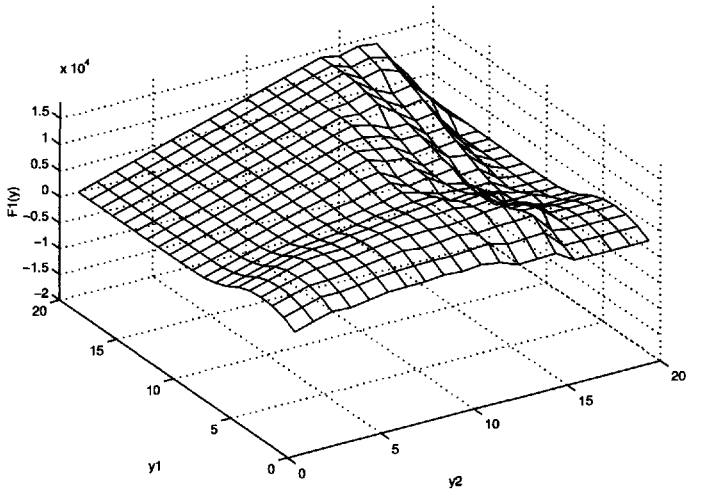


Figure 32.4. *Profit function of NO.*

The vertical axis marked $F1(y)$ shows the values of the profit function of the NO computed according to (32.70). The two horizontal axes marked $y1$ and $y2$ show the values of the service price and leasing price, respectively, of the NO. The figure shows the complex nature of the profit function, which exhibits different patterns in different regions of the price space. This space can be divided into four regions. In the first region both service price and leasing price are moderate, which results in the pattern where both operators have a positive share in the service market. In the second region the service price is high while the leasing price is moderate. In this region the NO has no customers and gets all its profit from leasing capacity to the VNO, which monopolizes the service market. The opposite picture can be observed in the third region, where the leasing price is high while the service price is moderate. In this region the NO becomes a monopolist in service provision and the VNO is squeezed out of the market. Finally, in the fourth region, where both prices are high, the market does not take off at all because the high service price discourages the customers

from subscribing to the service of the NO, while the high leasing price prevents the VNO from offering the service at an attractive price. The good news which can be derived from this and similar examples [3] is that despite its complex nature the profit function has a distinctive structure, and within each of the regions its behavior is close to concave. This circumstance can be exploited in numerical methods.

32.5 Summary

This chapter provides a survey of different applications of stochastic optimization to telecommunications. Some of the models are new. Stochastic programming is a methodology of choice for support of complex network design decisions in the presence of uncertainty. In telecommunications uncertainty is present on all levels of network design, starting from the level of technology, through the level of network design, to the enterprise level, where the top level strategic decisions are taken. Moreover, due to current changes in the industry environment, the adequate treatment of uncertainty is becoming paramount for making competitive design and investment decisions. As different examples show, accumulated modeling and computational experience in solving stochastic optimization problems represent a solid background for deployment of stochastic programming applications in telecommunications. At the same time, more work is needed in development of methodology, in particular in nonlinear and mixed integer stochastic programming models.

Acknowledgments

Thanks are due to Dr. Mario Bonatti of Italtel and Dr. Jan-Arild Audestad of Telenor for useful discussions which helped to shape some of the ideas in this paper.

Bibliography

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice–Hall, Englewood Cliffs, NJ, 1993.

[2] R. Andrade, A. Lisser, N. Maculan, and G. Plateau, *Planning Network Design under Uncertainty with Fixed Charge*, working paper.

[3] J.-A. Audestad, A. Gaivoronski, and A. Werner, *Modeling market uncertainty and competition in telecommunication environment: Network providers and virtual operators*, Telektronikk, 97 (2002), pp. 46–64.

[4] D. P. Bertsekas, *Network Optimization: Continuous and Discrete Models*, Athena Scientific, Fitchburg, MA, 1998.

[5] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*, Springer, New York, 1997.

[6] M. Bonatti and A. Gaivoronski, *Guaranteed approximation of Markov chains with applications to multiplexer engineering in ATM networks*, Ann. Oper. Res., 49 (1994), pp. 111–136.

[7] M. Bonatti, A. Gaivoronski, P. Lemonche, and P. Polese, *Summary of some traffic engineering studies carried out within RACE project R*1044, Eur. Trans. Telecommun., 5 (1994), pp. 207–218.

