

Linear Programming: 2: Theory and Extensions

George B. Dantzig
Mukund N. Thapa

Springer

Springer Series in Operations Research

Editors:

Peter W. Glynn Stephen M. Robinson

Springer

New York

Berlin

Heidelberg

Hong Kong

London

Milan

Paris

Tokyo

This page intentionally left blank

George B. Dantzig

Mukund N. Thapa

Linear Programming

2: Theory and Extensions

With 45 Illustrations



Springer

Professor George B. Dantzig
Department of Management Science
and Engineering
Computer Science Department
Stanford University
Stanford, CA 94305
USA
George@LinearProgram.com

Dr. Mukund N. Thapa
President and CEO
Optical Fusion Inc.
P.O. Box 60478
Palo Alto, CA 94306-0478
President and CEO
Stanford Business Software, Inc.
P.O. Box 60398
Palo Alto, CA 94306-0398
Consulting Professor
Department of Management Science
and Engineering
Stanford University
Stanford, CA 94305
USA
Mukund@LinearProgram.com

Series Editors:

Peter W. Glynn
Department of Management Science
and Engineering
Terman Engineering Center
Stanford University
Stanford, CA 94305-4026
USA
glynn@leland.stanford.edu

Stephen M. Robinson
Department of Industrial Engineering
University of Wisconsin–Madison
1513 University Avenue
Madison, WI 53706-1572
USA
smrobins@facstaff.wisc.edu

The Library of Congress has catalogued the first volume as follows
Library of Congress Cataloging-in-Publication Data
Dantzig, George Bernard, 1914–

Linear programming 1 : introduction / George B. Dantzig
& Mukund N. Thapa.

p. cm. — (Springer series in operations research)

Includes bibliographical references and index.

ISBN 0-387-94833-3 (hardcover : alk. paper)

I. Linear programming. I. Thapa, Mukund Narain-Dhami. II. Title.

III. Series.

T57.74.D365 1997

619.7'2—dc20

96-36411

ISBN 0-387-98613-8

Printed on acid-free paper.

© 2003 George B. Dantzig and Mukund N. Thapa

All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer-Verlag New York, Inc., 175 Fifth Avenue, New York, NY 10010, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed in the United States of America.

9 8 7 6 5 4 3 2 1

SPIN 10689385

Typesetting: Pages created by the authors using L^AT_EX/T_EX.

www.springer-ny.com

Springer-Verlag New York Berlin Heidelberg

A member of BertelsmannSpringer Science+Business Media GmbH

ABOUT THE AUTHORS

George B. Dantzig received the National Medal of Science from the President of the United States “for inventing Linear Programming and for discovering the Simplex Algorithm that led to wide-scale scientific and technical applications to important problems in logistics, scheduling, and network optimization, and to the use of computers in making efficient use of the mathematical theory.” He is world famous for his twin discoveries; linear programming and the Simplex Algorithm, which together have enabled mankind for the first time to structure and solve extremely complex optimal allocation and resource problems. Among his other discoveries are the Decomposition Principle (with Philip Wolfe) which makes it possible to decompose and solve extremely large linear programs having special structures, and applications of these techniques with sampling to solving practical problems subject to uncertainty.

Since its discovery in 1947, the field of linear programming, together with its extensions (mathematical programming), has grown by leaps and bounds and is today the most widely used tool in industry for planning and scheduling.

George Dantzig received his master’s from Michigan and his doctorate in mathematics from Berkeley in 1946. He worked for the U.S. Bureau of Labor Statistics, served as chief of the Combat Analysts Branch for USAF Headquarters during World War II, research mathematician for RAND Corporation, and professor and head of the Operations Research Center at the University of California, Berkeley. He is currently professor of Management Science and Engineering and Computer Science at Stanford University. He served as director of the System Optimization Laboratory and the PILOT Energy-Economic Model Project. Professor Dantzig’s seminal work has laid the foundation for the field of systems engineering, which is widely used in network design and component design in computer, mechanical, and electrical engineering. His work inspired the formation of the Mathematical Programming Society, a major section of the Society of Industrial and Applied Mathematics, and numerous professional and academic bodies. Generations of Professor Dantzig’s students have become leaders in industry and academia.

He is a member of the prestigious National Academy of Science, the American Academy of Arts and Sciences, and the National Academy of Engineering.

Mukund N. Thapa is the President & CEO of Optical Fusion, Inc., President of Stanford Business Software, Inc., as well as a consulting professor of Management Science and Engineering at Stanford University. He received a bachelor of technology degree in metallurgical engineering from the Indian Institute of Technology, Bombay, and M.S. and Ph.D. degrees in operations research from Stanford University in 1981. His Ph.D. thesis was concerned with developing specialized algorithms for solving large-scale unconstrained nonlinear minimization problems. By profession he is a software developer who produces commercial software products as well as commercial-quality custom software. Since 1978, Dr. Thapa has been applying the theory of operations research, statistics, and computer science to develop efficient, practical, and usable solutions to a variety of problems.

At Optical Fusion, Inc., Dr. Thapa is developing a multi-point IP-based videoconferencing system for use over networks. The feature-rich system will focus primarily on the needs of users and allow corporate users to seamlessly integrate conferencing in everyday business interactions. At Stanford Business Software, Dr. Thapa, ensures that the company produces high-quality turnkey software for clients. His expert knowledge of user friendly interfaces, data bases, computer science, and modular software design plays an important role in making the software practical and robust. His speciality is the application of numerical analysis methodology to solve mathematical optimization problems. He is also an experienced modeler who is often asked by clients to consult, prepare analyses, and to write position papers. At the Department of Management Science and Engineering, from time to time, Dr. Thapa teaches graduate-level courses in mathematical programming computation and numerical methods of linear programming.

TO

Tobias and Anja Dantzig, my parents, *in memoriam*
Anne S. Dantzig, my wife, and to
the great pioneers that made this field possible:
Wassily Leontief, Tjalling Koopmans, John von Neumann,
Albert Tucker, William Orchard-Hays, Martin Beale.

— George B. Dantzig

Radhika H. Thapa, my wife,
Isha, my daughter, and to
Devi Thapa & Narain S. Thapa, my parents.

— Mukund N. Thapa

This page intentionally left blank

Contents

LIST OF FIGURES	xv
LIST OF TABLES	xvii
PREFACE	xix
DEFINITION OF SYMBOLS	xxiii
1 GEOMETRY OF LINEAR INEQUALITY SYSTEMS AND THE SIMPLEX METHOD	1
1.1 CONVEXITY AND LINEAR INEQUALITY SYSTEMS	1
1.1.1 Affine & Convex Combinations	1
1.1.2 Two-dimensional Convex Regions	3
1.1.3 Line Segments, Rays, and Half Lines	5
1.1.4 General Convex Regions	6
1.1.5 Hyperplanes and Half-Spaces	7
1.1.6 Convexity of Half Spaces and Hyperplanes	8
1.1.7 Convexity of the Set of Feasible Solutions of an LP	9
1.1.8 Convex Polyhedrons, Polytopes, and Cones	9
1.1.9 Separating Hyperplane	11
1.2 SIMPLEX DEFINED	13
1.3 GLOBAL MINIMUM, EXTREME POINTS, AND EDGES	14
1.4 THE SIMPLEX METHOD VIEWED AS THE STEEPEST DESCENT ALONG EDGES	20
1.5 THE SIMPLEX INTERPRETATION OF THE SIMPLEX METHOD	24
1.6 NOTES & SELECTED BIBLIOGRAPHY	31
1.7 PROBLEMS	31
2 DUALITY AND THEOREMS OF THE ALTERNATIVES	43
2.1 THE DUALITY THEOREM	43
2.2 ADDITIONAL THEOREMS ON DUALITY	47
2.2.1 Unboundedness Theorem	47
2.2.2 Miscellaneous Theorems for the Standard Form	48

2.3	COMPLEMENTARY SLACKNESS	49
2.4	THEOREMS OF THE ALTERNATIVES	50
2.4.1	Gordan's Theorem	51
2.4.2	Farkas's Lemma	52
2.4.3	Stiemke's Theorem	53
2.4.4	Motzkin's Transposition Theorem	54
2.4.5	Ville's Theorem	55
2.4.6	Tucker's Strict Complementary Slackness Theorem	56
2.5	NOTES & SELECTED BIBLIOGRAPHY	58
2.6	PROBLEMS	59
3	EARLY INTERIOR-POINT METHODS	67
3.1	VON NEUMANN'S METHOD	70
3.1.1	The von Neumann Algorithm	73
3.1.2	Improving the Rate of Convergence	81
3.1.3	Von Neumann Algorithm as a Variant of the Simplex Algorithm	83
3.2	DIKIN'S METHOD	84
3.2.1	Dikin's Algorithm	87
3.2.2	Convergence of Dikin's Algorithm	89
3.3	KARMAKAR'S METHOD	100
3.3.1	Development of the Algorithm	100
3.3.2	Proof of Convergence	105
3.3.3	The Algorithm Summarized	114
3.3.4	Converting a Standard LP to a Starting Form for the Algorithm	115
3.3.5	Computational Comments	116
3.3.6	Complexity of von Neumann versus Karmarkar Algorithms .	118
3.4	NOTES & SELECTED BIBLIOGRAPHY	119
3.5	PROBLEMS	121
4	INTERIOR-POINT METHODS	123
4.1	NEWTON'S METHOD	123
4.2	THE LINEAR LEAST-SQUARES PROBLEM	127
4.3	BARRIER FUNCTION METHODS	128
4.3.1	The Logarithmic Barrier Function	128
4.3.2	Properties of Barrier Function Methods	130
4.4	THE PRIMAL LOGARITHMIC BARRIER METHOD FOR SOLVING LINEAR PROGRAMS	131
4.4.1	Details of the Method	131
4.4.2	Initial Feasible Solution	134
4.5	PRIMAL-DUAL LOGARITHMIC BARRIER METHODS	134
4.6	RECOVERING A BASIC FEASIBLE SOLUTION	137
4.7	COMPUTATIONAL COMMENTS	139
4.8	NOTES & SELECTED BIBLIOGRAPHY	140
4.9	PROBLEMS	146

5	DEGENERACY	149
5.1	EXAMPLES OF CYCLING	149
5.2	ON RESOLVING DEGENERACY	153
5.3	DANTZIG'S INDUCTIVE METHOD	154
5.4	WOLFE'S RULE	156
5.5	BLAND'S RULE	158
5.6	KRISHNA'S EXTRA COLUMN RULE	160
5.7	ON AVOIDING DEGENERATE PIVOTS	164
5.8	NOTES & SELECTED BIBLIOGRAPHY	166
5.9	PROBLEMS	167
6	VARIANTS OF THE SIMPLEX METHOD	173
6.1	INTRODUCTION	173
6.2	MAX IMPROVEMENT PER ITERATION	176
6.3	DUAL-SIMPLEX METHOD	179
6.4	PARAMETRIC LINEAR PROGRAMS	183
6.4.1	Parameterizing the Objective Function	183
6.4.2	Parameterizing the Right-Hand Side	187
6.5	SELF-DUAL PARAMETRIC ALGORITHM	188
6.6	THE PRIMAL-DUAL ALGORITHM	191
6.7	THE PHASE I LEAST-SQUARES ALGORITHM	197
6.8	NOTES & SELECTED BIBLIOGRAPHY	200
6.9	PROBLEMS	202
7	TRANSPORTATION PROBLEM AND VARIATIONS	207
7.1	THE CLASSICAL TRANSPORTATION PROBLEM	207
7.1.1	Mathematical Statement	208
7.1.2	Properties of the System	208
7.2	FINDING AN INITIAL SOLUTION	213
7.3	FINDING AN IMPROVED BASIC SOLUTION	214
7.4	DEGENERACY IN THE TRANSPORTATION PROBLEM	216
7.5	TRANSSHIPMENT PROBLEM	219
7.5.1	Formulation	219
7.5.2	Reduction to the Classical Case by Computing Minimum Cost Routes	222
7.5.3	Reduction to the Classical Case by the Transshipment Pro- cedure	222
7.6	TRANSPORTATION PROBLEMS WITH BOUNDED PARTIAL SUMS	225
7.7	NOTES & SELECTED BIBLIOGRAPHY	227
7.8	PROBLEMS	228

8	NETWORK FLOW THEORY	231
8.1	THE MAXIMAL FLOW PROBLEM	232
8.1.1	Decomposition of Flows	233
8.1.2	The Augmenting-Path Algorithm for Maximal Flow	234
8.1.3	Cuts in a Network	239
8.2	SHORTEST ROUTE	241
8.3	MINIMUM COST-FLOW PROBLEM	242
8.4	NOTES & SELECTED BIBLIOGRAPHY	243
8.5	PROBLEMS	245
9	GENERALIZED UPPER BOUNDS	251
9.1	PROBLEM STATEMENT	251
9.2	BASIC THEORY	253
9.3	SOLVING SYSTEMS WITH GUB EQUATIONS	253
9.4	UPDATING THE BASIS AND WORKING BASIS	257
9.5	NOTES & SELECTED BIBLIOGRAPHY	264
9.6	PROBLEMS	264
10	DECOMPOSITION OF LARGE-SCALE SYSTEMS	265
10.1	WOLFE'S GENERALIZED LINEAR PROGRAM	267
10.2	DANTZIG-WOLFE (D-W) DECOMPOSITION PRINCIPLE	280
10.2.1	D-W Principle	284
10.2.2	D-W Decomposition Algorithm and Variants	289
10.2.2.1	The D-W Algorithm	289
10.2.2.2	Variants of the D-W Algorithm	290
10.2.3	Optimality and Dual Prices	290
10.2.4	D-W Initial Solution	291
10.2.5	D-W Algorithm Illustrated	292
10.3	BENDERS DECOMPOSITION	299
10.3.1	Dual of D-W Decomposition	299
10.3.2	Derivation of Benders Decomposition	300
10.4	BLOCK-ANGULAR SYSTEM	306
10.5	STAIRCASE STRUCTURED PROBLEMS	308
10.5.1	Using Benders Decomposition	309
10.5.2	Using D-W Decomposition	310
10.5.3	Using D-W Decomposition with Alternate Stages Forming the Subproblems	312
10.6	DECOMPOSITION USED IN CENTRAL PLANNING	313
10.7	NOTES & SELECTED BIBLIOGRAPHY	315
10.8	PROBLEMS	317

11 STOCHASTIC PROGRAMMING: INTRODUCTION	323
11.1 OVERVIEW	324
11.2 UNCERTAIN COSTS	326
11.2.1 Minimum Expected Costs	326
11.2.2 Minimum Variance	327
11.3 UNCERTAIN DEMANDS	329
11.4 NOTES & SELECTED BIBLIOGRAPHY	332
11.5 PROBLEMS	332
12 TWO-STAGE STOCHASTIC PROGRAMS	335
12.1 THE DETERMINISTIC TWO-STAGE LP PROBLEM	335
12.2 THE ANALOGOUS STOCHASTIC TWO-STAGE LP PROBLEM	336
12.3 LP EQUIVALENT OF THE STOCHASTIC PROBLEM (EQ-LP)	337
12.3.1 LP Equivalent Formulation	337
12.3.2 Geometric Description of Benders Decomposition Algorithm	338
12.3.3 Decomposition Algorithm	341
12.3.4 Theory behind the Algorithm	348
12.4 SOLVING STOCHASTIC TWO-STAGE PROBLEMS USING SAM- PLING	349
12.4.1 Overview	349
12.4.2 Naive Sampling	350
12.4.3 Sampling Methodology	351
12.4.4 Estimating Upper Bound z_{UB} for Min z given $x = x^k$	351
12.4.5 Estimating Lower Bound z_{LB} for Min z	352
12.5 USE OF IMPORTANCE SAMPLING	354
12.5.1 Crude (Naive) Monte Carlo Methods	355
12.5.2 Monte Carlo Methods using Importance Sampling	356
12.6 NOTES & SELECTED BIBLIOGRAPHY	360
12.7 PROBLEMS	362
A PROBABILITY THEORY: OVERVIEW	367
A.1 BASIC CONCEPTS, EXPECTED VALUE, AND VARIANCE	367
A.2 NORMAL DISTRIBUTION AND THE CENTRAL LIMIT THEO- REM	370
A.3 CHI-SQUARE DISTRIBUTION, STUDENT'S t -DISTRIBUTION, AND CONFIDENCE INTERVALS	373
A.3.1 Chi-Square Distribution	373
A.3.2 Student's t -Distribution	375
A.3.3 Confidence Intervals	376
A.4 NOTES & SELECTED BIBLIOGRAPHY	377
REFERENCES	379
INDEX	439

This page intentionally left blank

List of Figures

1-1	Vector $(x_1, x_2)^T$	2
1-2	Examples of Two-Dimensional Convex Regions	3
1-3	Additional Examples of Two-Dimensional Convex Regions	3
1-4	Example of an Unbounded Two-Dimensional Convex Set and an Unbounded Three-Dimensional Convex Set	4
1-5	Examples of Two-Dimensional Nonconvex Regions	4
1-6	Points Common to Convex Sets	5
1-7	Example of a Line Segment	6
1-8	A Three-Dimensional Simplex	13
1-9	Local and Global Minima	14
1-10	Extreme Points	15
1-11	Joining Extreme Points	18
1-12	Geometrically the Iterates of the Simplex Algorithm Move Along the Edges of the Convex Set	21
1-13	Geometric Picture of the Distance of a Point to a Boundary	22
1-14	Movement of θ	24
1-15	Geometrically a Linear Program is a Center-of-Gravity Problem	26
1-16	Simplex Associated with an Iteration of the Simplex Algorithm ($m = 2$)	27
1-17	Geometry of the Simplex Algorithm for the Product Mix Problem	28
1-18	Simplex Associated with an Iteration of the Simplex Algorithm ($m = 3$)	30
1-19	Hyperplane H_j and Simplex \mathcal{S}	40
2-1	Illustration of the Duality Gap	45
2-2	Find Basic Feasible Solutions of Dual of Two Variable Primal	63
3-1	The Two-Dimensional Center-of-Gravity Problem: Find a Simplex that Contains the Origin	73
3-2	Finding an Improved Approximation	74
3-3	Convergence under Existence of a Ball \mathcal{B}	77
3-4	Degenerate Two-Dimensional Case	79
3-5	Decreasing x_s^t to Improve the Rate of Convergence	81
3-6	Ellipsoid Centered at $(1, 2)$	86
3-7	Ellipsoid Subproblem Centered at y^t	87

3-8	Comparison of a Move from a Point \hat{x}^t Near the “Center” Versus a Point \bar{x}^t Near the Boundary	101
3-9	Bound for $\hat{c}^T y^{t+1}$	105
4-1	Barrier Function Method: Approach of $x^*(\mu)$ to x^*	129
7-1	Network Representation of the Transportation Problem	209
7-2	Example of Standard Transportation Array	209
7-3	Cycling in the Transportation Problem	216
7-4	Perturbing the Transportation Problem	217
7-5	Example of a Standard Transshipment Array	219
7-6	The Transshipment Problem	221
8-1	A Simple Directed Network	232
8-2	Decomposition of Flows	234
8-3	Original Network and Associated Network with θ Flow	236
8-4	Example to Show Matching	245
8-5	Data for a Max-Flow Problem	246
10-1	Illustration of the Resolution Theorem	283
12-1	Benders Decomposition Applied to EQ-LP	339

List of Tables

2-1	Tucker Diagram (Partitioned)	54
5-1	Hoffman's Example of Cycling (Continued on the Right)	150
5-2	Hoffman's Example of Cycling (Continued from the Left)	151
5-3	Beale's Example of Cycling	152
5-4	Inductive Method and Wolfe's Rule Applied to Beale's Example . . .	157
6-1	Primal-Dual Correspondences	175
6-2	Primal-Simplex and Dual-Simplex Methods	180
9-1	An Example of GUB Constraints and Key Basic Variables	254
9-2	An Example of a Reordered Basis for GUB Constraints	254
12-1	Data for the Grape Grower's Dilemma	364
12-2	Data for Dinner Set Production Schedule	364

This page intentionally left blank

PREFACE

Linear Programming 2 continues where *Linear Programming 1* left off. We assume that the reader has an introductory knowledge of linear programming, for example has read *Linear Programming 1: Introduction* (or its equivalent) and has knowledge of linear algebra (reviewed in the appendices in *Linear Programming 1*). In this volume, we prove all theorems stated and those that were sketched but not proved in *Linear Programming 1*, and we describe various extensions.

Linear Programming 2 is intended to be an advanced graduate text as well as a reference. Portions of *Linear Programming 1* and *Linear Programming 2* have been used in a graduate-level course that we have taught together. The rest of the discussion here summarizes the contents of this volume.

OUTLINE OF CHAPTERS

Chapter 1 (Geometry): In this chapter we study the geometry and properties of linear inequality systems and how they are related to the Simplex Method, which can be described as a movement along the edges of a convex polyhedral set to obtain a global minimum of the objective function, generate a class of feasible solutions for which the objective $z \rightarrow -\infty$, or determine that the convex polyhedral set is infeasible. The important separating hyperplane concepts are also discussed and proved.

Chapter 2 (Duality and Theorems of the Alternatives): We provide proofs for the *Weak* and *Strong* Duality Theorems. This is followed by additional theorems on duality; that is, the *Unboundedness Theorem* and the *Primal/Dual Optimality Criteria*. The chapter also discusses complementary slackness and various Theorems of the Alternatives: Gordan's Theorem, Farkas's Lemma, Stiemke's Theorem, Motzkin's Transposition Theorem, Ville's Theorem, and Tucker's Strict Complementary Slackness Theorem.

Chapter 3 (Early Interior-Point Methods): In this chapter we trace the early development of interior-point methods. The earliest known method is that attributable to von Neumann [1948], followed by Frisch [1957] (only referenced here), and Dikin [1967]. A theoretical breakthrough was due to Khachian [1979] who developed a polynomial-time ellipsoid algorithm (only referenced

here). This was followed by Karmarkar's [1984] polynomial-time interior-point algorithm.

Chapter 4 (Interior-Point Methods): Since the development of Karmarkar's [1984] algorithm several new important practical interior-point algorithms emerged. Among these are the primal logarithmic barrier method, primal-affine algorithm, dual logarithmic barrier method, dual-affine algorithm, and the primal-dual algorithm. All these algorithms are described. The optimal solution obtained by an interior-point method is not necessarily at a vertex; we describe a technique to make it into a vertex.

Chapter 5 (Degeneracy): When degeneracy occurs, it is possible for the Simplex Algorithm to have an infinite sequence of iterations with no decrease in the value of z . The chapter illustrates this with examples due to Hoffman, Beale, and Kuhn. Then various methods for resolving degeneracy are presented: Dantzig's Inductive Methods, Wolfe's Rule, Bland's Rule, and Krishna's Extra Column Rule. This is followed by a technique that attempts to avoid degenerate pivot by making use of an extra objective function and resultant reduced cost calculation.

Chapter 6 (Variants of the Simplex Method): Over the years several variants of the Simplex Algorithm have been proposed as a way to reduce the number of iterations. We start by describing an efficient way of determining an incoming column that yields the maximum improvement per iteration. Next we describe the Dual-Simplex Method, Parametric Linear Programming, Self-Dual Parametric Algorithm, Primal-Dual Algorithm, and a Phase I Least-Squares Algorithm.

Chapter 7 (Transportation Problem and Variations): The Classical Transportation Problem is stated, and various theorems are proved about it. An example is provided for cycling under degeneracy when the most negative reduced cost is used to select an incoming column. This is followed by a discussion of the Transshipment Problem and transportation problems with bounded partial sums.

Chapter 8 (Network Flow Theory): Theorems are proved about the Maximal-Flow problem and the Shortest-Route problem.

Chapter 9 (Generalized Upper Bounds): In this chapter we discuss a variation of the Simplex Algorithm to efficiently solve linear programs that have upper bounds on subsets of variables such that each variable appears in at most one subset. Such constraints are called *generalized upper bounds*.

Chapter 10 (Decomposition): Decomposition is a term to describe breaking a problem into smaller parts and then using a variant of the Simplex Algorithm to solve the entire problem efficiently. The chapter starts by describing Wolfe's Generalized Linear Program (or a linear program with variable coefficients). The Dantzig-Wolfe Decomposition Principle is described for solving

this class of problems. This is followed by a description of Benders Decomposition which is the Dantzig-Wolfe Decomposition applied to the dual. Benders Decomposition is used to solve Stochastic Programs. Next we describe the application of Dantzig-Wolfe Decomposition to solving of Block-Angular systems. Then staircase structured problems are described; we show how to solve such problems using Dantzig-Wolfe Decomposition and Benders Decomposition. Finally, the possible use of decomposition to solve central planning problems is described.

Chapter 11 (Stochastic Programming Introduction): Here we introduce the concept of planning under uncertainty. Simple problems with uncertain demand and uncertain costs respectively are illustrated. This is followed by a discussion of the convexity property of multi-stage problems.

Chapter 12 (Two-Stage Stochastic Programs): An important class of optimization problems arise in dynamic systems that describe activities initiated at time t that have coefficients at time t and time $t + 1$. Such problems, called *dynamic linear programs*, typically have a nonzero submatrix with a staircase structure. The simplest dynamic linear program has only two stages; this is discussed in this chapter.

Appendix A (Probability Theory Overview): In this appendix we introduce some basic concepts and notation of probability theory for use in solving stochastic linear programs.

LINEAR PROGRAMMING 1.

In a graduate course that we have taught together at Stanford, portions of *Linear Programming 1: Introduction* and *Linear Programming 2: Theory & Extensions* have been used.

Professor George B. Dantzig
Stanford, California
USA

Dr. Mukund N. Thapa
Palo Alto, California
USA

This page intentionally left blank

DEFINITION OF SYMBOLS

The notation described below will be followed in general. There may be some deviations where appropriate.

- Uppercase letters will be used to represent matrices.
- Lowercase letters will be used to represent vectors.
- All vectors will be column vectors unless otherwise noted.
- Greek letters will typically be used to represent scalars.

\mathbb{R}^n	– Real space of dimension n .
c	– Coefficients of the objective function.
A	– Coefficient matrix of the linear program.
B	– Basis matrix (nonsingular). It contains the basic columns of A .
N	– Nonbasic columns of A .
x	– Solution of the linear program (typically the current one).
x_B	– Basic solution (typically the current one).
x_N	– Nonbasic solution (typically the current one).
(x, y)	– The column vector consisting of components of the vector x followed by the components of y . This helps in avoiding notation such as $(x^T, y^T)^T$.
L	– Lower triangular matrix with 1s on the diagonal.
U	– Upper triangular matrix (sometimes R will be used).
R	– Alternative notation for an upper triangular matrix.
D	– Diagonal matrix.
$\text{Diag}(d)$	– Diagonal matrix. Sometimes $\text{Diag}(d_1, d_2, \dots, d_n)$ will be used.
D_x	– $\text{Diag}(x)$.
I	– Identity matrix.

e_j	– j th column of an identity matrix.
e	– Vector of 1s (dimension will be clear from the context).
E_j	– Elementary matrix (j th column is different from the identity).
$\ v\ $	– The 2-norm of a vector v ; i.e., $\ v\ _2 = \sqrt{v^T v}$.
$\ v\ _1$	– The 1-norm of a vector v ; i.e., $\ v\ _1 = \sum_{i=1}^n v_i $.
$\ v\ _\infty$	– The ∞ -norm of a vector v ; i.e., $\ v\ _\infty = \max_{i=1, \dots, n} v_i $.
$\ A\ $	– The 2-norm of an $m \times n$ matrix A ; i.e., $\ A\ _2 = \sqrt{\lambda_{\max}(A^T A)}$.
$\ A\ _1$	– The 1-norm of an $m \times n$ matrix A ; i.e., $\ A\ _1 = \max_{j=1, \dots, n} \sum_{i=1}^m a_{ij} $.
$\ A\ _\infty$	– The ∞ -norm of an $m \times n$ matrix A ; i.e., $\ A\ _\infty = \max_{i=1, \dots, m} \sum_{j=1}^n a_{ij} $.
$\det(A)$	– Determinant of the matrix A .
$A_{\bullet j}$	– j th column of A .
$A_{i \bullet}$	– i th row of A .
B^t	– The matrix B at the start of iteration t .
$B[t]$	– Alternative form for the matrix B^t .
\bar{B}	– Update from iteration t to iteration $t + 1$.
B_{ij}^{-1}	– Element (i, j) of B^{-1} .
$X \subset Y$	– X is a proper subset of Y .
$X \subseteq Y$	– X is a subset of Y .
$X \cup Y$	– Set union, that is, the set $\{\omega \mid \omega \in X \text{ or } \omega \in Y\}$.
$X \cap Y$	– The set $\{\omega \mid \omega \in X \text{ and } \omega \in Y\}$.
$X \setminus Y$	– Set difference, that is, the set $\{\omega \mid \omega \in X, \omega \notin Y\}$.
\emptyset	– Empty set.
$ $	– Such that. For example, $\{x \mid Ax \leq b\}$ means the set of all x such that $Ax \leq b$ holds.
α^n	– A scalar raised to power n .
$(A)^n$	– A square matrix raised to power n .
A^T	– Transpose of the matrix A .
\approx	– Approximately equal to.
$\gg (\ll)$	– Much greater (less) than.
$\succ (\prec)$	– Lexicographically greater (less) than.
\leftarrow	– Store in the computer the value of the quantity on the right into the location where the quantity on the left is stored. For example, $x \leftarrow x + \alpha p$.
$O(v)$	– Implies a number $\leq kv$, where k , a fixed constant independent of the value of v , is meant to convey the notion that k is some small integer value less than 10 (or possibly less than 100) and not something ridiculous like $k = 10^{100}$.

-
- $\operatorname{argmin}_x f(x)$ – The value of x where $f(x)$ takes on its global minimum value.
- $\operatorname{argmin}_i \beta_i$ – The value of the least index i where β_i takes on its minimum value.
- LP – Linear program.
- $\operatorname{sign}(\alpha)$ – The sign of α . It is $+1$ if $\alpha \geq 0$ and -1 if $\alpha < 0$.

This page intentionally left blank

GEOMETRY OF LINEAR INEQUALITY SYSTEMS & THE SIMPLEX METHOD

In this chapter we study the geometry and properties of linear inequality systems and how they are related to the Simplex Method, which can be described as a movement along edges of convex polyhedral sets to obtain a global minimum of the objective function, generate a class of feasible solutions for which the objective $z \rightarrow -\infty$, or determine that the convex polyhedral set is infeasible.

1.1 CONVEXITY AND LINEAR INEQUALITY SYSTEMS

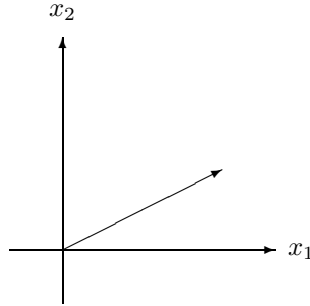
We denote $x \in \mathbb{R}^n$ to be either a point or a vector in n -dimensional space. As a vector it is a directed straight line segment formed by joining the origin to the point whose coordinates are (x_1, x_2, \dots, x_n) . This is shown in Figure 1-1 for a two-dimensional space.

Let α be a scalar. Then αx is the vector obtained by continuing, in the *same direction* as x , a distance $|\alpha||x|$ if $\alpha > 0$. On the other hand, if $\alpha < 0$ then αx is the vector obtained by continuing a distance $|\alpha||x|$ in the *opposite direction* of x .

1.1.1 AFFINE AND CONVEX COMBINATIONS

A *linear combination* of k vectors x^1, x^2, \dots, x^k is

$$x = \alpha_1 x^1 + \alpha_2 x^2 + \dots + \alpha_k x^k \quad (1.1)$$

Figure 1-1: Vector $(x_1, x_2)^T$

where α_j are real numbers. In *Linear Programming 1*, the set of all such linear combinations was defined as a *vector subspace* generated by x^1, x^2, \dots, x^k . If $\alpha_j \geq 0$ for all j , then (1.1) is called a *nonnegative linear combination*.

Definition: An *affine combination* of x^1, x^2, \dots, x^k is:

$$x = \alpha_1 x^1 + \alpha_2 x^2 + \dots + \alpha_k x^k, \quad (1.2)$$

where α_j are real numbers that satisfy $\sum_{j=1}^k \alpha_j = 1$.

Definition (Affine Hull): The set of all such affine combinations (1.2) is called the *affine hull* of x^1, x^2, \dots, x^k .

Definition: A *convex combination* of vectors x^1, x^2, \dots, x^k is:

$$x = \alpha_1 x^1 + \alpha_2 x^2 + \dots + \alpha_k x^k, \quad (1.3)$$

where $\alpha_j \geq 0$ are real numbers that satisfy $\sum_{j=1}^k \alpha_j = 1$.

Definition (Convexity Constraint): The condition

$$\alpha_1 + \alpha_2 + \dots + \alpha_k = 1, \quad \alpha_j \geq 0, \quad \text{for } j = 1, \dots, k \quad (1.4)$$

is called a *convexity constraint* on the α_j .

Definition (Convex Hull): The set of all such convex combinations (1.3) is called the *convex hull* of x^1, x^2, \dots, x^k .

▷ **Exercise 1.1** Show that the convex hull C of a subset of \Re^n has the property that every convex combination of a convex combination of points in C is also in C .

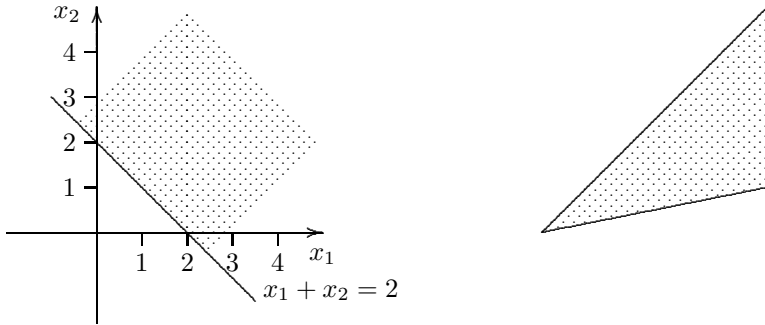


Figure 1-2: Examples of Two-Dimensional Convex Regions

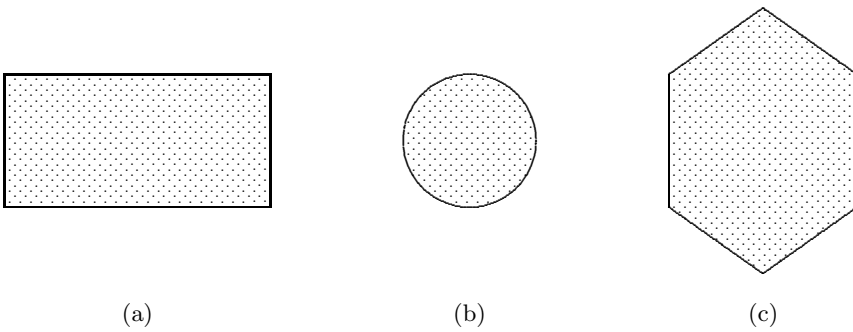


Figure 1-3: Additional Examples of Two-Dimensional Convex Regions

1.1.2 TWO-DIMENSIONAL CONVEX REGIONS

We begin by giving some simple examples of convex regions. The set of points (x_1, x_2) satisfying the relation

$$x_1 + x_2 \geq 2 \quad (1.5)$$

consists of a region in two-dimensional space on one side of the line (see Figure 1-2):

$$x_1 + x_2 = 2. \quad (1.6)$$

This is an example of a convex region, or, what is the same thing, a convex set of points. The region defined by the shaded area between two vectors indefinitely extended (see Figure 1-2) is also a convex set. Various other examples of convex regions in two dimensions can be constructed. For example, the region inside the rectangle (Figure 1-3a), the circle (Figure 1-3b), or the polygon (Figure 1-3c); but not the L-shaped region in Figure 1-5. Examples of convex regions in three dimensions are the volumes inside a cube and inside a sphere. Note that the latter regions

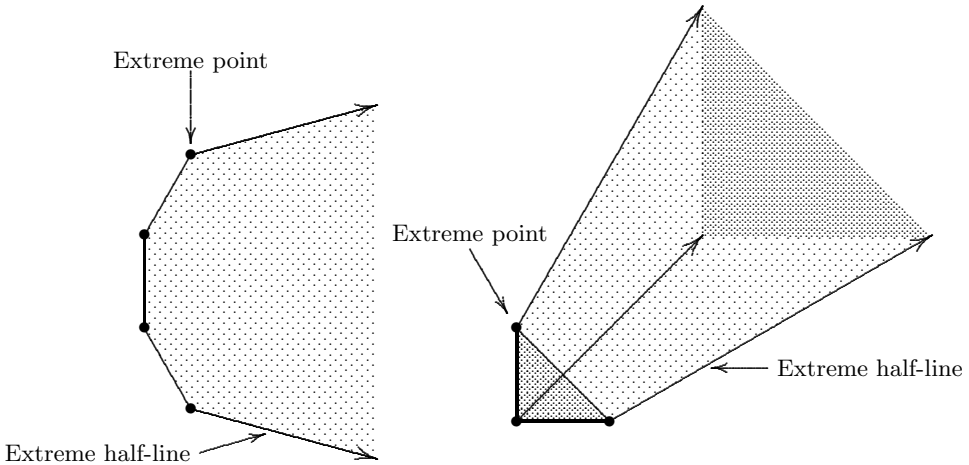


Figure 1-4: Example of an Unbounded Two-Dimensional Convex Set and an Unbounded Three-Dimensional Convex Set



Figure 1-5: Examples of Two-Dimensional Nonconvex Regions

may include or exclude the boundary; sometimes they may include or exclude parts of the boundary. The regions may be bounded in extent or unbounded. Examples of unbounded two-dimensional and three-dimensional convex sets are shown in Figure 1-4.

Definition (Convex Set): A set of points is called a *convex* set if all points on the straight line segment joining any two points in the set belong to the set.

- ▷ **Exercise 1.2** Show that in the case of a sphere any part of the boundary may be included or excluded without affecting the convexity of the region. Show that this is not necessarily true for a cube.

Clearly not all regions are convex; for example, neither of the two sets of points depicted by the shaded regions in Figure 1-5 is convex. The L-shaped region of

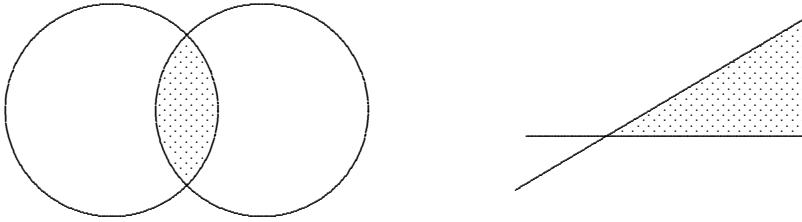


Figure 1-6: Points Common to Convex Sets

Figure 1-5 is not convex because it is possible to find two points, say p and q , in the set such that not all points on the line joining them belong to the set.

Definition: A *closed convex set* is one that includes its boundaries.

For example, a circle and its interior is a closed convex set; the interior of a circle is a convex set, but it is not closed.

Note that the region common to the two circles in Figure 1-6 is convex, as is the set of points belonging to the intersection of two or more regions defined by linear inequalities (see Figure 1-6).

▷ **Exercise 1.3** Prove that the set of points common to two or more convex sets is convex.

1.1.3 LINE SEGMENTS, RAYS, AND HALF LINES

Definition: The *line segment* joining two points, p and q , with coordinates (p_1, p_2, \dots, p_n) and (q_1, q_2, \dots, q_n) , respectively, in n -dimensional space is all points x whose coordinates are

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \lambda \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{pmatrix} + (1 - \lambda) \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{pmatrix} \quad (1.7)$$

where λ is a parameter such that $0 \leq \lambda \leq 1$. Clearly all the convex combinations of two points is a line segment joining them.

Example 1.1 (Line Segment) Consider the points $p = (p_1, p_2)$ and $q = (q_1, q_2)$ in two-dimensional space: $p = (1, 2)$ and $q = (5, 4)$. The line segment joining p and q is displayed in Figure 1-7. Next consider the point x , with coordinates (x_1, x_2) . By definition, if x is to be on the line segment joining p and q , then its components (x_1, x_2) satisfy:

$$\begin{aligned} x_1 &= \lambda p_1 + (1 - \lambda)q_1 = 1\lambda + 5(1 - \lambda) = -4\lambda + 5 \\ x_2 &= \lambda p_2 + (1 - \lambda)q_2 = 2\lambda + 4(1 - \lambda) = -2\lambda + 4. \end{aligned} \quad (1.8)$$

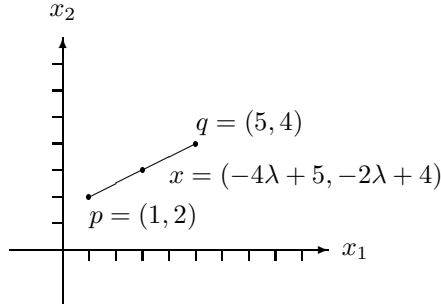


Figure 1-7: Example of a Line Segment

For example, let $\lambda = 1$, then $x_1 = 1$ and $x_2 = 2$ and the point x is point p . Likewise let $\lambda = 0$, then $x = q$. For other λ values ($0 < \lambda < 1$) we get all points between p and q . For example, when the parameter $\lambda = \frac{1}{2}$, the coordinates of x become $x_1 = 3$ and $x_2 = 3$, which is the point midway between p and q .

- ▷ **Exercise 1.4** Obtain the linear relationship between x_1 and x_2 by eliminating the parameter λ in Equation (1.8).

Definition (Ray and Half-Line): Let $x, p, q \in \mathfrak{R}^n$ and let $\theta \geq 0$. The *ray* generated by $q \in \mathfrak{R}^n$ is the set of points $\{x \mid x = \theta q\}$ as the scalar parameter θ varies from 0 to $+\infty$. A *half-line* anchored at $p \in \mathfrak{R}^n$ is the set of points $\{x \mid x = p + \theta q\}$ as the scalar parameter θ varies from 0 to $+\infty$. It is straightforward to see that every ray contains the origin and every half-line is a translation of a ray.

- ▷ **Exercise 1.5** Show the set of points generated by a ray is convex. Graph the half-line anchored at $p = (1, 1, 1)$ with the ray generated by $q = (1, 1, 1)$.

1.1.4 GENERAL CONVEX REGIONS

In linear programming we will be dealing with linear inequalities involving many variables, so it will not be possible to visualize the solution as a point in many dimensions. Accordingly we must be able to demonstrate algebraically whether or not certain sets are convex. The definition of a convex set requires that all points on a straight line segment joining any two points in the set belong to the set.

With the definition of a line segment, Equation (1.7), it is often easy to determine whether a given set is convex. For example, consider the region R defined by all points whose coordinates satisfy

$$x_1 + x_2 \geq 2. \quad (1.9)$$

To prove that this region is convex, let $p = (p_1, p_2)$ and $q = (q_1, q_2)$ be any two points in R . For p and q to be in R their respective coordinates must satisfy (1.9), whence

$$\begin{aligned} p_1 + p_2 &\geq 2 \\ q_1 + q_2 &\geq 2. \end{aligned} \tag{1.10}$$

Then the coordinates (x_1, x_2) of an arbitrary point x , on the segment joining p to q , are found by forming a convex combination of the coordinates of the two points, that is,

$$x = \lambda p + (1 - \lambda)q \tag{1.11}$$

where λ is the ratio of the distance xq to pq .

To prove convexity for (1.9) we wish to show that x lies in R , which means its coordinates should satisfy $x_1 + x_2 \geq 2$, or we need to show that

$$x_1 + x_2 = [\lambda p_1 + (1 - \lambda)q_1] + [\lambda p_2 + (1 - \lambda)q_2] \geq 2. \tag{1.12}$$

To prove this we multiply the first inequality of (1.10) by $\lambda \geq 0$ and the second, by $1 - \lambda \geq 0$ to obtain

$$\begin{aligned} \lambda p_1 + \lambda p_2 &\geq 2\lambda, \\ (1 - \lambda)q_1 + (1 - \lambda)q_2 &\geq 2(1 - \lambda). \end{aligned}$$

When added together, these two inequalities result in (1.12), which establishes algebraically the convexity of R .

1.1.5 HYPERPLANES AND HALF-SPACES

Definition (Hyperplane): In n dimensions, the set of points $x \in \mathfrak{R}^n$ whose coordinates (x_1, x_2, \dots, x_n) satisfy a linear equation

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b \tag{1.13}$$

where, for $j = 1, \dots, n$, at least one $a_j \neq 0$, is called a *hyperplane*. More precisely, it is an $(n - 1)$ -dimensional *hyperplane*.

Definition (Independence of Hyperplanes): A system of hyperplanes is *independent* if there exist no affine combination of the hyperplanes that results in $0x = 0$.

Definition (Dimension of Hyperplane Intersection): The intersection of p independent hyperplanes in \mathfrak{R}^n is called an $(n - p)$ -dimensional *hyperplane*.

A k -dimensional *hyperplane* H_k is called k -dimensional because we can choose any point x^o that lies on H_k and find k independent vectors v^1, v^2, \dots, v^k such

that $x^o + v^j$ lies in H_k ; moreover, it is easy to prove that all points $x \in H_k$ can be represented as

$$x^o + \sum_{j=1}^k \lambda_j v^j$$

for some chosen λ_j .

- ▷ **Exercise 1.6** In the hyperplane (1.13), assume that $a_1 \neq 0$. Choose a point x^o on the hyperplane and show how to find $n - 1$ independent vectors v^1, v^2, \dots, v^{n-1} such that the point $x^o + \sum_{j=1}^{n-1} \lambda_j v^j$ lies on the hyperplane for any choice of $\lambda_j, j = 1, \dots, n - 1$.
- ▷ **Exercise 1.7** Choose any point x^o on the k -dimensional hyperplane H_k and prove that there are k independent vectors v^1, v^2, \dots, v^k such that the point $x^o + \sum_{j=1}^k \lambda_j v^j$ lies on the hyperplane for any choice of $\lambda_j, j = 1, \dots, k$.
- ▷ **Exercise 1.8** Show that the lowest-dimensional intersection of hyperplanes containing a set of points in \mathfrak{R}^n is a subset of the vector subspace formed by the same set of points. Illustrate this in two dimensions.
- ▷ **Exercise 1.9** Show that a straight line is the lowest-dimensional hyperplane containing any two distinct points on it.

Definition (Half-Space): The set of points $x \in \mathfrak{R}^n$ whose coordinates satisfy a linear inequality such as

$$a_1 x_1 + a_2 x_2 + \dots + a_n x_n \leq b \quad (1.14)$$

is called a *half-space* or, to be precise, a *closed half-space* because we include the boundary. In two dimensions it is called a *half-plane*, and in one dimension it is a *half-line*.

1.1.6 CONVEXITY OF HALF SPACES AND HYPERPLANES

To prove the half-space defined by a linear inequality is convex, let p and q be any two points in the set, so that

$$a^T p \leq b \quad (1.15)$$

$$a^T q \leq b. \quad (1.16)$$

Let $0 \leq \lambda \leq 1$ be the value of the parameter associated with an arbitrary point x on the line segment joining p to q ; see (1.7). Multiplying (1.15) by $\lambda \geq 0$ and (1.16) by $(1 - \lambda) \geq 0$ and adding, one obtains

$$a^T \lambda p + a^T (1 - \lambda) q \leq b \quad (1.17)$$

whence, factoring a^T and substituting $x = \lambda p + (1 - \lambda)q$,

$$a^T x \leq b. \quad (1.18)$$

Hence, an arbitrary point x on the line segment joining any two points lies in the half-space, establishing convexity.

To prove that a hyperplane is convex, let (1.13) be written as

$$\begin{aligned} a^T x &\leq b \\ a^T x &\geq b. \end{aligned} \quad (1.19)$$

Each of these inequalities defines a half-space, and their intersection defines a hyperplane. Since a half-space is a convex set, then, by Exercise 1.3 on Page 5, a hyperplane is also a convex set. An n -dimensional space may contain many such convex sets. By Exercise 1.3, the common intersection of two or more of these convex sets is a convex set.

1.1.7 CONVEXITY OF THE SET OF FEASIBLE SOLUTIONS OF AN LP

THEOREM 1.1 (Set of Feasible Points for an LP is Convex) *The set of points corresponding to feasible (or optimal feasible) solutions of the general linear programming problem constitutes a convex set.*

▷ **Exercise 1.10** Prove Theorem 1.1.