[8] M. Bonatti and A. A. Gaivoronski, *Worst case analysis of ATM sources with application to access engineering of broadband multiservice networks*, in Proceedings of the 14th International Teletraffic Congress, J. Labetoulle and J. W. Roberts, eds., Elsevier, Amsterdam, 1994, pp. 559–570.

[9] M. De Prycker, *Asynchronous Transfer Mode, Solution for Broadband ISDN*, Prentice–Hall, Englewood Cliffs, NJ, 1995.

[10] M. Dempster and E. Medova, *Evolving system architectures for multimedia network design*, Ann. Oper. Res., 104 (2001), pp. 163–180.

[11] J. Dupačová, *Minimax approach to stochastic linear programming and the moment problem. Recent results*, Z. Angew. Math. Mech., 58 (1978), pp. T466–T467.

[12] Y. Ermoliev, A. Gaivoronski, and C. Nedeva, *Stochastic optimization problems with incomplete information on distribution functions*, SIAM J. Control Optim., 23 (1985), pp. 697–716.

[13] Y. Ermoliev and R. J.-B. Wets, eds., *Numerical Techniques for Stochastic Optimization*, Springer Verlag, Berlin, 1988.

[14] F. Fantauzzi, A. A. Gaivoronski, and E. Messina, *Decomposition methods for network optimization problems in the presence of uncertainty*, in Network Optimization, P. Pardalos, D. Hearn, and W. Hager, eds., Lecture Notes in Econom. and Math. Systems 450, Springer-Verlag, Berlin, 1997, pp. 234–248.

[15] B. Fortz, M. Labbe, and F. Maffioli, *Solving the two-connected network with bounded meshes problem*, Oper. Res., 48 (2000), pp. 866–877.

[16] A. Gaivoronski, *Linearization methods for optimization of functionals which depend on probability measures*, Math. Program. Stud., 28 (1986), pp. 157–181.

[17] A. A. Gaivoronski, *Implementation of stochastic quasigradient methods*, in Numerical Techniques for Stochastic Optimization, Y. Ermoliev and R. J.-B. Wets, eds., Springer-Verlag, Berlin, 1988, pp. 313–352.

[18] B. Gavish and I. Neuman, *Routing in a network with unreliable components*, IEEE Trans. Telecommun., 40 (1992), pp. 1248–1258.

[19] P. Kall and S. Wallace, *Stochastic Programming*, John Wiley, New York, 1994.

[20] S. Karlin and W. J. Studden, *Tchebycheff Systems: With Applications in Analysis and Statistics*, Interscience, New York, 1966.

[21] J. H. B. Kemperman, *The general moment problem: A geometric approach*, Ann. Math. Statist., 39 (1968), pp. 93–122.

[22] J. LABETOULLE AND J. W. ROBERTS, EDS., *The Fundamental Role of Teletraffic in the Evolution of Telecommunications Networks*, Proceedings of the 14th International Teletraffic Congress, Elsevier, Amsterdam, 1994.

[23] J. MEDHI, *Stochastic Models in Queueing Theory*, Academic Press, Boston, 1991.

[24] D. MITRA, K. G. RAMAKRISHNAN, AND Q. WANG, *Combined economic modeling and traffic engineering: Joint optimization of pricing and routing in multi-service networks*, in Proceedings of the 17th International Teletraffic Congress, Salvador, Brazil, Elsevier, Amsterdam, 2001.

[25] S. SEN, R. D. DOVERSPIKE, AND S. COSARES, *Network planning with random demand*, J. Telecommun. Syst., 3 (1994), pp. 11–30.

[26] A. TOMASGARD, J. AUDESTAD, S. DYE, L. STOUGIE, M. V. DER VLERK, AND S. WALLACE, *Modelling aspects of distributed processing in telecommunications networks*, Ann. Oper. Res., 82 (1998), pp. 161–184.

[27] L. TRIGEORGIS, *Real Options: Managerial Flexibility and Strategy in Resource Allocation*, MIT Press, Cambridge, MA, 1996.

[28] W. WHITT, *On approximation for queues,* I: *Extremal distributions*, AT&T Bell Laboratories Tech. J., 63 (1984).

[29] J. YEN, A. SCHAEFER, AND C. SMITH, *A stochastic SONET network design problem*, in Proceedings of the Ninth International Conference on Stochastic Programming, Berlin, 2001.

Index