Thus, if $p = (p_1, p_2, \dots, p_n, z_p)$ is a feasible solution and $q = (q_1, q_2, \dots, q_n, z_q)$ is another, the weighted linear combination of these two feasible solutions,

$$x = \lambda p + (1 - \lambda)q, \quad (1.20)$$

where λ is a constant, $0 \leq \lambda \leq 1$, is also a feasible solution. Moreover, assigning a fixed value for z , say $z = z_0$, the set of points satisfying $c^T x = z$ and $Ax = b$, $x \geq 0$ is also a convex set. In particular, setting $z_0 = \min z$, it is clear that *the set of minimal feasible solutions is also a convex set.*

1.1.8 CONVEX POLYHEDRONS, POLYTOPES, AND CONES

Definition (Convex Polyhedron): A convex polyhedron is the set of points common to one or more half-spaces.

Definition (Convex Polytope): A convex polyhedron that is bounded is called a convex polytope.

Definition (Convex Polygon): A convex polygon is a two-dimensional convex polytope.

▷ **Exercise 1.11** Prove that for the intersection of a set of half-spaces in \mathfrak{R}^n to be bounded it is necessary (but not sufficient) that their number must be $n+1$ or more. For the feasible set for a linear program $Ax = b$, $x \geq 0$, $A \in \mathfrak{R}^{m \times n}$ to be bounded, must m bear a special relation to n ?

▷ **Exercise 1.12** Show that the set of optimal feasible solutions to a linear program is a convex polyhedron.

▷ **Exercise 1.13** Show that the convex combination of the set of optimal feasible solutions to a linear program is a convex polytope.

Definition (Cone): A subset $K \subset \mathfrak{R}^n$ is a cone if and only if $x \in K$ implies that $\alpha x \in K$ for all $\alpha \geq 0$. In other words, K is a cone if and only if the ray generated by any point in the cone lies entirely in the cone. A convex cone is a cone that is also a convex set.

▷ **Exercise 1.14** Construct an example to show that a cone in general need not be convex.

▷ **Exercise 1.15** Prove that a cone is convex if and only if it contains every nonnegative linear combination of any finite number of points in it.

▷ **Exercise 1.16** Show that a convex cone is formed by the set C of all points $b = (b_1, b_2, \dots, b_m)$ generated by all choices of $x \geq 0$ in the expression

$$Ax = b.$$

Definition (Convex Polyhedral Cone): A convex polyhedral cone is a convex cone K that is the intersection of a finite number of half-spaces, each of which contains the origin.

Definition (Simplicial Cone): A simplicial cone of dimension m is defined to be:

$$\mathcal{S} = \left\{ b \mid b = \sum_{i=1}^m \alpha_i p_i; \alpha_i \geq 0 \text{ for } i = 1, \dots, m \right\},$$

where $\{p_1, p_2, \dots, p_m\}$ are linearly independent.

▷ **Exercise 1.17** Show that $\{ b \mid b = \sum_{i=1}^{m+1} \alpha_i p_i; \alpha_i \geq 0 \text{ for } i = 1, \dots, m+1 \}$ in \mathfrak{R}^m is not a simplicial cone.

1.1.9 SEPARATING HYPERPLANE

An important property of convex regions is the concept of separating hyperplanes and supporting hyperplanes.

Definition (Separating Hyperplane): Let \mathcal{C}_1 and \mathcal{C}_2 be two convex sets in \mathfrak{R}^n . A hyperplane

$$\alpha^T x = \beta$$

with

$$\begin{aligned} \alpha^T u &> \beta && \text{for all } u \in \mathcal{C}_1 \\ \alpha^T v &\leq \beta && \text{for all } v \in \mathcal{C}_2, \end{aligned}$$

is called a *separating hyperplane*. It separates the convex set \mathcal{C}_1 from the convex set \mathcal{C}_2 .

Definition (Supporting Hyperplane): Given a convex set \mathcal{C} in \mathfrak{R}^n , a hyperplane $\alpha^T x = \beta$, is called a *supporting hyperplane* if all $x \in \mathcal{C}$ satisfy $\alpha^T x \geq \beta$ (or if all $x \in \mathcal{C}$ satisfy $\alpha^T x \leq \beta$), with equality holding for at least one $x \in \mathcal{C}$.

LEMMA 1.2 (Separating Hyperplane for Half-Spaces) *Let*

$$\mathcal{C}_1 = \left\{ x \mid \sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = 1, \dots, k \right\} \quad (1.21)$$

$$\mathcal{C}_2 = \left\{ x \mid \sum_{j=1}^n a_{ij} x_j \geq b_i, \quad i = k+1, \dots, m \right\} \quad (1.22)$$

be disjoint convex sets. Then there exists a separating hyperplane that separates \mathcal{C}_1 from \mathcal{C}_2 .

Proof. By the hypothesis the combined set of inequalities that define \mathcal{C}_1 and \mathcal{C}_2 are infeasible. By the Infeasibility Theorem (see *Linear Programming 1*), there exist nonnegative multipliers $\pi_1, \pi_2, \dots, \pi_k$ on the inequalities in \mathcal{C}_1 and nonnegative multipliers $\pi_{k+1}, \pi_{k+2}, \dots, \pi_m$ on the inequalities in \mathcal{C}_2 so that

$$\sum_{i=1}^k \pi_i a_{ij} + \sum_{i=k+1}^m \pi_i a_{ij} = 0 \quad \text{for } j = 1, \dots, n \quad (1.23)$$

implying

$$\sum_{i=1}^k \pi_i a_{ij} = - \sum_{i=k+1}^m \pi_i a_{ij} \quad \text{for } j = 1, \dots, n \quad (1.24)$$

and

$$\sum_{i=1}^k \pi_i b_i + \sum_{i=k+1}^m \pi_i b_i > 0 \quad (1.25)$$

implying

$$\sum_{i=1}^k \pi_i b_i + \sum_{i=k+1}^m \pi_i b_i = \delta > 0. \quad (1.26)$$

Choosing any $\epsilon > 0$ and $\eta \geq 0$ such that $\delta = \epsilon + \eta$, we rewrite this last equation as

$$\sum_{i=1}^k \pi_i b_i - \epsilon = - \sum_{i=k+1}^m \pi_i b_i + \eta. \quad (1.27)$$

Note that multiplying the inequalities in \mathcal{C}_1 by $\pi_i \geq 0, i = 1, \dots, k$, and summing we obtain

$$\sum_{i=1}^k \sum_{j=1}^n \pi_i a_{ij} x_j \geq \sum_{i=1}^k \pi_i b_i. \quad (1.28)$$

Clearly, subtracting $\epsilon > 0$ from the right-hand side and rearranging the summations results in the strict inequality

$$\sum_{j=1}^n \left(\sum_{i=1}^k \pi_i a_{ij} \right) x_j > \sum_{i=1}^k \pi_i b_i - \epsilon \quad (1.29)$$

which holds for all $x \in \mathcal{C}_1$. Similarly, multiplying the inequalities in \mathcal{C}_2 by $\pi_i, i = k + 1, \dots, m$, and summing we obtain

$$\sum_{i=k+1}^m \sum_{j=1}^n \pi_i a_{ij} x_j \geq \sum_{i=k+1}^m \pi_i b_i. \quad (1.30)$$

Clearly, multiplying by -1 and adding $\eta \geq 0$ on the right hand side, and rearranging the summations, results in the strict inequality

$$\sum_{j=1}^n \left(- \sum_{i=k+1}^m \pi_i a_{ij} \right) x_j \leq \left(- \sum_{i=k+1}^m \pi_i b_i \right) + \eta. \quad (1.31)$$

Substituting (1.24) and (1.27) into (1.31) we obtain

$$\sum_{j=1}^n \left(\sum_{i=1}^k \pi_i a_{ij} \right) x_j \leq \sum_{i=1}^k \pi_i b_i - \epsilon \quad (1.32)$$

which holds for all $x \in \mathcal{C}_2$. Hence we have constructed a separating hyperplane defined by (1.29) and (1.32) which is:

$$\sum_{j=1}^n \left(\sum_{i=1}^k \pi_i a_{ij} \right) x_j = \sum_{i=1}^k \pi_i b_i - \epsilon \quad (1.33)$$

■

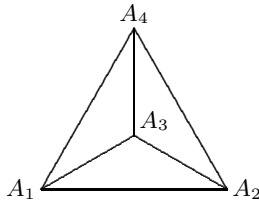


Figure 1-8: A Three-Dimensional Simplex

▷ **Exercise 1.18** Show that (1.33) is a separating hyperplane for any choice of $0 < \epsilon < \delta$ that *strictly* separates \mathcal{C}_1 from \mathcal{C}_2 . If $\epsilon = 0$ it is a supporting hyperplane for \mathcal{C}_1 but not for \mathcal{C}_2 , and if $\epsilon = \delta$ it is a supporting hyperplane for \mathcal{C}_2 but not for \mathcal{C}_1 .

THEOREM 1.3 (Separating Hyperplane for General Convex Sets) *If two convex sets \mathcal{C}_1 and \mathcal{C}_2 in \mathbb{R}^n are disjoint, there exists a separating hyperplane that separates \mathcal{C}_1 from \mathcal{C}_2 .*

Proof. Find a point $x^1 \in \mathcal{C}_1$ that is closest to the points in \mathcal{C}_2 and find a point $x^2 \in \mathcal{C}_2$ that is closest to the points in \mathcal{C}_1 . Because the convex sets are disjoint we can pick any point x^o on the closed line segment of positive length joining x^1 to x^2 and construct a hyperplane that passes through x^o perpendicular to the line joining x^1 to x^2 . This hyperplane separates \mathcal{C}_1 from \mathcal{C}_2 . ■

1.2 SIMPLEX DEFINED

There is a close connection between the Simplex Method and the the simplest higher-dimensional polyhedral set, the simplex.

Definition (m-Dimensional Simplex): In higher dimensions, say m , the convex hull of $m + 1$ points in general position (see definition below) is called an *m-dimensional simplex*.

Thus

- a zero-dimensional simplex is a point;
- a one-dimensional simplex is a line segment;
- a two-dimensional simplex is a triangle and its interior;
- a three-dimensional simplex is a tetrahedron and its interior. (See Figure 1-8).

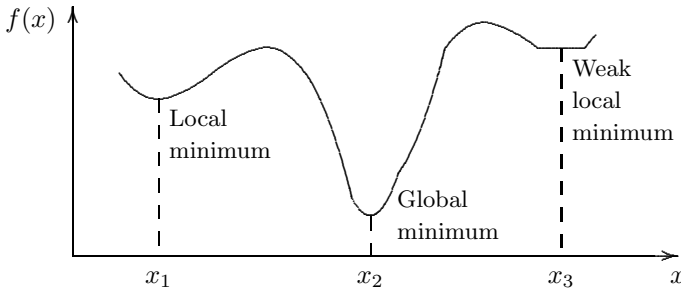


Figure 1-9: Local and Global Minima

Definition (General Position): Let $A_j = (a_{1j}, a_{2j}, \dots, a_{mj})$ be the coordinates of a point A_j in m -dimensional space. Algebraically a set of $m+1$ points $[A_1, A_2, \dots, A_{m+1}]$ of points in m dimensions is said to be in *general position* if the determinant of their coordinates and a row of ones, as in (1.34), is nonvanishing,

$$\begin{vmatrix} 1 & 1 & \dots & 1 \\ a_{11} & a_{12} & \dots & a_{1,m+1} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{m,m+1} \end{vmatrix} \neq 0. \quad (1.34)$$

Definition (Algebraic Definition of an m -Dimensional Simplex): The set of all points,

$$x = \lambda_1 A_1 + \lambda_2 A_2 + \dots + \lambda_{m+1} A_{m+1}, \quad (1.35)$$

generated by all choices of λ such that $\sum_{j=1}^{m+1} \lambda_j = 1$, $\lambda_j \geq 0$ is defined to be an m -dimensional simplex if the determinant (1.34) is nonvanishing.

Definition (Vertices of a Simplex): The points $x = A_j$ in (1.35) are called *vertices* or *extreme points* of the simplex.

1.3 GLOBAL MINIMUM, EXTREME POINTS, AND EDGES

In the calculus we learned that if a function $f(x)$, defined over an interval, has a continuous derivative $f'(x)$ and attains a minimum (or maximum) at a point x_0 within the interval, then the derivative $f'(x_0) = 0$. However, having a derivative $f'(\bar{x}) = 0$ does not necessarily imply that a minimum of $f(x)$ is attained at $f(\bar{x})$. See, for example, the point $x = x_1$ in Figure 1-9, where $f(x)$ is minimum only in

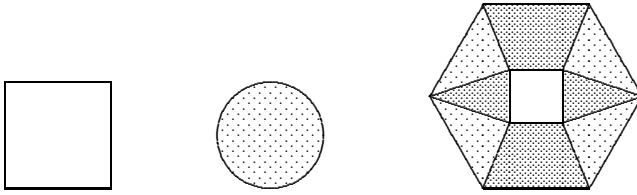


Figure 1-10: Extreme Points

the neighborhood of x_1 ; this is called a *local* minimum. However, it will also be noted that there is another local minimum at $x = x_2$, where $f(x)$ attains its lowest value; this is called a *global* minimum. The point $x = x_3$ is also a local minimum, and so also are its neighboring points.

LEMMA 1.4 (A Local Minimum of an LP is Global) *Any solution to a linear programming problem that is a local minimum solution is also a global minimum solution.*

Proof. To see this, let $p = (p_1, p_2, \dots, p_n, z_p)$ be a local minimum solution and assume on the contrary that it is not a global minimum solution, so that there is another solution $q = (q_1, q_2, \dots, q_n, z_q)$ with $z_q < z_p$. Then any point $x = (x_1, x_2, \dots, x_n, z) = (1 - \lambda)p + \lambda q$, $0 \leq \lambda \leq 1$, on the line segment joining these p and q points would be a feasible solution and its objective $z = (1 - \lambda)z_p + \lambda z_q$. In this case the value of z decreases from z_p to z_q as λ varies from 0 to 1. Thus all such feasible points (including those in the *neighborhood* of p) would have z values less than z_p contrary to the hypothesis that p is a local minimum and not a global minimum. This means that z_p is a global minimum. ■

- ▷ **Exercise 1.19** Prove that the point x which yields a global minimum for an LP is not necessarily unique. Illustrate with an example.
- ▷ **Exercise 1.20** Suppose p and q are both local minima. Prove that for a linear program all points on the line segment joining p and q are global minima.

Definition (Extreme Point or Vertex): Any point x in a convex set C that is not a midpoint of the line segment joining two other distinct points in C is by definition an *extreme point* or *vertex* of the convex set.

Example 1.2 (Degenerate and Nondegenerate Extreme Points) In Figure 1-10, the corners of the polygonal region bounded by a square are extreme points, as is every point on the circumference of a circle bounding a disk. The points where three or more facets of a diamond (assumed to be flat on the bottom) come together are extreme points

of the diamond. Notice also that the corners of the square facet on top of the diamond are degenerate extreme points because more than three hyperplanes intersect to generate each corner. Further note that none of the half-spaces defining these four degenerate extreme points can be dropped without changing the polyhedral set, implying that there are no redundant constraints.

▷ **Exercise 1.21** In Figure 1-10 find degenerate extreme points other than those described in Example 1.2. Are there any redundant constraints associated with these degenerate extreme points? Why?

▷ **Exercise 1.22** Given the coordinates (x_1, x_2) of the vertices of a simplex in \mathbb{R}^2 :

$$P_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \text{and} \quad P_3 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

1. Write down the relations defining the convex hull of P_1 , P_2 , and P_3 .
2. Use the FME process (see *Linear Programming 1*) to define the feasible region in terms of x_1 and x_2 alone.
3. Plot the points P_1 , P_2 , and P_3 and verify your result.

▷ **Exercise 1.23** In Figure 1-10, let the coordinates of the extreme points of the square facet be $(1, 1, 1)$, $(1, -1, 1)$, $(-1, 1, 1)$, and $(-1, -1, 1)$ and the coordinates of the remaining extreme points of the diamond be $(-1.5, 2, 0)$, $(1.5, 2, 0)$, $(2, 0, 0)$, $(1.5, -2, 0)$, $(-1.5, -2, 0)$, and $(-2, 0, 0)$. Determine algebraically the hyperplanes that represent the diamond by reducing the system representing the convex hull of these extreme points to a system of inequalities in nonnegative variables. Show that the basic solutions corresponding to the extreme points where four hyperplanes intersect are degenerate.

Definition (Basic Solution): Consider the canonical system

$$Ix_B + \bar{A}x_N = \bar{b}. \tag{1.36}$$

The special solution obtained by setting the independent variables x_N equal to zero and solving for the dependent variables x_B is called a *basic solution*.

THEOREM 1.5 (Basic Feasible Solution is an Extreme Point) *A basic feasible solution to a linear program $Ax = b$, $x \geq 0$, $c^T x = \min$ corresponds to an extreme point in the convex set of feasible solutions to the linear program.*

Proof. We assume that the rank of the system is m (see Exercise 1.25). It is easy to show that a basic feasible solution corresponds to an extreme point. By relabeling the indices, let $x^o = (\bar{b}_1, \bar{b}_2, \dots, \bar{b}_m, 0, \dots, 0)$ be a basic feasible solution with respect to basic variables x_1, x_2, \dots, x_m . By definition the columns of A corresponding to these variables are independent. Suppose, on the contrary, that

x^o is not an extreme point. Then, by definition, it is the average of two other distinct feasible solutions $p = (p_1, p_2, \dots, p_m, \dots, p_n) \geq 0$ and $q = (q_1, q_2, \dots, q_m, \dots, q_n) \geq 0$. It follows for all j corresponding to the nonbasic variables $j = m + 1, \dots, n$ that

$$x_j^o = \frac{1}{2}(p_j + q_j) = 0, \quad (1.37)$$

where $p_j \geq 0$ and $q_j \geq 0$. But this is possible only if $p_j = q_j = 0$ for $j = m + 1, \dots, n$. Thus p , q , and x^o have the same values (namely zero) for their components corresponding to nonbasics. Since the values of the basic variables are uniquely determined by the values of the nonbasics (independent variables), we must have $p = q = x^o$. This proves that x^o cannot be the average of two solutions p and q distinct from x^o and hence, by definition, x^o is an extreme point. ■

COROLLARY 1.6 (Extreme Point is a Basic Feasible Solution) *Each extreme point corresponds to one or more basic feasible solutions. If one of the basic feasible solutions is nondegenerate an extreme point corresponds to it uniquely.*

Proof. We again assume that the rank of the system is m (see Exercise 1.25). By relabeling the indices, let $x = (x_1^o, \dots, x_k^o, 0, \dots, 0) \geq 0$ be an extreme point, where $x_j^o > 0$ for $1 \leq j \leq k$. Then, the first k columns must be linearly independent, because, if not, for some column k ,

$$A_{\bullet k} = \sum_{j=1}^{k-1} y_j A_{\bullet j}.$$

This would imply that the class of solutions

$$x(\theta) = (x_1^o - \theta y_1, x_2^o - \theta y_2, \dots, x_{k-1}^o - \theta y_{k-1}, x_k^o + \theta, 0, \dots, 0)^T \quad (1.38)$$

are feasible solutions for some range $-\theta^o \leq \theta \leq \theta^o$, where $\theta^o > 0$. Therefore,

$$x^o = \frac{1}{2}x(-\theta^o) + \frac{1}{2}x(\theta^o)$$

is a midpoint of two other feasible solutions, a contradiction if x^o is an extreme point.

Since the maximum number of independent columns is m , this implies that $k \leq m$. If $k = m$, then, by definition of a basis, x^o is a basic feasible solution and the only one corresponding to the extreme point. However, if $k < m$, we can augment the independent columns $j = 1, \dots, k$ by $m - k$ additional independent columns with indices $j_{k+1}, j_{k+2}, \dots, j_m$ that can be used to construct the feasible basis. (such additional columns exist because we assumed the rank of the system is m). The corresponding (degenerate) basic feasible solution is obtained by letting $x_{j_{k+1}} = x_{j_{k+2}} = \dots = x_{j_m} = 0$. In general, this augmentation by $m - k$ independent columns is not unique. ■

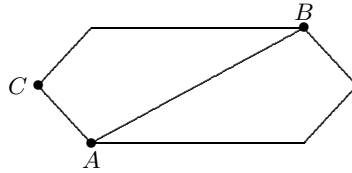


Figure 1-11: Joining Extreme Points

- ▷ **Exercise 1.24** Show in the preceding proof that the class of solutions $x(\theta)$ given by Equation (1.38) are feasible solutions for some range $-\epsilon \leq \theta \leq \epsilon$, where $\epsilon > 0$.
- ▷ **Exercise 1.25** Modify the proofs of Theorem 1.5 and of Corollary 1.6 if the rank of the system is $r < m$.

Definition (Edge): An *edge* of a convex polyhedron \mathcal{C} is the straight line segment joining two extreme points such that no point on the segment is the midpoint of two other points in \mathcal{C} not on the segment.

Definition (Neighbors): Two distinct extreme points having an edge in common are said to be *neighbors* or *adjacent* to each other.

- ▷ **Exercise 1.26** Why is the segment joining two extreme points AB in Figure 1-11 not an edge, but AC is an edge?

THEOREM 1.7 (Movements Along Edges) *The class of feasible solutions generated by increasing the value θ of a nonbasic variable and adjusting the values of the basic variables, while maintaining feasibility, corresponds to a movement along an edge of the convex set.*

Proof. The case where the increase in θ is unbounded is left as an exercise (see Exercise 1.28). Let

$$p = (\bar{b}_1, \bar{b}_2, \dots, \bar{b}_m; 0, 0, \dots, 0) \geq 0$$

be the basic feasible solution for iteration t and relabel the indices so that $s = m + 1$ is the index of the incoming variable. Then we know that if $x_{m+1} = \theta$, the change in the feasible solution is given by

$$x = (\bar{b}_1 - \theta v_1, \bar{b}_2 - \theta v_2, \dots, \bar{b}_m - \theta v_m, \theta, 0, 0, \dots, 0) \quad (1.39)$$

where $v = B^{-1}A_{\bullet m+1} = \bar{A}_{\bullet m+1}$ is the representation of the incoming column $A_{\bullet m+1}$ in terms of the basis. Let $\hat{\theta}$ be the maximum value of θ such that x in

(1.39) is nonnegative; by assumption $\hat{\theta}$ is finite. (Note that if $\hat{\theta} = 0$; the problem is degenerate and there is no change in the extreme point p but there is a change in the set of basic variables that define the extreme point p .) Assuming nondegeneracy, we let $\hat{\theta} = \hat{b}_{m+1} > 0$ be the maximum value of θ . Relabel the indices so that $r = 1$ is the index of the outgoing variable. Then it follows that $\bar{b}_1 - \hat{\theta}v_1 = 0$ (or $\hat{\theta} = \bar{b}_1 - v_1$). Denote the new basic feasible solution by

$$q = (0, \hat{b}_2, \dots, \hat{b}_m, \hat{b}_{m+1}, 0, 0, \dots, 0),$$

where

$$\hat{b}_i = \bar{b}_i - \hat{\theta}v_i \quad \text{for } i = 2, \dots, m.$$

It is easy to see that x given by (1.39) satisfies

$$x = (1 - \lambda)p + \lambda q, \quad 0 \leq \lambda \leq 1, \quad (1.40)$$

where $\lambda = \theta/\hat{b}_{m+1} = \theta/\hat{\theta} \geq 0$ and $(1 - \lambda) = (\hat{b}_{m+1} - \theta)/\hat{b}_{m+1} = (\hat{\theta} - \theta)/\hat{\theta} \geq 0$ (see Exercise 1.27). Thus, increasing x_{m+1} while keeping $x_{m+2} = x_{m+3} = \dots = x_n = 0$ is the same as increasing λ from 0 to 1, which, in turn, is the same as moving along the line segment joining p and q .

What is left to be shown is that the line segment joining p and q is actually an edge joining two adjacent extreme points. Clearly $0 \leq x_{m+1} \leq \hat{b}_{m+1}$, because this is the range of values of the incoming variable that keeps all other variables nonnegative. Furthermore, the $(m + 1)$ th component of any point u on the line segment joining p and q satisfies $0 \leq u_{m+1} \leq \hat{b}_{m+1}$ and $u_{m+2} = u_{m+3} = \dots = u_n = 0$. Conversely let y be any feasible point with the property that $y_{m+2} = y_{m+3} = \dots = y_n = 0$, then y must be on the line segment joining p and q because, letting $y_{m+1} = \theta > 0$, the value of $y = x$ is given by (1.39). Thus, $\theta \leq \hat{b}_{m+1}$ must be true for feasibility. Suppose next that u on the line segment joining p and q is the midpoint of some two other feasible points p' and q' . Thus

$$u = \frac{1}{2}p' + \frac{1}{2}q'.$$

Note $p'_j \geq 0$ and $q'_j \geq 0$ for all components j because p' and q' are feasible points. Clearly $p'_j = q'_j = 0$ for $j = m + 2, m + 3, \dots, n$ because $p'_j + q'_j = u_j = 0$ for all $j = m + 2, m + 3, \dots, n$ and $p'_j \geq 0$ and $q'_j \geq 0$. But as we have just shown such a $y = p'$ lies on the line segment joining p and q . Similarly q' also lies on the line segment joining p and q . By definition, an edge is a line segment having the property that every point on the line segment, if it is the midpoint of two other feasible points, has the property that these two points must also lie on the line segment. ■

▷ **Exercise 1.27** Verify that x given by Equation (1.39) satisfies Equation (1.40). Prove conversely that if x lies on the line segment joining p and q then it satisfies (1.39) for some choice of θ .

- ▷ **Exercise 1.28** Prove Theorem 1.7 for the case where the increase in θ is unbounded so that the movement is along an edge that is an extreme half-line.

1.4 THE SIMPLEX METHOD VIEWED AS THE STEEPEST DESCENT ALONG EDGES

It can be shown in general that the Simplex Algorithm can be viewed as a steepest descent “gradient” technique in which a “gradient direction” is defined in the space of nonbasic variables, say $x_{m+1}, x_{m+2}, \dots, x_n$. This gradient direction differs from the one usually used in the Calculus. Translating the origin to some trial solution point, the *usual* steepest gradient direction is defined by finding the limiting direction as $\rho \rightarrow 0$ from this origin to a point on the *spherical* surface

$$x_{m+1}^2 + x_{m+2}^2 + \dots + x_n^2 = \rho^2, \quad x_j \geq 0, \quad (1.41)$$

where some function $z = f(x)$ is minimized. In contradistinction, the Simplex Algorithm’s steepest gradient direction is found using a *planar* surface (instead of a spherical surface)

$$x_{m+1} + x_{m+2} + \dots + x_n = \rho, \quad x_j \geq 0. \quad (1.42)$$

In other words, in defining the gradient, the usual (Euclidean) distance (1.41) from the origin (located at some trial solution point) is replaced by (1.42), one based on the sum of the nonnegative values of the coordinates of the independent (nonbasic) variables.

THE SPECIAL CASE OF $n = m + 2$

Consider a linear programming problem with $n = m + 2$ that has a basic feasible solution with respect to some m basic variables, say x_3, x_4, \dots, x_{m+2} . The canonical form with respect to these variables is

$$\begin{array}{rcl} \bar{c}_1 x_1 + \bar{c}_2 x_2 & & = z - \bar{z}_0 \\ \bar{a}_{11} x_1 + \bar{a}_{12} x_2 + x_3 & & = \bar{b}_1 \\ \bar{a}_{21} x_1 + \bar{a}_{22} x_2 & + x_4 & = \bar{b}_2 \\ \vdots & \vdots & \vdots \\ \bar{a}_{m1} x_1 + \bar{a}_{m2} x_2 + \dots & + x_{m+2} & = \bar{b}_m \end{array} \quad (1.43)$$

with $\bar{b}_i \geq 0$ and where the problem is to find $x_j \geq 0$ and $\min z$ satisfying (1.43). The convex set of feasible solutions satisfying $(x_1, x_2, \dots, x_n) \geq 0$ will be denoted

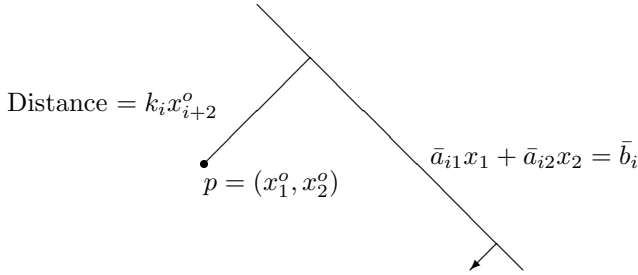


Figure 1-13: Geometric Picture of the Distance of a Point to a Boundary

given by (1.45) for $i = 1, \dots, m$,

$$\text{distance} = d_i = \frac{\bar{b}_i - \bar{a}_{i1}x_1^0 - \bar{a}_{i2}x_2^0}{[\bar{a}_{i1}^2 + \bar{a}_{i2}^2]^{\frac{1}{2}}} = k_i x_{i+2}^0, \quad (1.45)$$

where $k_i = (\bar{a}_{i1}^2 + \bar{a}_{i2}^2)^{-\frac{1}{2}}$. If the point (x_1^0, x_2^0) satisfies the inequality, then the geometric picture of the distance of a point from the boundary is shown in Figure 1-13.

If the slack variables x_{i+2} are replaced by $y_i = k_i x_{i+2}$ for $i = 1, \dots, m$, and the coordinates of a point P are the values of the independent variables, then *the value of the i th basic variable is just the distance from the point P to the corresponding i th constraint.*

Every basic solution to (1.43) has at least two $x_j = 0$; hence the corresponding P is at the same time a point in \mathcal{K} and at zero distance to two distinct boundary lines of \mathcal{K} . It is intuitively evident (and we show this rigorously below) that such a P is a vertex of \mathcal{K} . In particular, the basic feasible solution with respect to the canonical form (1.43) is associated with the point $(x_1^0 = 0, x_2^0 = 0)$ in Figure 1-12, hence the origin is always in the convex \mathcal{K} .

LEMMA 1.8 (Extreme Points) *Associated with every extreme point in the convex set of feasible solutions to the original linear program in standard form, (1.43), is an extreme point of \mathcal{K} and conversely; where \mathcal{K} is formed by dropping the basic feasible variables in the canonical form (1.43) and replacing equations by inequalities to get (1.44).*

Proof. Let $P = (x_1^o, x_2^o)$ and $Q = (x_1', x_2')$ be any two points in \mathcal{K} , and let the corresponding feasible solutions satisfying (1.43) be $p = (x_1^o, x_2^o, \dots, x_n^o)$ and $q = (x_1', x_2', \dots, x_n')$, which, as we saw in Theorem 1.1, lie in a convex set \mathcal{C} . It is easy to see that any point $\lambda P + (1 - \lambda)Q$ on the line joining P to Q corresponds to a point $\lambda p + (1 - \lambda)q$ that satisfies (1.43), and conversely. Hence line segments in the convex set \mathcal{C} of solutions satisfying (1.43) correspond to line segments in \mathcal{K} , and in particular the midpoint of a segment in \mathcal{C} corresponds to the midpoint in \mathcal{K} .

and conversely. It follows that nonextreme points must correspond to each other and it follows that extreme points (basic feasible solutions) to (1.43) correspond to extreme points of \mathcal{K} and conversely. ■

LEMMA 1.9 (Movement from One Vertex to the Next) *The movement along the edge corresponding to the class of feasible solutions generated by increasing a nonbasic variable and adjusting the values of the basic variables in the shift from one basic solution to the next, corresponds to a movement around the boundary of \mathcal{K} from one vertex to the next.*

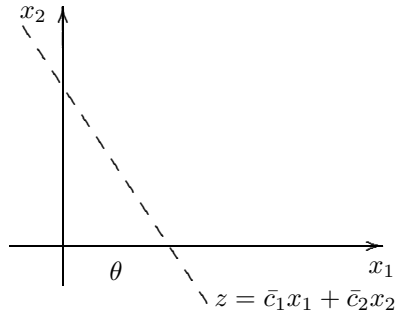
Proof. Let p and q be successive distinct extreme points corresponding to basic feasible solutions obtained by the Simplex Method under nondegeneracy, so that the line segment joining p to q is an edge in \mathcal{C} . If the corresponding vertices P and Q in \mathcal{K} were not neighbors, there would be a point X on the segment joining P to Q that would be the midpoint of two points P' and Q' in \mathcal{K} , but not on the segment. We shall show, however, that P' and Q' must lie on the line joining P to Q . We have shown that x , corresponding to X must be the midpoint of p' and q' corresponding to P' and Q' . However, x must also be on the line joining p to q since X was on the line joining P to Q . It follows, since the segment pq is an edge (see Theorem 1.7), that p' and q' must both be on this edge and hence their corresponding points P' and Q' must lie on the line joining P to Q . This shows that edges in the convex set of feasible solutions to (1.43), correspond to edges in Figure 1-12. ■

Thus, in the nondegenerate case, the Simplex Method proceeds from one vertex to the next in the convex region \mathcal{K} in the space of some initial set of nonbasic variables. Starting with the vertex at the origin and moving successively from one neighboring vertex to another, each step decreases the value of \bar{c}_0 until a minimum value for \bar{c}_0 is obtained, as shown by the arrows in Figure 1-12.

- ▷ **Exercise 1.29** Modify Figure 1-12 to illustrate the degenerate case and modify the preceding discussion to correspond to your drawing.

THE GENERAL CASE

Although our remarks have been restricted to the case of $n = m + 2$ for simplicity, they hold equally well for $n = m + k$. In the general case, the values of $k = n - m$ of any set of nonbasic variables become the coordinates of a point in k dimensions. In this geometry the convex set \mathcal{K} of feasible solutions is defined as before by a set of m inequalities formed by dropping the basic variables in the canonical form and by k inequalities $x_j \geq 0$ where x_j are the nonbasic variables. Each basic feasible solution corresponds to a vertex of \mathcal{K} . In the general (nondegenerate) situation, there are $n - m$ edges leading from each vertex to $n - m$ neighboring vertices; these correspond to the $n - m$ basic solutions obtained by introducing one of the $n - m$ nonbasic variables in place of one of the basic variables.

Figure 1-14: Movement of θ

STEEPEST DESCENT ALONG EDGES

The simplex criterion of choosing $\bar{c}_s = \min \bar{c}_j < 0$ followed by an increase in x_s corresponds to a movement along that edge of the convex set that induces the greatest decrease in z per unit change in the variable introduced.

For example, for $n = m + 2$ (see Figure 1-14), if $\bar{c}_1 < \bar{c}_2$ then any movement for a distance θ along the x_1 -axis produces a greater decrease in z than an equal movement of θ along the x_2 -axis and therefore the steepest descent direction using as boundary the planar surface (line) $x_1 + x_2 = \rho$, $x_1 \geq 0$, $x_2 \geq 0$, and $\bar{c}_1 x_1 + \bar{c}_2 x_2 = z$, is in the direction $(1, 0)$. In general, use

$$\begin{aligned} x_1 + x_2 + \cdots + x_n &= \rho, & x_j &\geq 0 \\ \bar{c}_1 x_1 + \bar{c}_2 x_2 + \cdots + \bar{c}_n x_n &= z \end{aligned}$$

▷ **Exercise 1.30** Consider the problem of minimizing $f(x) = \sum_{j=m+1}^n \bar{c}_j x_j$ subject to (1.42) for fixed ρ where $x_j \geq 0$. Show that the solution is to choose $x_s = \rho$ and all other $x_j = 0$ where $\bar{c}_s = \min \bar{c}_j$. Compare this steepest descent direction with that obtained using (1.41) instead of (1.42). Does this steepest descent direction depend on the value of ρ ?

▷ **Exercise 1.31** Consider the problem of minimizing $f(x) = \sum_{j=m+1}^n \bar{c}_j x_j$ subject to (1.41) for fixed ρ where x_j is unrestricted in sign. Show that the solution is to choose $x_j = \bar{c}_j \rho / \sqrt{\sum \bar{c}_k^2}$. What is the steepest gradient direction as $\rho \rightarrow 0$? Does this steepest gradient direction depend on the value of ρ ?

1.5 THE SIMPLEX INTERPRETATION OF THE SIMPLEX METHOD

While the Simplex Method appears to be a natural one to try in the n -dimensional space of the variables, it might be expected, a priori, to be inefficient as there

could be considerable wandering on the outside edges of the convex set of solutions before an optimal extreme point is reached. This certainly appears to be true when $n - m = k$ is small as in Figure 1-12, where $k = 2$. However, empirical experience with thousands of practical problems indicates that the number of iterations is usually close to the number of basic variables in the final set that were not present in the initial set. In practical applications for an m -equation problem with m different variables in the final basic set, the number of iterations may possibly run from m as a minimum to $2m$, and very rarely to more than $3m$. The number is usually less than $3m/2$ when there are fewer than 50 equations and 200 variables (to judge from informal empirical observations). Some believe that for a randomly chosen problem with fixed m , the number of iterations grows in proportion to n .

W. M. Hirsch conjectured in 1957, that, by proper choice of variables to enter the basic set, it is always possible for linear programs with bounded solution sets to pass from any basic feasible solution to any other in m or fewer pivot steps, where each basic solution generated along the way must be feasible. For the cases $m \leq 5$ the conjecture is known to be true. For $m > 5$, the problem is a famous unsolved conjecture.

When the Simplex Method is viewed in the m -dimensional space associated with the columns of coefficients of the variables, as will be done in this section, the method appears to be quite efficient. It was in this geometry that the method was first seriously proposed, after it had been set aside earlier as apparently unpromising when viewed in the geometry of the rows.

GEOMETRY OF THE CASE $m = 2$

In *Linear Programming 1*, both the Blending Model II and the Product Mix Model were graphically solved using, as the coordinates of a point, the coefficients of a variable in one of the equations and the cost form. In both examples, one of the equations of the model was a convexity constraint of the form

$$x_1 + x_2 + \cdots + x_n = 1, \quad x_j \geq 0, \quad : \pi_0 \quad (1.46)$$

leaving, for the case $m = 2$, one other equation and cost form

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b \quad : \pi_1 \quad (1.47)$$

$$c_1x_1 + c_2x_2 + \cdots + c_nx_n = z \text{ (min)}, \quad (1.48)$$

where π_0 and π_1 are the corresponding dual multipliers. The variables x_j were interpreted as nonnegative weights to be assigned to a system of points $A_j = (a_j, c_j)$ in two dimensional space (u, v) so that their weighted average (center of gravity) is a point $R = (b, \text{min } z)$; that is, the $x_j \geq 0$ are chosen so that the center of gravity lies on the "requirement line" $u = b$ (constant) such that the v coordinate is minimum (see Figure 1-15).

In Figure 1-15, the shaded area \mathcal{C} represents the set of all possible centers of gravity G formed by assigning different weights x_j to the points A_j . It is easy to prove that these form a convex region \mathcal{C} , called the *convex hull* (see Section 1.1

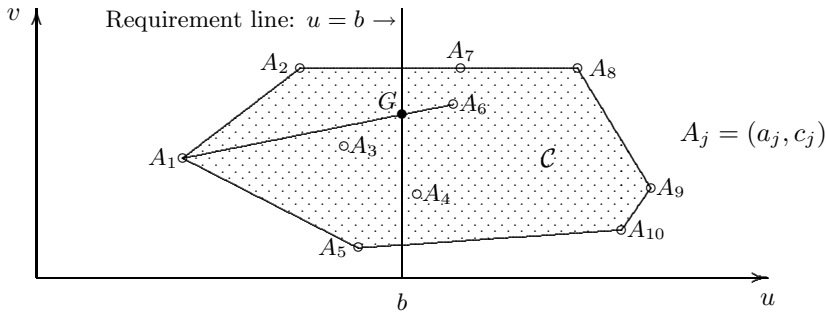


Figure 1-15: Geometrically a Linear Program is a Center-of-Gravity Problem

for the definition of a convex hull) of the set of points A_j . To see this, let G' be any point in \mathcal{C} obtained by using nonnegative weights w'_1, w'_2, \dots, w'_n and let G'' be any other point obtained by using nonnegative weights $w''_1, w''_2, \dots, w''_n$. Let $G^* = \lambda G' + (1 - \lambda)G''$, where $0 \leq \lambda \leq 1$, be any point on the line segment joining G' to G'' . It follows that G^* must also lie in \mathcal{C} because it can be obtained by using weights $w^* = \lambda w'_j + (1 - \lambda)w''_j$ for $j = 1, \dots, n$. Moreover, if $w'_j \geq 0, w''_j \geq 0, \sum w'_j = 1, \sum w''_j = 1$ and $0 \leq \lambda \leq 1$, then $w^*_j \geq 0, \sum w^*_j = 1$. This establishes the convexity of \mathcal{C} .

It is also easy to see that any column (activity) corresponding to a point A_j that is not an extreme point of the convex hull can be dropped from the linear programming problem. Thus the points A_3, A_4 , and A_6 in the interior of \mathcal{C} in Figure 1-15 and A_7 on an edge can be dropped; that is, one can set $x_3 = x_4 = x_6 = x_7 = 0$ and still obtain a feasible solution with just as low a minimum value.

A basic feasible solution corresponds to a pair of points, say A_1 and A_6 in Figure 1-15, such that the line joining A_1 to A_6 intersects the constant line $u = b$ in a point G on the line segment between A_1 and A_6 . For this to be true we would want

$$\lambda a_1 + (1 - \lambda)a_6 = b, \quad (0 \leq \lambda \leq 1).$$

But this corresponds to the basic feasible solution to (1.46) and (1.47) found by setting $x_1 = \lambda, x_6 = (1 - \lambda)$ and $x_j = 0$ for all other j .

To improve the solution, the Simplex Method first computes the relative cost factors \bar{c}_j by eliminating the basic variables from the cost equation. We shall now show that this is the same as first computing the line joining A_1 to A_6 , which we will refer to as the *solution line*, and then substituting the coordinates of a point A_j into the equation of the line to see how much (if any) in the v -direction it is above or below the line (see Figure 1-16).

In the Simplex Method the basic variables x_1 and x_6 are eliminated from the cost equation (1.48) by multiplying (1.46) by π_0 and multiplying (1.47) by π_1 and subtracting from (1.48). Thus π_0 and π_1 must be chosen so that the relative cost

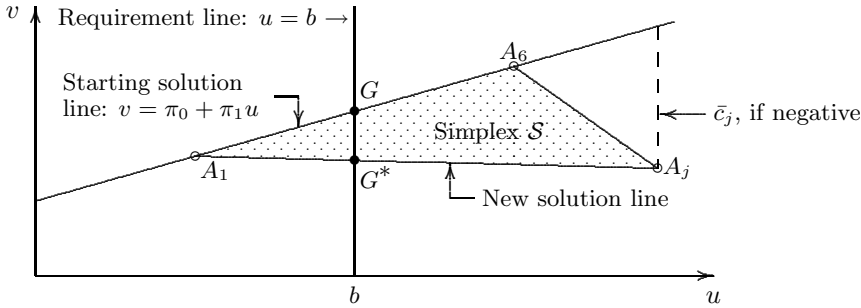


Figure 1-16: Simplex Associated with an Iteration of the Simplex Algorithm ($m = 2$)

factors for basic columns 1 and 6 are zero:

$$c_1 - (\pi_0 + \pi_1 a_1) = 0 \tag{1.49}$$

$$c_6 - (\pi_0 + \pi_1 a_6) = 0 \tag{1.50}$$

The relative cost factors \bar{c}_j for the remaining j are given by

$$\bar{c}_j = c_j - (\pi_0 + \pi_1 a_j). \tag{1.51}$$

Let us compare this with what we need to do geometrically. First we need to compute the equation of the line joining A_1 to A_6 in (u, v) space. Let

$$v = \pi_0 + \pi_1 u \tag{1.52}$$

be the equation of the line, where constants π_0 and π_1 are chosen so that the line passes through the points $A_1 = (a_1, c_1)$ and $A_6 = (a_6, c_6)$. Substituting $u = a_1$ and $v = c_1$ into Equation (1.52) gives the condition that A_1 lies on this line, and substituting $u = a_6, v = c_6$ yields the condition for A_6 to be on this line. But these are precisely conditions (1.49) and (1.50). To determine *how much* a point with coordinates $u = a_j, v = c_j$ is above or below the solution line in the v -direction, we first determine the ordinate of the point where the line $u = a_j$ cuts $v = \pi_0 + \pi_1 u$, namely at $v = \pi_0 + \pi_1 a_j$, and then subtract this value from the ordinate c_j of A_j which is exactly what we did to compute \bar{c}_j in (1.51). Thus A_j is *above, on, or below* the line according to whether $\bar{c}_j > 0, \bar{c}_j = 0,$ or $\bar{c}_j < 0$ is true.

The condition that a basic feasible solution be minimal is that $\bar{c}_j \geq 0$ for all nonbasic variables c_j . *Geometrically* it states that a *basic feasible solution is optimal* if all points A_j lie on or above the solution line corresponding to some pair of A_j s. For example, in Figure 1-15, the requirement line $u = b$ cuts the line segment joining A_5 to A_{10} , and all other points A_j lie above the extended support line joining these two points; hence the minimal solution is obtained by using x_5 and x_{10} as basic variables.

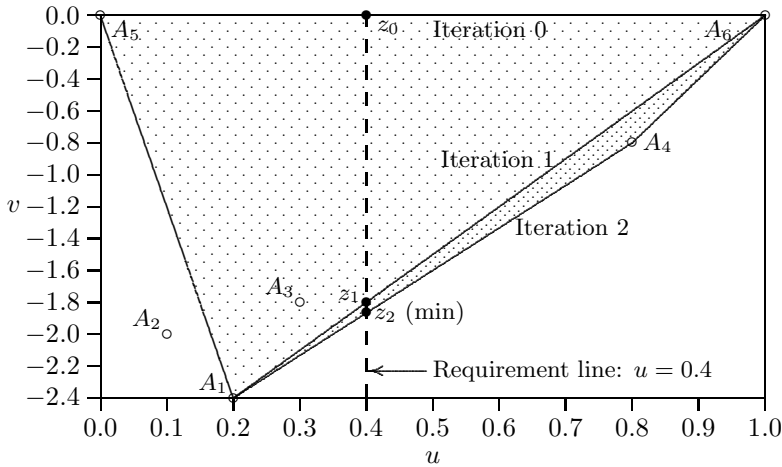


Figure 1-17: Geometry of the Simplex Algorithm for the Product Mix Problem

On the other hand, if there is a point A_j , as in Figure 1-16, below a given solution line, then join A_j to A_1 and to A_6 and consider the convex figure S formed by the points A_1 , A_6 , and A_j . This is the *convex hull of three points in general position in $m = 2$ dimensions, which is a two-dimensional simplex*. If A_j is below the solution line, every point of this simplex S is on or below the solution line. Recall that G is the intersection of the requirement line with the solution line. If G is not at a vertex, there is a segment G^*G on the requirement line belonging to S with points below the solution line with G^* the lowest point on the requirement line in S . Thus there exists a *new solution line* passing through G^* — it is either A_1A_j or A_6A_j depending on whether A_j is on the right or left of $u = b$. We are now able to repeat the iterative process with the pair of points A_1, A_j or A_6, A_j .

In Figure 1-17, we illustrate these steps of the Simplex Algorithm geometrically on (1.53) below, which is the Product Mix Problem of *Linear Programming 1*. Find $\min z, y_j \geq 0$, such that

$$\begin{aligned} .2y_1 + .1y_2 + .3y_3 + .8y_4 + 0y_5 + 1y_6 &= .4 \\ -2.4y_1 - 2.0y_2 - 1.8y_3 - .8y_4 + 0y_5 + 0y_6 &= z \text{ (min)} \end{aligned} \quad (1.53)$$

and the convexity constraint

$$y_1 + y_2 + y_3 + y_4 + y_5 + y_6 = 1 \quad (1.54)$$

Let the coordinates of a point A_j in Figure 1-17 be the coefficients of y_j in the

third and first equations:

$$A_1 = \begin{pmatrix} .2 \\ -2.4 \end{pmatrix}; A_2 = \begin{pmatrix} .1 \\ -2.0 \end{pmatrix}; A_3 = \begin{pmatrix} .3 \\ -1.8 \end{pmatrix};$$

$$A_4 = \begin{pmatrix} .8 \\ -0.8 \end{pmatrix}; A_5 = \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix}; A_6 = \begin{pmatrix} 1.0 \\ 0.0 \end{pmatrix}.$$

The simplex iterations may be summarized as follows:

Iteration	Basic variables	Solution line through	Simplex vertices
0	y_5, y_6	A_5, A_6	} A_5, A_6, A_1
1	y_1, y_6	A_1, A_6	
2	y_1, y_4	A_1, A_4	} A_4, A_6, A_1

▷ **Exercise 1.32** Consider the system of equations:

$$\begin{aligned} a_{11}y_1 + a_{12}y_2 + \cdots + a_{1n}y_n &= 0 \\ a_{21}y_1 + a_{22}y_2 + \cdots + a_{2n}y_n &= 0 \\ y_1 + y_2 + \cdots + y_n &= 1 \end{aligned} \tag{1.55}$$

where $a_{1j}^2 + a_{2j}^2 = 1$ for $j = 1, \dots, n$. Plot the column coefficients of (1.55) as points in \mathbb{R}^2 and show that the problem geometrically is to find weights on a set of n points lying on a unit circle with center at the origin so that the weighted center of gravity of the n points is the origin.

THE GEOMETRY OF THE CASE $m = 3$

For $m = 3$ dimensions, consider the problem of finding $x_j \geq 0$ and $\min z$ satisfying linear constraints, where one of the equations of the model is a convexity constraint,

$$x_1 + x_2 + \cdots + x_n = 1, \quad (x_j \geq 0). \quad : \pi_0 \tag{1.56}$$

leaving two other constraints and a cost equation.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \quad : \pi_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \quad : \pi_2 \\ c_1x_1 + c_2x_2 + \cdots + c_nx_n &= z. \end{aligned} \tag{1.57}$$

Define as coordinates (u_1, u_2, v) of a point the coefficients of x_j in (1.57); thus $A_j = (a_{1j}, a_{2j}, c_j)$. The *requirement line* is $u_1 = b_1, u_2 = b_2$. A basic feasible solution corresponds to a two-dimensional simplex with vertices, say A_1, A_2, A_3 such that the requirement line intersects this two-dimensional simplex at some interior point G as shown in Figure 1-18. Let $v = \pi_0 + \pi_1u_1 + \pi_2u_2$ be the equation of the current solution plane, that is, the plane passing through the vertices A_1, A_2 , and A_3 of the simplex. If A_j is a point below this solution plane, then algebraically the

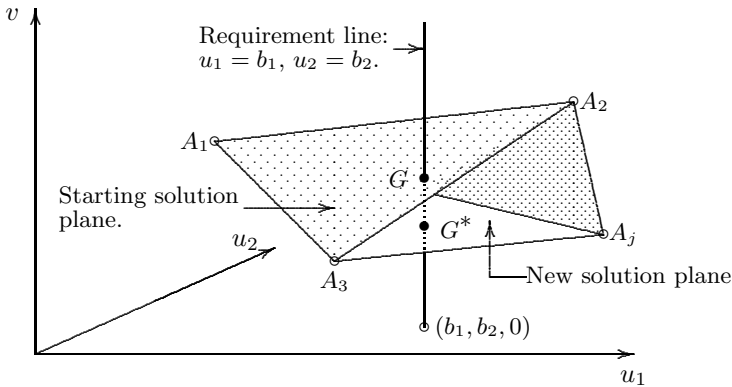


Figure 1-18: Simplex Associated with an Iteration of the Simplex Algorithm ($m = 3$)

difference $\bar{c}_j = c_j - (\pi_0 + \pi_1 a_{1j} + \pi_2 a_{2j}) < 0$ is the vertical distance that A_j is below the plane. In this case, a three-dimensional simplex with vertices A_j , A_1 , A_2 , and A_3 can be formed and a point G^* found where the requirement line pierces the simplex at its lowest point. Then G^* is on one of the three faces $A_1A_2A_j$, $A_2A_3A_j$, $A_1A_3A_j$, depending on the position of A_j . In Figure 1-18, G^* lies on the face $A_1A_3A_j$ and these three vertices A_1 , A_3 , A_j are used to determine the new two-dimensional simplex.

The simplex criterion used to select a new basic variable x_s does not select an arbitrary x_j corresponding to an A_j below the solution plane, but an $A_s = A_j$ that is a *maximum* distance $\bar{c}_s = \min \bar{c}_j$ below the plane. Inspection of figures such as Figure 1-15 and Figure 1-16 give credence to the belief that choosing such a point would result in the point having a good chance of being in the optimal solution. If the point chosen on each iteration is not dropped on some later iteration, then no more than m iterations would be required to obtain an optimal solution. Empirical evidence on thousands of problems confirms this choice criterion as a pretty good practice, particularly when one notes that the computational work per iteration is low. In Chapter 6 other criteria will be presented for selecting the incoming column that result in fewer iterations and/or fewer computations than the simplex criterion used here.

- ▷ **Exercise 1.33** Study Figure 1-12 and Figure 1-14. Construct an example to show for $n = m + 2$ that the simplex criterion $\bar{c}_s = \min \bar{c}_j$ could cause a maximum number of iterations to be performed.

1.6 NOTES & SELECTED BIBLIOGRAPHY

Klee & Minty [1972] created special examples (see *Linear Programming 1*) to show that the Simplex Algorithm, using the criterion $\bar{c}_s = \min \bar{c}_j$ to select the incoming column, would in the worst case pass through every extreme point before termination. However, computational experience on many thousands of practical problems using this criterion has demonstrated that the Simplex Algorithm required less than $3m$ steps. See, for example, Klotz [1988], Lustig [1987], and Gill, Murray, Saunders, & Wright [1989].

The results reported in Gill, Murray, Saunders, & Wright [1989] for 53 problems drawn from practical situations and varying sizes ($m = 28$ to $m = 2263$; and $n = 32$ to $n = 9799$) show that: 23 of the problems are solved in less than m iterations; 14 are solved in between m and $2m$ iterations; 3 are solved in between $2m$ and $3m$ iterations; and 13 required more than $3m$ iterations. For many years the explanation for this observed efficiency of the Simplex Method remained a mystery. Papers that began to appear in the early 1980s provided a partial theoretical explanation for randomly generated problems solved by the Simplex Algorithm and its variants; see Borgwardt [1982a, 1982b, 1987a, 1987b] and Smale [1982]. For example, Smale showed that when the linear program $\min z = c^T x$, $x \geq 0$, $Ax \geq b$ is solved by the self-dual parametric algorithm (see Section 6.5), the average number of iterations grows proportional to n when m is fixed. Borgwardt showed, for the linear program $\min z = -c^T x$, $x \geq 0$, $Ax \leq e$, where $e = (1, 1, \dots, 1)^T$ and where c and the rows of A are assumed to be independently, and identically distributed, and symmetrically distributed under rotation about the origin, that the expected number of iterations (by a variant of the Simplex Algorithm) grows proportional to $n^4 m^{1/(n-1)}$.

In 1957, W. M. Hirsch conjectured that, by proper choice of variables to enter the basic set, it is always possible for linear programs with bounded solution sets to pass from any basic feasible solution to any other in m or fewer pivot steps, where each basic solution generated along the way must be feasible. For the cases $m \leq 5$ the conjecture is known to be true. For $m > 5$, the problem is a famous unsolved conjecture. Klee and Walkup [1967] have constructed examples to show that the Hirsch conjecture is false if the set of feasible solutions is unbounded. They have also proved that the Hirsch conjecture is true for all polytopes for which $n - m \leq 5$. Todd [1980] has provided a counter-example for the Monotonic Bounded Hirsch conjecture.

The concept of separating hyperplanes and the theorems of the alternatives can be used to prove many important results in mathematical theorems. For further discussions on separation theorems see Avriel [1976], Berge [1963], and Rockafellar [1970]. Tucker [1955] is recommended as reading for Section 1.3 which discusses properties of the Simplex Method.

1.7 PROBLEMS

1.1 *Review:* Define the following terms:

- (a) Convex set, extreme point of a convex set.
- (b) Polytope, polyhedron.

(c) *Degenerate basic feasible solution* for:

$$\begin{array}{ll} \text{Minimize} & c^T x = z \\ \text{subject to} & Ax = b \\ & x \geq 0. \end{array}$$

(d) Convexity constraint.

- 1.2 Review the relationship between convex sets and linear programming.
 1.3 Look at the feasible region defined by:

$$\begin{array}{l} x_1 + 2x_2 \leq 6 \\ 2x_1 + x_2 \leq 6 \\ 2x_1 + 2x_2 \leq 7 \\ x_1, x_2 \geq 0. \end{array} \quad (1.58)$$

- (a) Set up an initial tableau with three slack variables.
 (b) Draw the feasible region in (x_1, x_2) -space. Label the constraints.
 (c) Notice that, including nonnegativity, we have five constraints and five variables. We can associate each variable with a constraint, so that for each extreme point of the feasible region there corresponds a basic feasible solution, and so that the extreme point is the intersection of the constraints associated with the nonbasic variables of the solution. What is this association, i.e., what variables do we associate with what constraints? What is the basic feasible solution corresponding to each extreme point of the feasible region?
 (d) Suppose we add the constraint:

$$x_1 \leq 3. \quad (1.59)$$

In your diagram, the extreme point $(3, 0)$ of the feasible region is now the intersection of three constraints, and any two of them will uniquely specify that extreme point. Thus there are three distinct bases that correspond to that extreme point. What are the basic variables in each of these three bases? Is it still true that there is a one-to-one correspondence between basic feasible solutions and extreme points of the feasible region? Show that the basic feasible solutions corresponding to the extreme point $(3, 0)$ are all degenerate.

- (e) In part (d) we created an example of degeneracy by using a redundant system of inequalities. The redundancy can be seen in the diagram in that we could remove one of the constraints without changing the feasible region. Give an example of degeneracy with a *nonredundant* system of inequalities. Draw a picture to demonstrate this.

1.4 *Dantzig [1963].*

- (a) Show that the set of possible values of any variable x_k of a linear program forms a convex set, in this case, a straight line segment $a \leq x_k \leq b$.
 (b) Show that the set of possible values of two variables, say (x_1, x_2) or (x_1, z) satisfying a linear program, forms a convex set in two dimensions.

- (c) As a corollary to part (a), show if x_k is treated as a parameter and can take on a range of possible values, then the value of $\min z$ becomes a convex function of x_k .
- 1.5 Let $F = \{x \in \mathbb{R}^n \mid Ax \leq b\}$. For each $x \in F$, let $T(x) = \{i \mid A_{i\bullet}x = b_i\}$ be the “tight” constraints of x . Show that x is an extreme point of F if and only if the rank of $A_{T(x)\bullet}$ is n .
- 1.6 Given the coordinates (x_1, x_2) of five points in \mathbb{R}^2 :

$$P_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad P_3 = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \quad P_4 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \text{and} \quad P_5 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}.$$

- (a) Write down the relations defining the convex hull of $P_1, P_2, P_3, P_4,$ and P_5 .
- (b) Use the FME process (see *Linear Programming 1*) to define the feasible region in terms of x_1 and x_2 only.
- (c) Plot the points $P_1, P_2, P_3, P_4,$ and P_5 and verify your result.
- 1.7 The hypercube

$$0 \leq x_j \leq 1 \quad \text{for } j = 1, \dots, n$$

- is expressed through $2n$ inequalities. If expressed as a convex hull of its extreme points, show that there are 2^n variables $\lambda_j \geq 0$ and n variables x_j in $n + 1$ constraints.
- 1.8 *Dantzig [1963]*.
- (a) The process of increasing the variable x_s in the Simplex Algorithm, while holding the other independent variables fixed at zero, generates a class of solutions corresponding to an edge in a convex polyhedron of feasible solutions in the case that the vertex corresponds to a nondegenerate basic feasible solution. What can happen in the case that the vertex is degenerate?
- (b) If a basic solution is nondegenerate, there are precisely $n - m$ neighbors of its corresponding extreme point, and these are generated by increasing one of the $n - m$ independent variables while holding the remainder fixed at zero. What can happen if the basic solution is degenerate?
- 1.9 *Dantzig [1963]*. Devise a method for finding the second best basic feasible solution. Generalize to the third best, fourth best, etc. Discuss any complications.
- 1.10 *Dantzig [1963]*. Show that if r variables have unique and nonnegative values when the remaining variables are set equal to zero, the feasible solution is an extreme-point solution.
- 1.11 *Dantzig [1963]*. Given an extreme-point solution (v_1, v_2, \dots, v_n) , show that if the variables x_j are set equal to zero corresponding to $v_j = 0$, then the remaining variables are uniquely determined and $x_j = v_j > 0$.
- 1.12 *W. M. Hirsch Conjecture, [1957, Private Communication with Dantzig], unsolved*. Does there exist a sequence of m or less pivot operations, each generating a new basic feasible solution, which starts with some given basic feasible solution and ends with some other given basic feasible solution, where m is the number of equations? Expressed *geometrically*:

Given a bounded convex region in $(n - m)$ -dimensional space defined by n half-planes, is m an upper bound for the minimum-length chain of adjacent vertices joining two given vertices? If not, what is the minimum length chain of adjacent vertices joining the two given vertices.

- 1.13 *Gale in Dantzig [1963]*. Prove that a square homogeneous linear inequality system always has a nontrivial solution.
- 1.14 *Dantzig [1963]*. Suppose $P_1, P_2, \dots, P_k, \dots$ is an infinite collection of points in m -dimensional space. Let \mathcal{C} be the set of points generated by forming nonnegative linear combinations of finite subsets of these points. Let \mathcal{C}' be the set of points generated by forming nonnegative linear combinations of subsets of m or fewer of these points. Show that \mathcal{C} and \mathcal{C}' are identical convex cones.
- 1.15 *Ph.D. Comprehensive Exam, June 15, 1967, at Stanford*. Let

$$\mathcal{C} = \{x \in \mathbb{R}^n \mid Ax = b, \quad x \geq 0, \quad c^T x = z_{\min}\}$$

where $x_{\min} > -\infty$ is the minimal value of x in the linear program

$$\text{Minimize } z = c^T x \quad \text{subject to } Ax = b, \quad x \geq 0.$$

- (a) Prove that \mathcal{C} is a convex set in \mathbb{R}^n .
- (b) What is true about \mathcal{C} if the solution to the linear program is unique?
- (c) Prove that

$$\bar{\mathcal{C}} = \{(b, z) \mid Ax = b, \quad x \geq 0, \quad c^T x = z, \text{ for some } x\}$$

is a convex set.

- (d) Prove that

$$\bar{\mathcal{C}} = \{(b, \bar{z}) \mid \text{Min } c^T x = \bar{z} \text{ for } Ax = b, \quad x \geq 0 \text{ and each fixed } b\}$$

is a convex set.

- (e) Prove that \bar{z} in part (d) can be regarded as a convex function of b .

- 1.16 Carry out the steps of the Simplex Method both algebraically and geometrically on

- (a) The Product Mix Problem (*Linear Programming 1*):

$$\begin{array}{llllll} \text{Minimize} & -12x_1 - 20x_2 - 18x_3 - 40x_4 & & = & z \\ \text{subject to} & 4x_1 + 9x_2 + 7x_3 + 10x_4 + x_5 & & = & 6000 \\ & x_1 + x_2 + 3x_3 + 40x_4 & + & x_6 & = & 4000 \\ & x_j \geq 0, j = 1, \dots, 6. \end{array}$$

- (b) The Blending Problem (*Linear Programming 1*):

Minimize the *Objective*

$$4.1x_1 + 4.3x_2 + 5.8x_3 + 6.0x_4 + 7.6x_5 + 7.5x_6 + 7.3x_7 + 6.9x_8 + 7.3x_9 = z$$

subject to

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 = 1$$

$$.2x_1 + .5x_2 + .3x_3 + .3x_4 + .3x_5 + .6x_6 + .4x_7 + .1x_8 + .1x_9 = .3$$

$$.3x_1 + .4x_2 + .2x_3 + .4x_4 + .3x_5 + .3x_6 + .5x_7 + .3x_8 + .1x_9 = .3$$

$$.5x_1 + .1x_2 + .5x_3 + .3x_4 + .4x_5 + .1x_6 + .1x_7 + .6x_8 + .8x_9 = .4$$

and $x_j \geq 0, j = 1, \dots, 9$.

Show the correspondence between the algebraic and geometric methods.

1.17 Dantzig [1963].

- (a) Use the Fourier-Motzkin Elimination procedure (see *Linear Programming 1*) to solve

$$\begin{aligned} 3y_1 + 4y_2 &= v \quad (\text{max}) \\ 2y_1 + y_2 &\leq 2 \\ -3y_1 + y_2 &\leq -3 \\ y_1 - 2y_2 &\leq 6 \\ 3y_1 + 9y_2 &\leq 1 \\ -y_1 &\leq -2. \end{aligned}$$

- (b) Solve the preceding, using the following variant of the Simplex Method: for those with positive right-hand sides introduce *slack* variables $y_j \geq 0$; for those with nonpositive right-hand sides introduce *artificial excess* variables $y_j \geq 0$. Apply the usual Simplex Method to minimizing the sum of artificial variables, in this case $y_4 + y_7 = w$. However, note that y_1 and y_2 are not restricted in sign; see part (c).
- (c) Invent a variant of the Simplex Method that permits specified variables to be unrestricted in sign. Apply this to part (b).

1.18 Dantzig [1963]. Solve

$$\begin{aligned} 3y_1 + 4y_2 &= v \quad (\text{max}) \\ 2y_1 + y_2 &\leq 2 \\ y_1 - 2y_2 &\leq 6 \\ 3y_1 + 9y_2 &\leq 1 \\ y_1 &\geq 0 \\ y_2 &\geq 0. \end{aligned}$$

using the Simplex Method. Interpret geometrically the simplex steps in the two-dimensional space of y_1 and y_2 .

1.19 Dantzig [1963].

- (a) Given a system

$$\begin{aligned} c_1x_1 + c_2x_2 + \cdots + c_nx_n &= z \quad (\text{min}) \\ a_1x_1 + a_2x_2 + \cdots + a_nx_n &= b \\ x_1 + x_2 + \cdots + x_n &= 1 \end{aligned} \quad (x_j \geq 0)$$

show that the solution line $v = \pi_0^* + \pi_1^*u$ associated with the minimal basic solution must satisfy

$$\begin{aligned} \pi_0^* + \pi_1^*b &= \min z \\ c_j - (\pi_0^* + \pi_1^*a_j) &\geq 0 \end{aligned}$$

- (b) Prove in part (a) that the convex hull of points $A_j = (a_j, c_j)$ lies on or above some given line $v = \pi_0 + \pi_1u$, if

$$c_j - (\pi_0 + \pi_1a_j) \geq 0.$$

Use this to show that such a line must cut the requirement line $u = b_1$ in a point, whose ordinate $v \leq \min z$.

- 1.20 *Dantzig [1963]*. Note that the dual of a standard linear program is a system of inequalities in unrestricted variables. Suppose one is given a system in the latter form; review how its dual may be used as a *third way* to get a standard linear program from a system of linear inequalities. Find the standard linear program of which this is the dual:

$$\begin{aligned}\pi_0 + .2\pi_1 &\leq -2.4 \\ \pi_0 + .1\pi_1 &\geq -2.0 \\ \pi_0 + .3\pi_1 &\leq -1.8 \\ \pi_0 + .8\pi_1 &\leq -.8 \\ \pi_0 &\leq 0 \\ \pi_0 + \pi_1 &\leq 0 \\ \pi_0 &\leq v.\end{aligned}$$

Solve the dual, by using the Simplex Method and also by using the elimination method, and prove that $\max v = \min z$ of the dual original system.

- 1.21 *Dantzig [1963]*. If $v = \pi_0 + \pi_1 u_1 + \pi_2 u_2$ represents the solution plane associated with A_1, A_2, A_3 in Figure 1-18, interpret the conditions

$$v_j - (\pi_0 + \pi_1 a_{1j} + \pi_2 a_{2j}) = 0 \quad \text{for } j = 1, 2, 3$$

and the quantities

$$v_j - (\pi_0 + \pi_1 a_{1j} + \pi_2 a_{2j}) = \bar{c}_j$$

both algebraically in the Simplex Method and geometrically.

- 1.22 *Wolfe [1960]*. A third geometry of the Simplex Method can be obtained by regarding a column j as representing a line $\pi_0 + a_j \pi_1 = c_j$ in (π_0, π_1) -space. Thus, this procedure can be interpreted to be in the same space as the space of independent variables π_0 and π_1 of the *dual linear programming problem* $\pi_0 + b\pi_1 = v$ (max), $\pi_0 + a_j \pi_1 \leq c_j$, for $j = 1, \dots, n$. Show that the simplex procedure for solving the dual is different from the interpretation of the simplex procedure for solving the original problem in this geometry. (The procedure of Kelley, see Wolfe [1960], for solving nonlinear programs is based on this geometry.)

- 1.23 (a) Interpret the problem: Find $x_j \geq 0$, $j = 1, \dots, 4$, and $\min z$ satisfying

$$\begin{aligned}x_1 + 2x_2 + 3x_3 + 4x_4 &= z \text{ (Min)} \\ x_1 + x_2 + x_3 + x_4 &= 4 \\ x_1 + 2x_2 + 3x_3 + 4x_4 &= -2\end{aligned}$$

as a Center-of-Gravity-Problem, see Section 1.5

- (b) Dualize and graph the dual problem.
 (c) Solve the dual using the Fourier-Motzkin Elimination Method (see *Linear Programming 1*).
 (d) Solve the primal using the Simplex Method. Trace the steps of the procedure as graphed in (a) and (b).
- 1.24 *Minkowski [1896]*. *Theorem: A feasible solution of a bounded linear program can be expressed as a nonnegative linear combination of basic feasible solutions.*

Geometrically stated, a point of a bounded convex polyhedron \mathcal{C} , defined as the intersection of finitely many half-spaces, can be expressed as a nonnegative linear combination of extreme points of \mathcal{C} .

Show that the theorem is false if \mathcal{C} is unbounded.

1.25 *Steinitz [1913]. Theorem:* Let \mathcal{M} be a given set of points in a Euclidean $(m-1)$ -dimensional space and let Q be in the convex hull of \mathcal{M} . It is possible to find m points P_1, P_2, \dots, P_m (not necessarily different) of \mathcal{M} , and m real numbers $x_1 \dots x_m$ so that $x_i \geq 0$, $\sum_1^m x_i = 1$, and $\sum_1^m x_i P_i = Q$.

1.26 *Dantzig [1963]. Theorem:* Let \mathcal{M} be a given infinite set of points in Euclidean m -dimensional space and let Q be in the convex cone spanned by \mathcal{M} . It is possible to find m points P_1, P_2, \dots, P_m (not necessarily different) of \mathcal{M} , and m real numbers $x_1 \geq 0, \dots, x_m \geq 0$, so that $\sum_1^m x_i P_i = Q$.

Hint: Establish this theorem for any point Q representable as a nonnegative finite linear combination of points $P_i \in \mathcal{M}$. Show that all such points Q define the convex cone spanned by \mathcal{M} .

1.27 For the following system, is $(4, 9, 0, 3, 0, 0)^T$ an extreme point? If so, why? If not, is it on an edge?

$$\begin{aligned} x_1 + x_2 & & - 3x_4 + 3x_5 + x_6 & = & 4 \\ x_1 + 2x_2 & & - 5x_4 + 5x_5 + 3x_6 & = & 7 \\ & - x_2 + x_3 + 2x_4 - 5x_5 + x_6 & = & -3 \\ x_j & \geq 0, & j & = 1, \dots, 6. \end{aligned}$$

1.28 Consider a polyhedron in n -dimensions defined by the following set of linear constraints

$$\{x \mid Ax \leq b\}$$

Suppose that we wish to embed this polyhedron in the “smallest” possible rectangle whose sides are parallel to the coordinate axes. Discuss how linear programming can be used to solve this problem.

1.29 Let P_1, P_2, \dots, P_n be points in \mathfrak{R}^m where m and n are finite positive integers and P_j are distinct points in \mathfrak{R}^m .

(a) Prove

$$\mathcal{S} = \left\{ P \mid P = \sum_{j=1}^n P_j x_j, \sum_{j=1}^n x_j = 1, x_j \geq 0 \right\} \in \mathfrak{R}^m$$

is a convex set.

(b) Prove \mathcal{S} is a bounded convex set in \mathfrak{R}^m .

(c) Prove \mathcal{S} is a bounded polyhedral set (i.e., polytope) in \mathfrak{R}^m .

(d) Prove that there are no extreme points of \mathcal{S} other than some subset of the P_1, P_2, \dots, P_n .

(e) Prove that \mathcal{S} has only a finite number of extreme points.

(f) Prove that \mathcal{S} has at least one extreme point.

(g) Given some P_i , say P_1 , how would you determine whether or not P_1 is an extreme point?

(h) If the system

$$\sum_{j=2}^n x_j = 1 \text{ where } x_j \geq 0 \text{ for } j = 2, \dots, n,$$

$$\sum_{j=2}^n P_j x_j = P_1,$$

is infeasible, prove that there exist $(\pi_0, \pi)^T$ such that

$$\begin{aligned} \pi_0 + \pi^T P_j &\geq 0 \quad \text{for } j = 2, \dots, n \\ \pi_0 + \pi^T P_1 &\leq -1 \end{aligned}$$

Also prove that every point P_j , $j > 1$, is separated from P_1 by two parallel hyperplanes a distance greater than Δ where

$$\Delta = \min_{j=2, \dots, n} (\pi_0 + \pi^T P_j) - (\pi_0 + \pi^T P_1).$$

What is used to measure distance? What would be the formula if the Euclidean distance was used?

- (i) Given two polytopes \mathcal{C}_1 and \mathcal{C}_2 in \mathfrak{R}^m defined by

$$\begin{aligned} \mathcal{C}_1 &= \left\{ P \mid P = \sum_{j=1}^n P_j x_j, \sum_{j=1}^n x_j = 1, x_j \geq 0 \right\} \in \mathfrak{R}^m \\ P_j &\in \mathfrak{R}^m \quad \text{for } j = 1, \dots, n \end{aligned}$$

$$\begin{aligned} \mathcal{C}_2 &= \left\{ Q \mid Q = \sum_{k=1}^{\bar{n}} Q_k y_k, \sum_{k=1}^{\bar{n}} y_k = 1, y_k \geq 0 \right\} \in \mathfrak{R}^m, \\ Q_k &\in \mathfrak{R}^m \quad \text{for } k = 1, \dots, \bar{n}, \end{aligned}$$

how would you determine whether \mathcal{C}_1 and \mathcal{C}_2 have points in common?

- (j) Given two nonempty polyhedral sets \mathcal{C}_1 and \mathcal{C}_2 in \mathfrak{R}^m defined by:

$$\begin{aligned} \mathcal{C}_1 &= \{ x \mid Ax \geq b \}, \quad A : m \times n, \\ \mathcal{C}_2 &= \{ x \mid \bar{A}x \geq \bar{b} \}, \quad \bar{A} : m \times n, \end{aligned}$$

how would you determine whether the polyhedral sets \mathcal{C}_1 and \mathcal{C}_2 have a point $x = x^o$ in common?

- 1.30 *Bazarrá, Jarvis, & Sherali [1990].* Let \mathcal{S} be a nonempty open set; i.e., if $x_0 \in \mathcal{S}$ then there exists an $\epsilon > 0$ such that $\|x - x_0\| < \epsilon$ implies that $x \in \mathcal{S}$. Show that the problem

$$\begin{aligned} \text{Minimize} \quad & c^T x = z \\ \text{subject to} \quad & x \in \mathcal{S}, \end{aligned}$$

with $c \neq 0$ possesses no optimal solution.

- 1.31 *Ph.D. Comprehensive Exam, September 21, 1974, at Stanford.* Solve the following problems:

- (a) Consider the set

$$\mathcal{X} = \{ x \mid Ax \leq a \}$$

where A is an $m \times n$ matrix. The dimension of \mathcal{X} is defined to be the dimension of the highest-dimensional hyperplane whose interior lies in \mathcal{X} . Suppose there is a point $\bar{x} \in \mathcal{X}$ such that

$$A_i \bar{x} < a_i$$

where A_i denotes the i th row of A . Let

$$\mathcal{Y} = \{y \mid By \leq b\}$$

where $[B, b]$ is obtained from $[A, a]$ by deleting its i th row. Prove that the sets \mathcal{X} and \mathcal{Y} have the same dimension.

- (b) Devise a scheme to determine the dimension of the set

$$\mathcal{Z} = \{z \mid Cz = c, z \geq 0\}, \quad \text{where } C \text{ is a } m \times n \text{ matrix.}$$

Assume the set \mathcal{Z} is nonempty.

- 1.32 *Ph.D. Comprehensive Exam, September 21, 1974, at Stanford.* Let $A = [a_{ij}]$ denote a positive matrix of order $m \times n$, i.e.,

$$a_{ij} > 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n.$$

Prove, given $r_i \geq 0, i = 1, \dots, m$ and $c_j \geq 0, j = 1, \dots, n$ with

$$\sum_{i=1}^m r_i = \sum_{j=1}^n c_j$$

that there exists $u_i \geq 0, i = 1, \dots, m, v_j \geq 0, j = 1, \dots, n$ such that

$$\begin{aligned} \sum_{j=1}^n u_i a_{ij} v_j &= r_i, & i = 1, \dots, m, \\ \sum_{i=1}^m u_i a_{ij} v_j &= \alpha c_j, & j = 1, \dots, n. \end{aligned}$$

- 1.33 *Ph.D. Comprehensive Exam, September 27, 1975, at Stanford.* In a paper submitted for publication in an operations research journal, the author considered a set

$$\mathcal{S} = \{(x, y) \mid Ax + By \geq c, x \geq 0, y \geq 0\}$$

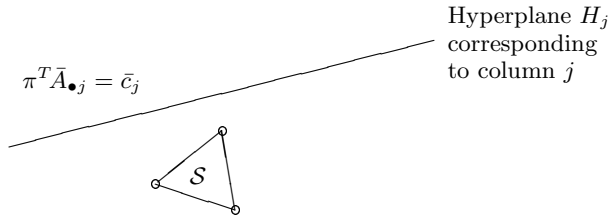
where A is an $m \times n$ matrix, B is a positive semi-definite $m \times m$ matrix and $c \in \mathbb{R}^m$. The author explicitly assumed the set \mathcal{S} is compact in \mathbb{R}^{n+m} . A reviewer of the paper pointed out that the only compact set of the above form is the empty set. Prove the reviewer's assertion.

- 1.34 *Ph.D. Comprehensive Exam, September 1982, at Stanford.* Given two sets

$$\begin{aligned} \mathcal{S} &= \{x \in \mathbb{R}^n \mid Ax = b, x \geq 0\}, \\ \mathcal{T} &= \{x \in \mathbb{R}^n \mid \bar{A}x \leq \bar{b}\}. \end{aligned}$$

- (a) Prove \mathcal{S} is a convex set.
 (b) Set up a linear program for determining whether \mathcal{S} and \mathcal{T} have a point in common.
 (c) Show how to use the Simplex Method for determining a separating hyperplane when \mathcal{S} and \mathcal{T} have no point in common. That is, find a hyperplane $\pi^T x = \pi_o$ such that

$$\pi^T x < \pi_o \quad \text{for all } x \in \mathcal{S} \quad \text{and} \quad \pi^T x > \pi_o \quad \text{for all } x \in \mathcal{T}.$$

Figure 1-19: Hyperplane H_j and Simplex \mathcal{S}

- 1.35 *Ph.D. Comprehensive Exam, September 26, 1987, at Stanford.* Yinyu Ye [1987] proposes a criterion that, if satisfied by column j on iteration t of the Simplex Method, allows one to drop the column because it cannot be in any optimal basis.

The PRIMAL linear program is

$$\begin{aligned} & \text{Minimize} && \bar{c}^T x = z \\ & \text{subject to} && \bar{A}x = \bar{b}, \quad \bar{b} > 0, \quad \bar{A} : m \times n, \\ & && x \geq 0. \end{aligned}$$

Assume we are in iteration t and the system is in canonical form so that

$$\begin{aligned} \bar{A} &= [I, N], & \text{note that } \bar{b} > 0, \\ \bar{c} &= [0, \bar{c}_N], \end{aligned}$$

where N refers to the nonbasic column of \bar{A} , \bar{c} the corresponding relative cost factors, and I is the identity matrix.

The DUAL problem is

$$\begin{aligned} & \text{Maximize} && \pi^T \bar{b} = \underline{z} \\ & \text{subject to} && \pi^T \bar{A} \leq \bar{c} \\ & && \pi \leq 0. \end{aligned}$$

Assume z_{LB} is a known finite lower bound for the primal system.

- Prove (or cite a theorem) that an optimal primal feasible solution exists.
- Prove $z_{LB} \leq \max \underline{z}$.
- Prove that dual optimal $\pi = \pi^*$ satisfies conditions $\pi^T \bar{b} \geq z_{LB}$, $\pi_1 \leq 0$, $\pi_2 \leq 0, \dots, \pi_m \leq 0$.
- Prove that conditions on π described in (c) form a simplex \mathcal{S} in the dual m -dimensional space of π .
- Prove (or cite a theorem) that the set of π that satisfies the set of dual conditions,

$$\pi^T \bar{A}_{\bullet,j} \leq \bar{c}_j, \quad j = 1, \dots, n,$$

form a convex set in π -space where $\bar{A}_{\bullet,j}$ is column j of \bar{A} .

- Prove that a basic feasible solution to the primal problem corresponds to the point in π -space where m hyperplanes,

$$H_j = \{ \pi \mid \pi^T \bar{A}_{\bullet,j} = \bar{c}_j \}$$

for some $j = j_1, j_2, \dots, j_m$, intersect. Show that the origin $\pi = 0$ is such a point π . Show that the origin $\pi = 0$ lies in the simplex \mathcal{S} , and in fact is one of the vertices of \mathcal{S} . See Figure 1-19.

- (g) Prove that optimal basic feasible solution to the primal problem corresponds to a point π^* in π space where certain m of the hyperplanes H_j intersect. Prove $\pi^* \subset \mathcal{S}$.
- (h) Prove that if a column j corresponding to a hyperplane $\pi^T \bar{A}_{\bullet j} = \bar{c}_j$ does not intersect simplex \mathcal{S} , column j cannot be a basic column of any optimal basic feasible solution and therefore can be dropped.
- (i) Specify exactly what the m coordinates of each of the $m + 1$ vertices of \mathcal{S} are. Devise an algebraic test for checking that H_j does not intersect \mathcal{S} .

This page intentionally left blank

DUALITY AND THEOREMS OF THE ALTERNATIVES

2.1 THE DUALITY THEOREM

The primal problem for a linear program stated in von Neumann “symmetric” form is:

$$\begin{array}{llll} & \text{Minimize} & c^T x = z & \\ \text{PRIMAL:} & \text{subject to} & Ax \geq b, & A : m \times n, \\ & & x \geq 0, & \end{array} \quad (2.1)$$

and the dual problem is

$$\begin{array}{llll} & \text{Maximize} & b^T y = v & \\ \text{DUAL:} & \text{subject to} & A^T y \leq c, & A : m \times n, \\ & & y \geq 0. & \end{array} \quad (2.2)$$

The von Neumann symmetric form is actually not symmetric but skew-symmetric because the full system of relations is:

$$\begin{pmatrix} 0 & A & -b \\ -A^T & 0 & c \\ b & -c & 0 \end{pmatrix} \begin{pmatrix} y \\ x \\ 1 \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad x \geq 0, y \geq 0. \quad (2.3)$$

The *Duality Theorem* is a statement about the range of possible z values for the primal versus the range of possible v values for the dual. This is depicted graphically in Figure 2-1, for the case where the primal and dual are both feasible.

Von Neumann stated but did not prove the Duality Theorem: *If the primal (2.1) and dual (2.2) have feasible solutions, then there exist optimal feasible solutions to*

both the primal and the dual that are equal. We shall formally state and prove the Duality Theorem using the Infeasibility Theorem, which is proved using the Fourier-Motzkin Elimination Process; see *Linear Programming 1*. Here we state the Infeasibility Theorem without proof.

THEOREM 2.1 (Infeasibility Theorem) *The system of linear inequalities*

$$\sum_{j=1}^n a_{ij}x_j \geq b_j \quad \text{for } i = 1, \dots, m \quad (2.4)$$

is infeasible if and only if there exists a nonnegative linear combination of the inequalities that is an infeasible inequality. In matrix notation, the system $Ax \geq b$ is infeasible if and only if there exists a vector $y \geq 0$ such that $y^T A x \geq y^T b$ is an infeasible inequality, namely one where $y^T A = 0$ and $y^T b > 0$.

▷ **Exercise 2.1** State the Infeasibility Theorem in terms of the system

$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned} \quad (2.5)$$

and apply Phase I of the Simplex Algorithm to prove the Infeasibility Theorem.

COROLLARY 2.2 (Infeasible Equation) *If a system of linear equations in nonnegative variables is infeasible, there exists a linear combination of the equations that is an infeasible equation in nonnegative variables.*

Assuming that primal and dual solutions exist, the weaker form of the Duality Theorem, which follows, is obvious.

THEOREM 2.3 (Weak Duality Theorem) *If x^o is any feasible solution to the primal (2.1) and y^o is any feasible solution to the dual (2.2), then*

$$y^{oT}b = v^o \leq z^o = c^T x^o. \quad (2.6)$$

Proof. We have

$$\begin{aligned} Ax^o &\geq b & y^{oT}A &\leq c^T \\ c^T x^o &= z^o & y^{oT}b &= v^o \end{aligned}$$

Multiplying $Ax^o \geq b$ by y^{oT} on the left and multiplying $y^{oT}A \leq c^T$ by x^o on the right we obtain

$$\begin{aligned} y^{oT}Ax^o &\geq y^{oT}b = v^o \\ y^{oT}Ax^o &\leq c^T x^o = z^o \end{aligned}$$

Therefore,

$$v^o = y^{oT}b \leq y^{oT}Ax^o \leq c^T x^o = z^o.$$

This concludes our proof. ■

COROLLARY 2.4 (Bounds on the Objectives) *Every feasible solution y^o to the dual yields a lower bound $y^{oT}b$ to values of z^o for feasible solutions x^o to the primal. Conversely, every feasible solution x^o to the primal yields an upper bound $c^T x^o$ to values of v^o for feasible solutions y^o to the dual.*

▷ **Exercise 2.2** Prove Corollary 2.4.

COROLLARY 2.5 (Optimality) *If $v^o = z^o$ then $v^o = \max v$ and $z^o = \min z$.*

We can depict the relationship by plotting the points v^o and z^o on a line as shown in Figure 2-1.

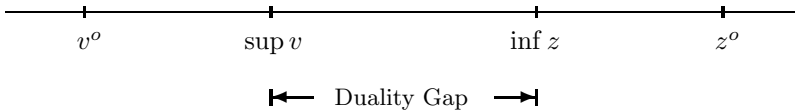


Figure 2-1: Illustration of the Duality Gap

We are now ready to formally state and prove Von Neumann's Duality Theorem which states that if feasible solutions to the primal and dual exist then the *duality gap* (depicted in Figure 2-1) is zero and $\sup v$ is actually attained for some choice of y , and $\inf z$ is attained for some choice of x .

THEOREM 2.6 (Strong Duality Theorem) *If the primal system $\min z = c^T x$, $Ax \geq b$, $x \geq 0$ has a feasible solution and the dual system $\max v = b^T y$, $A^T y \leq c$, $y \geq 0$ has a feasible solution, then there exist optimal feasible solutions $x = x^*$ and $y = y^*$ to the primal and dual systems such that*

$$b^T y^* = \max v = \min z = c^T x^*. \quad (2.7)$$

Proof. Consider the system of inequalities and corresponding infeasibility multipliers:

$$Ax \geq b \quad : \quad \tilde{y} \quad (2.8)$$

$$Ix \geq 0 \quad : \quad \tilde{u} \quad (2.9)$$

$$-A^T y \geq -c \quad : \quad \tilde{x} \quad (2.10)$$

$$Iy \geq 0 \quad : \quad \tilde{v} \quad (2.11)$$

$$b^T y - c^T x \geq 0 \quad : \quad \theta \quad (2.12)$$

We first show that (2.8) through (2.12) is a feasible system from which it follows by the Weak Duality Theorem 2.3 that strong duality holds. Assume, on the contrary, that (2.8) through (2.12) is an infeasible system. In general, if $M s \geq d$

is an infeasible system, then, by the Infeasibility Theorem, there exist infeasibility multipliers $\pi \geq 0$ such that $(\pi^T M)s \geq \pi^T d$ is an infeasibility inequality, i.e., one where $\pi^T M = 0$ and $\pi^T d > 0$. These multipliers are not unique; because of the homogeneity, π may be replaced by any scalar multiple $\lambda\pi \geq 0$, where $\lambda \geq 0$.

Let the infeasibility multipliers be $\tilde{y}^T \geq 0$, $\tilde{u}^T \geq 0$, $\tilde{x}^T \geq 0$, $\tilde{v}^T \geq 0$, and $\theta = 1$ for (2.8) through (2.12), respectively, where we assume that the multipliers have been rescaled so that $\theta = 1$. Note that it must be true that the scalar $\theta > 0$ because $\theta = 0$ would imply that there is no feasible solution to the system (2.8) through (2.11), contrary to the hypothesis that $Ax \geq b$, $x \geq 0$ and $A^T y \leq c$ and $y \geq 0$ are feasible systems.

Applying the infeasibility multipliers on the left of (2.8) through (2.12) and summing, we obtain the relations:

$$-\tilde{x}^T A^T + \tilde{v}^T I + b^T = 0 \quad \text{or} \quad A\tilde{x} \geq b \quad (2.13)$$

$$\tilde{y}^T A + \tilde{u}^T I - c^T = 0 \quad \text{or} \quad \tilde{y}^T A \leq c^T \quad (2.14)$$

$$\tilde{y}^T b - \tilde{x}^T c > 0 \quad \text{or} \quad \tilde{y}^T b > c^T \tilde{x} \quad (2.15)$$

If we multiply (2.13) on the left by \tilde{y}^T and (2.14) on the right by \tilde{x} , we obtain $\tilde{y}^T b \leq \tilde{y}^T A\tilde{x} \leq c^T \tilde{x}$ which contradicts (2.15). Hence we see that (2.8) through (2.12) is always a feasible system.

Since the system (2.8) through (2.12) is feasible, let x, y be any feasible solution satisfying (2.8) through (2.12). Multiplying (2.8) on the left by $y^T \geq 0$ and (2.10) on the left by $x^T \geq 0$, we obtain

$$y^T b \leq y^T Ax = x^T A^T y \leq x^T c = c^T x.$$

Comparing this with $y^T b \geq c^T x$ from (2.12) we conclude that $b^T y = c^T x$. Therefore every feasible solution of (2.8) through (2.12) satisfies the conditions of the Duality Theorem. ■

- ▷ **Exercise 2.3** Show that the proof of the Strong Duality Theorem 2.6 is essentially a proof that there is no separating hyperplane between the inequalities defining the primal feasible region and the dual feasible region when both the primal and dual systems are feasible.

THEOREM 2.7 (Primal/Dual Interchange Theorem) *For every theorem involving primal and dual problems there is an analogous theorem in which the word dual (meaning dual system) is replaced by the word primal (meaning primal system) and the word primal (meaning primal system) is replaced by the word dual (meaning dual system).*

- ▷ **Exercise 2.4** Prove the Theorem 2.7.
- ▷ **Exercise 2.5** Why does Theorem 2.7 not apply to the theorem: “the dual of the dual is the primal.”

2.2 ADDITIONAL THEOREMS ON DUALITY

2.2.1 UNBOUNDEDNESS THEOREM

Definition (Homogeneous Inequalities and Solution): A system of linear inequalities $Ax \geq b$ is *homogeneous* if the right-hand side vector is $b = 0$. A solution $x = x^h$ is called a *homogeneous solution* associated with $Ax \geq b$ if $Ax^h \geq 0$.

A fundamental property of homogeneous solutions of linear systems of inequalities is that any scalar multiple of a homogeneous solution to the system of linear inequalities is a homogeneous solution.

Definition: A homogeneous solution is called *nontrivial* if $x^h \neq 0$.

THEOREM 2.8 (Unboundedness) Consider the primal (2.1) and dual (2.2) systems.

- I If a feasible solution to the primal system exists, but not to the dual, there exists, for the primal, a class of solutions $x = x^* + \lambda x^h$, $z = z^* + \lambda z^h$ such that $Ax^* \geq b$, $x^* \geq 0$, $Ax^h \geq 0$, $x^h \geq 0$, and $z^h = c^T x^h < 0$, such that $z \rightarrow -\infty$ as $\lambda \rightarrow \infty$.
- II If a feasible solution to the dual system exists, but not to the primal, there exists, for the dual, a class of solutions $y = y^* + \lambda y^h$, $v = v^* + \lambda v^h$ such that $A^T y^* \leq c$, $y^* \geq 0$, $A^T y^h \leq 0$, $y^h \geq 0$, and $v^h = b^T y^h > 0$, such that $v \rightarrow \infty$ as $\lambda \rightarrow \infty$.
- III If neither the primal nor dual system has a feasible solution there exist non-negative homogeneous solutions x^h , y^h , to the primal and dual systems such that $z^h = c^T x^h < 0$ and $v^h = b^T y^h > 0$.

Proof.

- I For the dual $Ay^T \leq c$, $y \geq 0$ to be infeasible, there must (by the Infeasibility Theorem) exist multipliers $x^h \geq 0$ such that $Ax^h = 0$, $c^T x^h < 0$. Thus Part I follows.
- II We prove Part II by applying Theorem 2.7 to Part I. We replace the word *primal* with the word *dual*, the word *dual* with the word *primal*, and change the objective so that we are minimizing.
- III To prove Part III we note that if the primal problem is not feasible, we can make it feasible by replacing b with $b = Ae$, where $e = (1, 1, \dots, 1)^T$. Such transformations of the right-hand side have no effect on the feasibility or non-feasibility of the dual. Since the dual system is still infeasible, it follows from Part I that we can find a x^h such that $z^h = c^T x^h < 0$. In an analogous way we can apply Part II to show that we can find a y^h such that $v^h = b^T y^h > 0$.

This completes the proof. ■

▷ **Exercise 2.6** Apply the Simplex Algorithm to the primal problem, assuming the case of a degeneracy rule to resolve degeneracy (see Chapter 5), to prove Part I of Theorem 2.8.

2.2.2 MISCELLANEOUS THEOREMS FOR THE STANDARD FORM

In this section we state theorems for a primal system in standard form:

$$\begin{array}{ll} \text{Minimize} & c^T x = z \\ \text{subject to} & Ax = b, \quad A: m \times n, \\ & x \geq 0, \end{array} \quad (2.16)$$

and its dual:

$$\begin{array}{ll} \text{Maximize} & b^T \pi = v \\ \text{subject to} & A^T \pi \leq c, \quad A: m \times n, \\ & \pi \geq 0. \end{array} \quad (2.17)$$

THEOREM 2.9 (Primal/Dual Optimality Criteria) *Let $(x_1^*, \dots, x_n^*, z^*)$ be a feasible solution to a primal linear program in standard form and $(\pi_1^*, \dots, \pi_m^*, v^*)$ be a feasible solution to its dual, satisfying:*

$$\bar{c}^* = c - A^T \pi^* \geq 0, \quad b^T \pi^* = v^*. \quad (2.18)$$

Then a necessary and sufficient condition for optimality of both solutions is

$$\bar{c}_j^* = 0 \quad \text{for} \quad x_j^* > 0. \quad (2.19)$$

Proof. Let $x_j \geq 0$ be any feasible solution satisfying the primal problem (2.16), and π be any multipliers satisfying the dual problem (2.17). Thus, $\bar{c} = c - A^T \pi \geq 0$. If $Ax = b$ in (2.16) is multiplied on the left by π^T and subtracted from the z -form, we get

$$\bar{c}^T x = z - v. \quad (2.20)$$

By the sufficiency hypothesis, there is a particular feasible solution $x_j = x_j^* \geq 0$, $z = z^*$, and particular multipliers, $\pi_i = \pi_i^*$ satisfying $\bar{c}_j^* \geq 0$, such that $\bar{c}_j^* = 0$, if $x_j^* > 0$. Substituting these values in (2.20), the left-hand side vanishes term by term and $v^* = z^*$. Applying Corollary 2.5 we conclude that $\max v = v^* = z^* = \min z$. This proves the sufficient part of the theorem.

To show the necessary part, by the Duality Theorem, we have $v^* = z^*$. Substituting this into (2.20), the left-hand side must be zero. Since $\bar{c}_j \geq 0$ and $x_j \geq 0$ by hypothesis, the left-hand side is nonnegative term by term, and hence for it to be zero *all terms on the left must vanish*, which means $\bar{c}_j^* = 0$ for $x_j^* > 0$. ■

THEOREM 2.10 (Existence of a Primal Optimum) *If a feasible solution exists for the primal and z has a finite lower bound, an optimal feasible solution exists.*

- ▷ **Exercise 2.7** Prove Theorem 2.10 by proving that the dual is feasible and then applying the Strong Duality Theorem.
- ▷ **Exercise 2.8** Prove that Theorem 2.10 is an immediate consequence of applying the Simplex Method to the primal problem.

COROLLARY 2.11 (Existence of a Dual Optimum) *If an optimal feasible solution exists for the primal, there exists an optimal feasible solution to the dual.*

- ▷ **Exercise 2.9** Show that Corollary 2.11 is a special case of Theorem 2.10.
- ▷ **Exercise 2.10** Prove Corollary 2.11 by showing that the terminal conditions of the Simplex Method generate an optimal feasible solution to the dual.

2.3 COMPLEMENTARY SLACKNESS

When the primal and dual systems are expressed in von Neumann symmetric form, as systems of inequalities in nonnegative variables, Theorem 2.9 takes on a more elegant symmetric form.

Let $x_j \geq 0$ be any feasible solution satisfying (2.1) and $y_i \geq 0$ be any feasible solution satisfying (2.2); we assume here that feasible solutions exist. We rewrite the former in standard equality form by substituting a vector of slack variables x_s :

$$\begin{aligned} \text{Minimize} \quad & c^T x & = z \\ \text{subject to} \quad & Ax - Ix_s = b \\ & x & \geq 0 \end{aligned} \tag{2.21}$$

where $x_s = (x_{n+1}, x_{n+2}, \dots, x_{n+m})^T \geq 0$ are variables that measure the extent of inequality, or *negative slack*, between the left and right-hand sides of the inequalities.

It will be convenient to let $y_s = (y_{m+1}, y_{m+2}, \dots, y_{m+n}) \geq 0$ measure the *positive slack* in the inequalities of the dual system. Then (2.2) in standard equality form becomes:

$$\begin{aligned} \text{Maximize} \quad & b^T y & = v \\ \text{subject to} \quad & A^T y + Iy_s = c \\ & y & \geq 0 \end{aligned} \tag{2.22}$$

where $y_s = (y_{m+1}, y_{m+2}, \dots, y_{m+n})^T \geq 0$.

Let (x, x_s, z) be any feasible solution to the primal system (2.21) and let (y, y_s, v) be any feasible solution to the dual system (2.22). Multiplying $Ax - Ix_s = b$ on the left by y^T and subtracting from the z form in (2.21) we get

$$c^T x - y^T A x + y^T x_s = z - y^T b. \quad (2.23)$$

Multiplying $A^T y + I y_s = c$ on the left by x^T and subtracting the v form of equation (2.22) from it, we get

$$-b^T y + x^T A^T y + x^T y_s = x^T c - v. \quad (2.24)$$

Adding (2.23) and (2.24) and cancelling we obtain

$$y_s^T x + y^T x_s = z - v. \quad (2.25)$$

The left-hand side of (2.25) is nonnegative term by term, hence $0 \leq z - v$ or $v \leq z$.

Since we are assuming that primal and dual solutions exist, the hypothesis of the Duality Theorem is satisfied and there exist optimal feasible solutions, $(x, x_s, z) = (x^*, x_s^*, z^*)$ and $(y, y_s, v) = (y^*, y_s^*, v^*)$, to both systems with $z^* = v^*$. Hence

$$(y_s^*)^T x^* + (y^*)^T x_s^* = 0. \quad (2.26)$$

Since $(x^*, x_s^*) \geq 0$ and $(y^*, y_s^*) \geq 0$, the left-hand side of (2.26) vanishes term by term:

$$\begin{aligned} [y_s^*]_j x_j^* &= 0 & \text{for } j = 1, \dots, n \\ y_i^* [x_s^*]_i &= 0 & \text{for } i = 1, \dots, m \end{aligned}$$

establishing the following theorem.

THEOREM 2.12 (Complementary Slackness) *For optimal feasible solutions of the primal (2.1) and dual (2.2) systems, whenever the k th relation of either system is slack, the k th variable of its dual is zero; if the k th variable is positive in either system, the k th relation of its dual is tight, i.e.,*

$$x_k y_{m+k} = 0 \quad k = 1, \dots, n \quad \text{and} \quad y_k x_{m+k} = 0 \quad k = 1, \dots, m. \quad (2.27)$$

Comment: By a “slack constraint” we mean that the value of the slack variable in the optimum solution is positive. By a “tight constraint” we mean that the value of the slack variable is zero.

Comment: The primal variable x and dual slacks y_s (similarly, the dual variables y and primal slacks x_s) are called *complementary variables*.

2.4 THEOREMS OF THE ALTERNATIVES

Although we state the various theorems of the alternatives using the term *dual linear program*, these theorems (except Tucker’s) predate 1947–1948 when linear programming was formulated and the term *dual* was first used. Instead the authors of these theorems referred to the dual system in homogeneous form as adjoint or transpose systems.

2.4.1 GORDAN'S THEOREM

THEOREM 2.13 (Gordan [1873]) *Either a linear homogeneous system of equations $Ax = 0$ possesses a nontrivial solution in nonnegative variables or there exists an equation, formed by taking some linear combination of the equations, that has all positive coefficients. That is, either there exists an $x = x^o$ such that*

$$Ax^o = 0, \quad 0 \neq x^o \geq 0 \quad (2.28)$$

or there exists a π such that

$$\pi^T A > 0. \quad (2.29)$$

Proof. If (2.28) has a solution, then so does

$$\begin{pmatrix} A \\ e^T \end{pmatrix} x = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad x \geq 0, \quad (2.30)$$

where $e = (1, 1, \dots, 1)^T$. If (2.30) has no solution, then by Corollary 2.2 of the Infeasibility Theorem, there exist $(\tilde{\pi}, \eta)$ such that

$$\tilde{\pi}^T A + \eta e^T = 0, \quad \tilde{\pi}^T 0 + \eta > 0. \quad (2.31)$$

Since $\eta > 0$ by (2.31), this implies

$$\tilde{\pi}^T A = -\eta e^T < 0. \quad (2.32)$$

Substituting $\pi = -\tilde{\pi}$, we obtain $\pi^T A > 0$. ■

Example 2.1 (Illustration of Gordan's Theorem) The system

$$\begin{aligned} 2x_1 - 3x_2 &= 0 \\ 3x_1 - 2x_2 &= 0 \end{aligned}$$

has only a trivial solution $x_1 = 0$, and $x_2 = 0$. Therefore, according to Gordan's theorem, there must exist a π such that $\pi^T A > 0$. It is easy to verify that one such π is $\pi_1 = -1$, $\pi_2 = 1$. On the other hand, the system

$$\begin{aligned} x_1 - 2x_2 + x_3 &= 0 \\ x_2 - x_3 &= 0 \end{aligned}$$

has the nontrivial solution $x_1 = 1$, $x_2 = 1$, $x_3 = 1$, implying, according to Gordan's theorem, that $\pi^T A > 0$ results in an infeasible system. It is easy to verify that this is indeed the case:

$$\begin{aligned} \pi_1 &> 0 \\ -2\pi_1 + \pi_2 &> 0 \\ \pi_1 - \pi_2 &> 0. \end{aligned}$$

▷ **Exercise 2.11** Prove the converse of Gordan's Theorem, namely, if (2.29) is true then this implies that the only feasible solution to $Ax = 0$, $x \geq 0$ is $x = 0$.

2.4.2 FARKAS'S LEMMA

THEOREM 2.14 (Farkas's Lemma [1902]) *If a linear homogeneous inequality*

$$b^T \pi \leq 0 \quad (2.33)$$

holds for every π satisfying a system of homogeneous inequalities

$$A^T \pi \leq 0, \quad (2.34)$$

then the inequality $b^T \pi \leq 0$ is a nonnegative linear combination $x \geq 0$ of the inequalities of the system $A^T \pi \leq 0$, that is,

$$Ax = b, \quad x \geq 0. \quad (2.35)$$

Proof. Assume the hypothesis is true but, on the contrary assume, that there exists no nonnegative linear combination x of (2.34) that yields (2.33). Then there exists no feasible solution to the system (2.35). By Corollary 2.2 of the Infeasibility Theorem, there exist multipliers $\pi_i = \pi_i^0$, that, when applied to (2.35), yield an infeasible equation; the coefficients and right hand side of this equation are

$$\begin{aligned} A^T \pi^0 &\leq 0, \\ b^T \pi^0 &> 0, \end{aligned}$$

contrary to our hypothesis. ■

Example 2.2 (Illustration of Farkas's Lemma) Note by adding the system of inequalities

$$\begin{aligned} \pi_1 + 2\pi_2 &\leq 0 \\ \pi_1 + \pi_2 &\leq 0 \end{aligned}$$

the linear homogeneous inequality

$$2\pi_1 + 3\pi_2 \leq 0$$

holds for every π satisfying the inequalities. It is easy to verify the statement of Farkas's lemma that there exists $x \geq 0$ satisfying

$$\begin{aligned} x_1 + x_2 &= 2 \\ 2x_1 + x_2 &= 3, \end{aligned}$$

namely $x_1 = 1$ and $x_2 = 1$.

▷ **Exercise 2.12** Prove that the following is an equivalent statement of Farkas's Lemma. Either:

1. there exists an $x \geq 0$ for which $Ax = b$,
2. or there exists a π for which $A^T \pi \geq 0$ and $b^T \pi < 0$.

- ▷ **Exercise 2.13** State and prove the analogue of Farkas's Lemma for linear equation systems.
- ▷ **Exercise 2.14** Show that Farkas's Lemma implies the Infeasibility Theorem 2.1.
- ▷ **Exercise 2.15** Apply the FME algorithm (*Linear Programming 1*) to prove Farkas's Lemma.

2.4.3 STIEMKE'S THEOREM

THEOREM 2.15 (Stiemke [1915]) *Either a linear homogeneous system $Ax = 0$ possesses a solution with all variables positive or there exists a linear combination of the equations that has all nonnegative coefficients, one or more of which are positive. That is, either there exists an $x = x^o$ such that*

$$Ax^o = 0, \quad x^o > 0, \quad (2.36)$$

or there exists a π such that

$$0 \neq A^T\pi \geq 0. \quad (2.37)$$

Proof. If the homogeneous system possesses a strictly positive solution, there exists, by positive rescaling, a solution to the system

$$\begin{aligned} Ax &= 0 \\ x &\geq e, \quad \text{where } e = (1, 1, \dots, 1)^T. \end{aligned} \quad (2.38)$$

Replacing x with $x' + e$, where $x' \geq 0$, results in the system

$$\begin{aligned} Ax' &= -Ae \\ x' &\geq 0 \end{aligned} \quad (2.39)$$

Either this system possesses a feasible solution or there exist, by Corollary 2.2 of the Infeasibility Theorem, multipliers π such that the resulting linear combination

$$(\pi^T A)x' = -(\pi^T A)e \quad (2.40)$$

is an infeasible equation in nonnegative variables. That is, $\pi^T A \geq 0$ and $-\pi^T A e < 0$; but $\pi^T A e > 0$ implies that at least one component of $\pi^T A$ is > 0 establishing the theorem. ■

Example 2.3 (Illustration of Stiemke's Theorem) The linear homogeneous system

$$\begin{aligned} x_1 - x_2 &= 0 \\ 2x_1 - x_2 &= 0 \\ x_1 > 0, x_2 > 0 \end{aligned}$$

		Primal								
Dual	Variables	$x_1 \geq 0$	\cdots	$x_k \geq 0$	$x_{k+1} \geq 0$	\cdots	$x_n \geq 0$	Rel	Const	
	π_1	a_{11}	\cdots	a_{1k}	a_{1k+1}	\cdots	a_{1n}	=	0	
	π_2	a_{21}	\cdots	a_{2k}	a_{2k+1}	\cdots	a_{2n}	=	0	
	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	
	π_m	a_{m1}	\cdots	a_{mk}	a_{mk+1}	\cdots	a_{mn}	=	0	
	Relation	\leq	\cdots	\leq	\leq	\cdots	\leq			
Constants	0	\cdots	0	0	\cdots	0				

Table 2-1: Tucker Diagram (Partitioned)

has no solution. Therefore, according to Stiemke’s Theorem, there must exist a π such that $0 \neq \pi^T A \geq 0$. It is easy to verify that one such π is $\pi_1 = -1, \pi_2 = 1$, resulting in

$$A^T \pi = \begin{pmatrix} 1 & 2 \\ -1 & -1 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

On the other hand, the system

$$\begin{aligned} x_1 - 2x_2 + x_3 &= 0 \\ x_2 - x_3 &= 0 \\ x_1 > 0, x_2 > 0, x_3 > 0 \end{aligned}$$

has a solution $x_1 = 1, x_2 = 1, x_3 = 1$, implying, according to Stkemke’s theorem, that $0 \neq \pi^T A \geq 0$ does not hold. It is easy to verify that this is indeed the case because

$$\begin{aligned} \pi_1 &\geq 0 \\ -2\pi_1 + \pi_2 &\geq 0 \\ \pi_1 - \pi_2 &\geq 0 \end{aligned}$$

implies $\pi_1 = 0, \pi_2 = 0$, or $A^T \pi = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

2.4.4 MOTZKIN’S TRANSPOSITION THEOREM

The pair of homogeneous systems $Ax = 0, x \geq 0$, and $A^T \pi \leq 0$ may be viewed as dual linear programs with zero-coefficient objectives $0^T x = \min$ and $0^T \pi = \max$. These are displayed in the Tucker Diagram shown in Table 2-1. We assume each column has at least one nonzero coefficient.

THEOREM 2.16 (Motzkin [1936]) *Consider any arbitrary subset of k columns; for example the first k columns shown as a partition in the diagram (see Table 2-1) to the left of the vertical double line. Either there exists a solution to the dual system $A^T \pi \leq 0$, such that all inequalities corresponding to the subset hold strictly, or the primal system $Ax = 0, x \geq 0$ has a solution such that at least one corresponding variable of the subset has a positive value.*

Proof. If there exists a solution to the primal system with the requisite property, then one exists such that

$$x_1 + x_2 + \cdots + x_k = 1 \quad (2.41)$$

where $j = 1, \dots, k$ is the assumed subset. The remainder of the proof parallels the proof of Gordan's Theorem (See Theorem 2.13). ■

▷ **Exercise 2.16** Complete the proof of the Motzkin Transposition Theorem.

Example 2.4 (Illustration of Motzkin's Theorem) Consider the system:

$$\begin{aligned} x_1 + x_2 - x_3 &= 0 \\ x_2 &= 0. \end{aligned}$$

The dual system satisfies:

$$\begin{aligned} \pi_1 &\leq 0 \\ \pi_1 + \pi_2 &\leq 0 \\ -\pi_1 &\leq 0. \end{aligned}$$

The dual clearly implies that $\pi_1 = 0$ and π_2 can take on any nonpositive value.

If we choose as the subset the variables x_1 and x_2 , the first inequality in the dual corresponding to this subset cannot hold strictly because $\pi_1 = 0$ as we have just shown. Hence, according to Motzkin's theorem there must exist $x \geq 0$ such that $x_1 + x_2 > 0$. This is clearly true, because one solution is $x_1 = 1, x_2 = 0, x_3 = 1$.

On the other hand, if we choose as the subset only the variable x_2 , the dual inequality corresponding to this subset can be made strict by choosing $\pi_2 < 0$. In this case, according to Motzkin's theorem, $x_2 = 0$, which is indeed the case.

COROLLARY 2.17 (Complementary Pair) Consider the von Neumann primal/dual pair of homogeneous systems $Ax \geq 0, x \geq 0$ and $A^T y \leq 0, y \geq 0$ with zero-coefficient objectives, where A is $m \times n$. If we subtract slack variables $u \geq 0$ in the primal to obtain $Ax - Iv = 0, x \geq 0$, then given an index p there exists a pair of complementary solutions (v^p, y^p) such that either

$$v_p^p > 0, y_p^p = 0,$$

or

$$v_p^p = 0, y_p^p > 0.$$

▷ **Exercise 2.17** Prove Corollary 2.17.

2.4.5 VILLE'S THEOREM

THEOREM 2.18 (Ville [1938]) Consider the dual homogeneous programs with all zero-coefficient objective forms,

$$Ax \geq 0, \quad x \geq 0, \quad (2.42)$$

$$A^T y \leq 0, \quad y \geq 0. \quad (2.43)$$

Let either system be the primal and the other the dual. Either there exists a solution to the primal where all inequalities hold strictly or there exists a nontrivial solution to the dual.

Example 2.5 (Illustration of Ville's Theorem) Consider the primal and dual homogeneous linear programs with zero-coefficient objectives:

$$\begin{array}{lcl} x_1 + x_2 \geq 0 & & y_1 \leq 0 \\ x_1 \geq 0, x_2 \geq 0 & \text{and} & y_1 \geq 0. \end{array}$$

Clearly, there exists a solution that satisfies the primal problem with all inequalities holding strictly, namely, $x_1 = 1, x_2 = 1$. Then by Ville's theorem the dual has only a trivial solution, i.e., $y_1 = 0$, which is indeed the case. On the other hand, consider the primal and dual homogeneous linear programs with zero-coefficient objectives:

$$\begin{array}{lcl} x_1 + x_2 \geq 0 & & y_1 \leq 0 \\ -x_2 \geq 0 & \text{and} & y_1 - y_2 \leq 0 \\ x_1 \geq 0, x_2 \geq 0 & & y_1 \geq 0, y_2 \geq 0. \end{array}$$

The second inequality in the primal cannot hold strictly because clearly $x_2 = 0$. Then by Ville's theorem we must have a nontrivial solution to the dual, which is the case because, for example, $y_1 = 0, y_2 = 1$, satisfies the dual.

▷ **Exercise 2.18** Show that Ville's Theorem 2.18 is a special case of the Motzkin's Transposition Theorem 2.16 by introducing slack variables into the primal system.

2.4.6 TUCKER'S STRICT COMPLEMENTARY SLACKNESS THEOREM

Sharper forms of the various Theorems of Alternatives can be obtained by judicious application of Motzkin's Transposition Theorem; in particular Tucker's Theorem 2.21.

Definition (Complementary Pair): Let $Ax - Iv = 0, x \geq 0, v \geq 0$ and $A^T y + Iu = 0, y \geq 0, u \geq 0$ be a pair of homogeneous primal/dual linear programs with all zero coefficient objectives and slacks added. The corresponding (x_j, u_j) and (y_i, v_i) are called *complementary pairs*.

LEMMA 2.19 (Complementary Slackness for Homogeneous Case) *Every solution to the homogeneous primal (2.42) and dual (2.43) systems is optimal and the products of all complementary pairs vanish.*

LEMMA 2.20 (Combining Solutions) *If (x^0, y^0) and (x^1, y^1) are two pairs of feasible solutions to (2.42) and (2.43) then $(x^0 + x^1, y^0 + y^1)$ are also feasible solutions to (2.42) and (2.43) and satisfy the complementary slackness property.*

▷ **Exercise 2.19** Prove Lemmas 2.19 and 2.20.

THEOREM 2.21 (Tucker [1956]) *There exist solutions to the homogeneous primal (2.42) and dual (2.43) programs that have all zero coefficient objectives such that every variable in one system and its complementary slack in the other system have one positive and one zero value.*

Comment: Tucker's Theorem is also known as the *Strict Complementary Slackness Theorem* because it states that optimal solutions can be found such that in every complementary pair exactly one variable is positive and the other variable is zero.

Proof. We prove the theorem by demonstrating strict complementary slackness for the pairs (y, v) . Then, by interchanging the role of the primal and dual in the proof, we can find a solution that also satisfies strict complementary slackness for the pairs (x, u) .

Augment the primal system $Ax \geq 0, x \geq 0$ with slack variables $v \geq 0$ to obtain the system

$$Ax - Iv = 0, \quad x \geq 0, \quad v \geq 0$$

and add slack variables $u \geq 0$ to the dual to obtain

$$A^T y + Iu = 0, \quad y \geq 0, \quad u \geq 0.$$

Find any feasible solution $(x, v) = (\bar{x}^o, \bar{v}^o)$ to the primal and any feasible solution $y = \bar{y}^o$ to its dual; by Lemma 2.19 these are optimal and satisfy complementary slackness.

If strict complementary slackness holds for each pair $(\bar{y}_i^o, \bar{v}_i^o)$ for $i = 1, \dots, m$, we are done. Otherwise, we find the first index $i = p$ for which $\bar{v}_p^o = 0$ and $\bar{y}_p^o = 0$. Next, partition the primal system so that Motzkin's subset (see Section 2.4.4) consists of one slack variable, v_p , then by Corollary 2.17, either there exists a solution in which $v_p > 0$ for the primal system or $y_p > 0$ for the dual. In the first case find a primal solution $(x, v) = (x^p, v^p)$ with $v_p^p > 0$ and set $(y, u) = (y^p, u^p) = (0, 0)$. In the second case find a dual solution $y = y^p, u = u^p = 0$ with $y_p^p > 0$ and set $(x, v) = (x^p, v^p) = (0, 0)$. Next add the current two solutions to obtain:

$$\begin{aligned}\bar{x}^1 &= \bar{x}^o + x^p \\ \bar{v}^1 &= \bar{v}^o + v^p \\ \bar{y}^1 &= \bar{y}^o + y^p \\ \bar{u}^1 &= \bar{u}^o + u^p\end{aligned}$$

which is also a feasible solution to the primal/dual system by Lemma 2.20. By Lemma 2.19 (\bar{x}^1, \bar{v}^1) and (\bar{y}^1, \bar{u}^1) are optimal and also satisfy complementary slackness. We know that none of the values (\bar{x}^o, \bar{v}^o) and (\bar{y}^o) have decreased because we have added nonnegative numbers. Hence, if $p < m$, we have examined the solution (\bar{x}^1, \bar{v}^1) and (\bar{y}^1, \bar{u}^1) from components $i = p + 1, \dots, m$ to find if any pair of \bar{y}_i^1, \bar{v}_i^1 fails to meet strict complementary slackness. If any pair fails, repeat the process iteratively until a solution $(\bar{x}^k, \bar{v}^k), (\bar{y}^k, \bar{u}^k)$ is obtained where all pairs \bar{y}_i^k, \bar{v}_i^k satisfy strict complementary slackness. ■

Example 2.6 (Illustration of Strict Complementary Slackness) Consider the primal and dual homogeneous programs with zero-coefficient objectives:

$$\begin{array}{rcl} x_1 + x_2 & \geq & 0 \\ -x_1 & \geq & 0 \\ x_1 \geq 0, x_2 \geq 0 & & \end{array} \quad \text{and} \quad \begin{array}{rcl} y_1 - y_2 & \leq & 0 \\ y_1 & \leq & 0 \\ y_1 \geq 0, y_2 \geq 0. & & \end{array}$$

A solution to the primal is

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

and a solution to the dual is

$$y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

It is easy to verify Tucker's strict complementary slackness.

▷ **Exercise 2.20** Consider the primal problem (with vacuous objective)

$$\begin{array}{rcl} x_1 + x_2 + x_3 & \geq & 0 \\ x_1 & - & x_3 \geq 0 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. & & \end{array}$$

Write down its dual and show there exists a nontrivial primal solution that does not satisfy strict complementary slackness. Show how to modify your solution, by adding to it one or more solutions according to the proof of Tucker's theorem, to obtain one that satisfies strict complementary slackness.

▷ **Exercise 2.21** Consider the primal homogeneous system $Ax \geq 0$, $x \geq 0$ with all zero coefficient objective and A given by

$$A = \begin{pmatrix} 1.0 & -0.5 & -0.6 \\ -0.1 & 1.0 & -0.1 \\ -0.3 & -0.2 & 1.0 \end{pmatrix}.$$

What does Tucker's Strict Complementary Slackness Theorem say about its dual?

▷ **Exercise 2.22** For the primal (2.42) and dual (2.43) systems show how to obtain a strictly complementary solution by formulating and solving one linear program.

2.5 NOTES & SELECTED BIBLIOGRAPHY

As noted in this chapter, associated with every linear programming problem is another linear programming problem called the *dual*. The fundamental notion of duality and the term was introduced by John von Neumann (in conversations with George Dantzig in October 1947) and appears implicitly in a working paper he wrote a few weeks later (von

Neumann, [1947]). George Dantzig's report, *A Theorem on Linear Inequalities*, dated January 5, 1948, contains the first known rigorous (unpublished) proof of the Duality Theorem. Subsequently Gale, Kuhn, & Tucker [1951] independently formulated the Duality Theorem which they proved by means of a classical lemma due to Farkas [1902]. This theorem (see Theorem 2.14), known as *Farkas' Lemma*, first appeared as a lemma in Farkas's 1902 paper. A constructive proof of the Duality Theorem using the Simplex Method can be found in Dantzig [1963]. J. Abadie in verbal communications [1965] with one of the authors showed how to use the Infeasibility Theorem to prove von Neumann's Strong Duality Theorem. Our proof is a more concise version of Abadie's.

Tobias Dantzig, mathematician and author, well known for his books popularizing the history of mathematics, suggested around 1955, to his son George, the term *primal* as the natural antonym to dual since both primal and dual derive from the Latin.

A systematic presentation of theoretical properties of dual linear programs can be found in Gale [1951] and Goldman & Tucker [1956a,b]. A review of von Neumann's contributions can be found in Kuhn & Tucker [1958]. Today everyone cites von Neumann as the originator of the Duality Theorem and credits Gale, Kuhn, & Tucker as the publishers of the first rigorous proof.

As already noted, there are several important duality-type results, known as "Either Or" theorems of the alternatives, that predated the linear programming era: Farkas [1902], Gordan [1873], Motzkin [1936], Stiemke [1915], and Ville [1938]. The earliest known result on feasibility is one concerning *homogeneous systems*, Gordan [1873]. Tucker [1956] presented a sharper form of the Theorem of Alternatives as presented in this chapter.

A natural question to ask is why not use the classical method of Lagrange multipliers to solve the linear programming problem. If we were to do so we would be required to find optimal multipliers (or prices π), which, if they exist, must satisfy a "dual" system with the property that the \bar{c}_j (or relative cost factors) and optimal x_j satisfy $\bar{c}_j x_j = 0$ for $j = 1, \dots, n$. The latter leads to 2^n possible cases of either $\bar{c}_j = 0$ or $x_j = 0$. It is here that this classical approach breaks down, for it is not practical to consider all 2^n possible cases for large n . In a certain sense, however, the Simplex Method can be viewed as a systematic way to eliminate most of these cases and to consider only a few. Indeed, it immediately restricts the number of cases by considering only those with $n-m$ of the $x_j = 0$ at one time and such that the coefficient matrix of the remaining m variables is nonsingular; moreover the unique value of these variables is positive (under nondegeneracy). The conditions $\bar{c}_j x_j = 0$ tell us that $\bar{c}_j = 0$ for $x_j > 0$ and this determines uniquely π_i and the remaining \bar{c}_j . If it turns out that not all $\bar{c}_j \geq 0$, the case is dropped and a special new one is examined on the next iteration, and so on.

2.6 PROBLEMS

- 2.1 Assuming Farkas's Lemma is true, derive the Duality Theorem.
- 2.2 *Ph.D. Comprehensive Exam, September 21, 1991, at Stanford.* Given the linear program, find $x_j \geq 0$, $\min z$, satisfying, in detached coefficients

$-z$	x_1	x_2	x_3	x_4	x_5	x_6	b
1	4	3	2	1	0	-1	0
	1	-.5	-.4	1			2
	-.2	1	-.5		1		3
	-.3	-.2	1			1	4

(a) Prove every basic feasible solution satisfies

$$x_1x_4 = 0, \quad x_2x_5 = 0, \quad x_3x_6 = 0.$$

(b) Prove that the set of feasible x is bounded.

2.3 In this exercise we examine a different proof of the Duality Theorem 2.6 which also uses the Infeasibility Theorem 2.1. Refer to the von Neumann symmetric form (2.1) and (2.2).

(a) Consider the primal system

$$\begin{aligned} Ax &\geq b \\ Ix &\geq 0 \\ -c^T x &\geq -\sup v \quad (\text{i.e., } c^T x \leq \sup v \leq \inf z) \end{aligned}$$

where $v = b^T y$, $A^T y \leq c$, and $y \geq 0$. Show in a detailed step-by-step way that assuming this system is infeasible leads to a contradiction.

(b) Consider in an analogous way the dual system

$$\begin{aligned} y^T A &\leq c^T \\ -y^T I &\leq 0 \\ -y^T b &\leq -\inf z \quad (\text{i.e., } y^T b \geq \inf z \geq \sup v) \end{aligned}$$

where $z = c^T x$, $Ax = b$, and $x \geq 0$. Show in an analogous detailed step-by-step way that assuming this system is infeasible leads to a contradiction

(c) Conclude from (a) and (b) that the Duality Theorem 2.6 is true.

(d) Redo the proof of part (b) by viewing the dual system as a primal problem and applying the conclusions that we have already arrived at for the primal problem. Therefore show that is not necessary to go through a detailed step by step proof to arrive at an analogous conclusion for the dual.

2.4 *Ph.D. Comprehensive Exam, September 21, 1991, at Stanford.* Consider the two linear programs (i) $0^T x = \min z$ subject to $Ax \geq 0$, $x \geq 0$, and (ii) $0^T y = \max v$ subject to $A^T y \leq 0$, $y \geq 0$.

(a) Prove that either program is the dual of the other.

(b) Prove that either there exists an $x \geq 0$ such that $Ax > 0$ or there exists a nontrivial solution to the dual linear program.

2.5 Consider the problem:

$$\begin{aligned} \text{Minimize} \quad & c^T x + g^T y = z \\ \text{subject to} \quad & Ax = b \\ & -Bx + Fy = d \\ & x \geq 0 \\ & y \geq 0. \end{aligned}$$

We are given a point $x = x^o$ that satisfies $Ax^o = b$, $x^o \geq 0$. We wish to determine whether there exists a $y = y^*$ such that (x^o, y^*) is optimal for the full problem. Let π_1 and π_2 be the multipliers on $Ax = b$ and $-Bx + Fy = d$ respectively. We perform a number of checks:

- (a) $Ax^o = b$, $x^o \geq 0$ is satisfied. However, we note that x^o is not a basic feasible solution.
- (b) Solve $\min g^T y$ subject to $Fy = d + Bx^o$, $y^o \geq 0$. The optimal solution is $y = y^*$ with optimal multipliers $\pi_2 = \pi_2^*$.
- (c) Solve $\min (c + B^T \pi_2^*)^T x$ subject to $Ax = b$. The optimal solution is $x = x^*$ with optimal multipliers $\pi_1 = \pi_1^*$.

Prove that (x^o, y^*) is an optimal solution to the original system and (π_1^*, π_2^*) is the optimal dual solution if

$$(c + B^T \pi_2^*)^T x^o = b^T \pi_1^*.$$

- 2.6 *Dantzig [1963]*. The Fourier-Motzkin Elimination method permits one to drop a variable by increasing the number of inequalities. Dualize the procedure and find a method for decreasing the number of inequalities by increasing the number of variables.
- 2.7 Suppose there exists a solution to a homogeneous system of inequalities, $Ax \geq 0$, each of which is satisfied strictly. Show that there exists a solution to $Ax \geq e$, where $e = (1, 1, \dots, 1)^T$.
- 2.8 Consider the linear program:

$$\begin{aligned} & \text{Minimize} && c^T x \\ & \text{subject to} && Ax \geq b, && A: m \times n, \\ & && x \geq 0. \end{aligned} \tag{2.44}$$

Suppose that $x = x^*$ is a basic *feasible optimal solution* for this program and that $x_B^* > 0$, i.e., the optimal basic feasible solution is nondegenerate. Show that the dual to this linear program has a *unique* optimal solution.

- 2.9 Show that the set $\{x \mid Ax \geq b\} \neq \emptyset$ is unbounded if and only if there exists an $x \neq 0$ such that $Ax \geq 0$.
- 2.10 Consider von Neumann's primal-dual pair of LPs, (2.1) and (2.2). Show that it is impossible for the primal's feasible region (set of feasible solutions) and the dual's feasible region to be both nonempty and bounded.
- 2.11 Devise an efficient way to test that a given solution $x = x^*$ is an optimal solution for a linear program in standard form by considering the following cases:
 - (a) $x_k > 0$, for $k = 1, \dots, m + l$, where $0 < l \leq n - m$; if columns $1, \dots, m$ are nonsingular and also if they are singular;
 - (b) $x_k > 0$, for $k = 1, \dots, m - l$, where $0 < l < m$.

Clearly explain your approach.

- 2.12 Show that the primal has a unique solution if the dual is nondegenerate and that the dual has a unique solution if the primal is nondegenerate. Note that the dual $\max b^T \pi$, subject to $A^T \pi \leq c$, of a linear program in standard form is defined to be nondegenerate if for any π , with $A^T \pi \leq c$, we have $c - A^T \pi$ with at most m zeros.

2.13 *Dantzig [1963].*

- (a) Suppose that an optimal solution with respect to a given objective form z is not unique and that it is desired to introduce an alternative objective \hat{z} and to minimize \hat{z} , given that z is minimum. Show that an optimal solution exists that is basic in the restraint system, excluding the z and \hat{z} forms. Prove that this solution can be obtained by first dropping all variables x_i , such that $\bar{c}_j > 0$ at the end of Phase II, and then replacing the z form by the \hat{z} form.
- (b) Generalize the usual Phase I, Phase II procedure to find a solution that is as “feasible as possible” ($\min w$) and given that it is and is not unique, find the one that minimizes z , given that $w = \min w$.

2.14 *Dantzig [1963].*

- (a) Show that it is not possible for $z \rightarrow -\infty$, if no positive combination of activities vanishes. Discuss what this means in a practical situation if a positive combination vanishes except for a positive reduced cost, a negative reduced cost, a zero reduced cost.
- (b) Show that if $z \rightarrow -\infty$, there exists a homogeneous feasible solution to the system. Show that it is possible to have $z \rightarrow +\infty$ and $z \rightarrow -\infty$ in the same system.
- (c) Does a column with all negative entries in the original tableau imply that (if feasible solutions exist) a class of solutions exists such that $z \rightarrow -\infty$?

2.15 *Dantzig [1963].* Prove that if an optimal solution $x_j^o \geq 0$, $z = z^o = \min z$ exists to a linear program, then the system of equations formed by dropping all x_j , such that $x_j^o = 0$ and setting $z = z^o$ is redundant.

2.16 *Ph.D. Comprehensive Exam, June 15, 1967, at Stanford.* Given the system

$$\begin{aligned} \sum_{j=1}^n y_j &= 1, \\ \sum_{j=1}^n a_{ij}y_j &\leq M \quad \text{for } i = 1, \dots, m, \\ y_j &\geq 0 \quad \text{for } j = 1, \dots, n. \end{aligned} \tag{2.45}$$

Systems such as these have application in “game” theory.

- (a) Show that the problem of finding the minimum M satisfying (2.45) is a linear program.
- (b) Show that the dual of the linear program in part (a) is of the form

$$\begin{aligned} &\text{Maximize} && N, \\ &\text{subject to} && \sum_{i=1}^m x_i = 1 \\ & && \sum_{i=1}^m a_{ij}x_i \geq N \quad \text{for } j = 1, \dots, n \\ & && x_i \geq 0 \quad \text{for } i = 1, \dots, m. \end{aligned} \tag{2.46}$$

- (c) Prove

$$N \leq \sum_{i=1}^m \sum_{j=1}^n x_i a_{ij} y_j \leq M$$

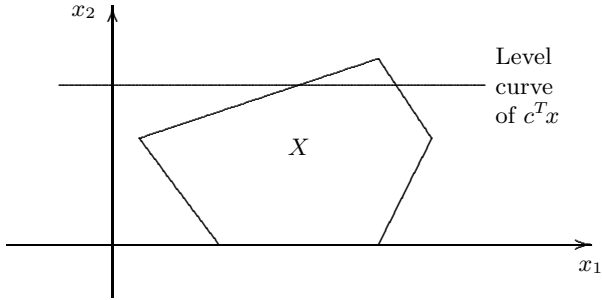


Figure 2-2: Find Basic Feasible Solutions of Dual of Two Variable Primal

and

$$\max N = \min M.$$

- (d) Show that primal and dual feasible solutions for the linear programs (2.45) and (2.46) always exist.
- (e) Why is $\max N = \min M$ positive, if all $a_{ij} > 0$?
- 2.17 *Ph.D. Comprehensive Exam, June 15, 1967, at Stanford.* We know that linear programming problems whose variables have lower and upper bounds permit a special variant of the Simplex Method.
- (a) State the variant
- (b) Considering a problem of this type as primal, state the associated dual problem.
- 2.18 *Ph.D. Comprehensive Exam, March 31, 1969, at Stanford.* Let

$$\mathcal{S} = \{x \mid Ax \geq b, x \geq 0\} \quad \text{and} \quad \mathcal{T} = \{y \mid A^T y \leq c, y \geq 0\}$$

- be the set of feasible solutions of primal and dual linear programs. Prove that if \mathcal{S} and \mathcal{T} are nonempty, then at least one of them must be unbounded. Could both of them be unbounded?
- 2.19 *Ph.D. Comprehensive Exam, September 25, 1971, at Stanford.* Consider a two-variable linear programming problem of the form

$$\begin{aligned} &\text{Maximize} && c^T x = z \\ &\text{subject to} && Ax \leq b, \\ &&& x \geq 0, \end{aligned}$$

- for which the constraint set X and the objective function are as indicated in Figure 2-2. Use the figure to determine the number of basic feasible solutions of the dual problem.
- 2.20 *Ph.D. Comprehensive Exam, September 27, 1975, at Stanford.* Von Neumann's Minimax Theorem for a two-person zero-sum game (see Problem 2.16) is as follows.

Von Neumann's Minimax Theorem. Given $\sum_{i=1}^m x_i = \sum_{j=1}^n y_j = 1$, $x_i \geq 0$ for $i = 1, \dots, m$ and $y_j \geq 0$ for $j = 1, \dots, n$:

$$\max_x \min_{\{y|x\}} \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_i y_j = \max_y \min_{\{x|y\}} \sum_{i=1}^m \sum_{j=1}^n a_{ij} x_i y_j$$

where $\{y | x\}$ means y given x .

Prove that von Neumann's Minimax Theorem for finite two-person zero-sum games is a special case of the Duality Theorem for linear programs.

2.21 *Ph.D. Comprehensive Exam, September 1979, at Stanford.* Given

$$\begin{aligned} Ax &= b \\ Bx + Gy &= d \end{aligned} \quad (2.47)$$

with $x \geq 0$ and $y \geq 0$. Suppose $x = x^o \geq 0$ satisfies $Ax^o = b$, but that $Gy = d - Bx^o$, $y \geq 0$, is infeasible.

(a) Prove there exists a $\pi = \pi^o$ such that

$$G^T \pi^o \leq 0 \quad \text{and} \quad (d - Bx^o)^T \pi^o > 0. \quad (2.48)$$

(b) How can you use Phase I of the Simplex Method to find such a π^o ?

(c) Show that every feasible solution to (2.47) must satisfy

$$\begin{aligned} Ax &= b, \quad x \geq 0, \\ (\pi^o)^T Bx &\geq d^T \pi^o, \end{aligned} \quad (2.49)$$

and that the current solution $x = x^o$ violates the latter condition.

(d) Suppose $x = x^*$ satisfies (2.49) and $((\pi^o)^T B)x^* = d^T \pi^o$ (i.e., tight). Suppose

$$Gy = d - Bx^*, \quad y \geq 0, \quad (2.50)$$

is now feasible. Prove that every basic feasible solution to the system (2.50) is degenerate.

2.22 *Ph.D. Comprehensive Exam, September 24, 1988, at Stanford.* Given a linear program

$$x \geq 0, \quad Ax = b, \quad c^T x = z \quad (\text{min}),$$

let π denote the dual variables.

(a) State the dual problem in terms of π .

(b) Given alleged solutions $x = x^*$ to the primal and $\pi = \pi^*$ to the dual, state the conditions that must be satisfied by x^* and π^* in order to be optimal feasible solutions to the primal and the dual, respectively.

(c) Assume B is an optimal feasible basis. Partition A , c , and x into basic and nonbasic components, thus:

$$\begin{aligned} A_B x_B + A_N x_N &= b, & x &= (x_B, x_N)^T \geq 0, \\ c_B^T x_B + c_N^T x_N &= z \quad (\text{min}) \end{aligned}$$

Let $B^T \pi^* = c_B$ where $B = A_B$. Restate the optimality conditions in terms of the partitioned structure.

-
- (d) Assume $x^* = (x_B^*, x_N^* = 0)$ is an optimal basic feasible solution and that $N^T \pi^* < c$ where $N = A_N$ (note the strict inequality). Prove x^* is the unique optimal primal feasible solution.
- (e) Assume further that x_B^* in part (d) is degenerate. Prove that π^* is an optimal dual feasible solution but is not the unique optimal dual feasible solution. Show how to go about numerically constructing another optimal dual solution.

This page intentionally left blank

EARLY INTERIOR-POINT METHODS

An interior-point algorithm is one that improves a feasible interior solution point of the linear program by steps through the interior, rather than one that improves by steps around the boundary of the feasible region, as the classical Simplex Algorithm does. The earliest interior-point method is due to the famous mathematician John von Neumann. His method for finding a feasible solution to a linear program with a convexity constraint is notable for its simplicity and remarkable convergence properties; see Section 3.1. Since a general linear program combined with its dual can be reformulated into a feasibility problem of this restricted form, *von Neumann's algorithm* may be viewed as a method for solving the general linear program.

Just like there are many variants of the Simplex Method (which we refer to as pivot step algorithms), so there are many variants of interior methods such as *projective* and/or *potential reduction*, *affine*, and *path-following*.

1. *Projective and Potential Reduction Methods.* These methods measure the approach toward an optimal solution by the reduction of the value of a potential function rather than the reduction of the value of the linear objective. For example, Karmarkar's algorithm is typically based on projective geometry but uses a potential function to measure progress of the solution towards optimality. The potential function is typically designed to ensure the following: (a) the objective function decreases at each iteration, (b) the solution point stays in the interior of the feasible space, and (c) the algorithm converges in polynomial time. In practice, these methods have not done well.
2. *Affine Methods.* These methods approximate the feasible region, at each iteration, by an ellipsoid and optimize over the ellipsoid. The implementation of

such methods is easy as we saw in the discussion of one such method in *Linear Programming 1: Introduction*. In this chapter, we discuss Dikin's method, an early affine method. In practice these methods perform quite well but not as well as the path-following methods.

3. *Path-Following Methods*. These methods follow a certain path as the optimal solution is approached. The linear program is first transformed into an unconstrained nonlinear optimization problem, called a *logarithmic barrier function*. The logarithmic barrier function typically consists of the objective function and one or more additional terms, multiplied by a scalar positive parameter, that increase in value as the iterates approach the boundary. In effect, the additional terms throw up a barrier at the boundary. The unconstrained optimization problem is solved and the parameter value reduced for the next iteration. The optimal values of the sequence of unconstrained problems approach the optimal solution of the linear program along a path through the interior of the feasible region.

Path-following methods have performed the best in theory and practice in recent times. In Chapter 4 we will describe the primal logarithmic barrier method and the primal-dual logarithmic barrier method.

Some other interior-methods inscribe an ellipsoidal ball in the feasible region with its center at the current iterate, or first transform the feasible space and then inscribe a hypersphere with the current iterate at the center. Then an improving direction is found by joining the current iterate to the point on the boundary of the ellipsoid or sphere that maximizes (or minimizes) the linear objective function (obtained by solving a least-squares problem). A point is then selected on the improving direction line as the next current iterate. Sometimes this iterate is found along a line that is a linear combination of the improving direction and some other direction.

In 1967 Dikin proposed an affine method that in its original form is not a finite method but one that converges in the limit. In particular, Dikin's method as described in Section 3.2, has an asymptotic rate of convergence of $1 - 1/\sqrt{m}$. This method has the distinction of having been rediscovered by many; for example, the *primal affine method* is the same as *Dikin's method*.

During the period 1979–2003, there has been intense interest in the development of interior-point methods. These methods are related to classical least-square methods used in numerical analysis for making fits to data or fitting simpler functional forms to more complicated ones. Therefore interior research can tap into the vast literature of approximation theory. A theoretical breakthrough came in 1979: the Russian mathematician L. G. Khachian (based on the work of Shor, 1971–1972) discovered an ellipsoid algorithm whose running time in its worst case was significantly lower than that of the Simplex Algorithm in its worst case. Its iterates are not required to be feasible. Other theoretical results quickly followed, notably that of N. Karmarkar, who discovered an interior-point algorithm whose running-time performance in its worst case was significantly lower than that of Khachian's. This

was followed by the theoretical results of others that improved on the upper-bound estimates of the worst-case performance as the dimensions of the problems and the amount of input data increased indefinitely.

The algorithm best suited for solving a particular problem or a special class of problems may not be the same algorithm best suited for solving any problem from the broad class of problems defined by $Ax \geq b$, $x \geq 0$, $c^T x = \min$. One criterion used for comparing algorithms is upper bounds on worst-case performance times as the dimensions of the problem grow indefinitely in size. This criterion turns out to be totally misleading for deciding which algorithm to use for practical problems because these theoretical upper-bound estimates are many many times greater than any experienced with practical problems.

Attempts to characterize in a simple way the class (or classes) of practical problems from which one might be able to derive a theoretical explanation of the excellent performance times of some of the algorithms used in practice have, in general, failed. In special cases, such as the shortest-path problem, the performance of shortest-path algorithms for the entire class of shortest-path problems is comparable to that observed on actual problems. There has been progress proving that *average performance* on classes of *randomly generated* problems using a parametric variant of the Simplex Method resembles that obtained on practical problems, but no one claims these randomly generated problems are representative of the class of practical linear programs.

Because the theoretical results can be totally misleading as to what algorithm to choose to solve a practical problem, an empirical approach is used. The linear programming profession has accumulated a large collection of test problems drawn from practical sources. These are used to compare the running times of various proposed algorithms. The general goal of these efforts is to find *the algorithm* that surpasses the performance of all other algorithms in the collection.

For example, Karmarkar claimed (when he developed his method) that on very large problems his technique would be significantly faster. As of this writing, as far as the authors can ascertain, there appears to be no one algorithm that is a clear winner, i.e., that solves *all (or almost all)* of the test problems faster than all the other proposed methods. On problems with many bounding hyperplanes in the neighborhood of the optimum point, an interior method will probably do better than an exterior method. On problems with relatively few boundary planes (which is often the case in practice) an exterior method will be hard to beat. For this reason, it is likely that the commercial software of the future will be some sort of a hybrid because one does not know which kind of problem is being solved or because one wishes to obtain an extreme-point solution. Many specialized efficient codes have been proposed for solving structured linear programs such as network problems, staircase problems, block-angular systems, and multi-stage stochastic systems.

Definition (Polynomial Time): Let the problem data size (length of the input data stream) be \mathcal{L} , the total number of bits required to store the data of a linear program in m equations and n variables in the computer. An algorithm is said to have a *polynomial worst-case* running time if the algorithm's execu-

tion time (in, say, seconds) to find an optimal solution on the computer is less than some polynomial expression in \mathcal{L} , m , and n . Otherwise the algorithm is said to be NP (nonpolynomial).

Worst-Case Measures Can be Misleading. For example, given a linear program in m equations and n variables, it may be stated that a method requires less than $O(n^p m^q)$ iterations, where $O(n^p m^q)$ means some fixed constant times $n^p m^q$. If the constant for the worst-case bound were huge, say 10^{100} (which may be larger than the number of all the electrons in the universe), then such a bound would be ridiculous. Implicit in such statements about a worst-case bound is the assumption that the fixed constant is small, say 10 or 100. Usually this assumption is valid, and it has become common practice to compare worst-case bounds of algorithms *as if* the fixed constants for each of the algorithms are the same.

In general, given a linear program in m equations and n variables, projective methods require less than $O(n)$ iterations. Path-following methods require less than $O(\sqrt{n})$ iterations. Each of these also require $O(n^3)$ arithmetic operations per iteration to solve a linear least-squares subproblem in order to find a steepest descent direction. However, with refinements (such as rank-one updates) it is possible to solve each least-squares problem in $O(n^{5/2})$ arithmetic operations instead of $O(n^3)$. When the number of operations is multiplied by the bound on the number of iterations, we find that Karmarkar's projective method is bounded by $O(n^{7/2})$ arithmetic operations while path-following methods are bounded by $O(n^3)$ arithmetic operations to obtain an optimal solution within a tolerance $\epsilon > 0$. The time required to carry out the arithmetic operations depend on \mathcal{L} , the digits of input data. Thus the bound on the *time* to execute Karmarkar's algorithm is $O(n^{7/2}\mathcal{L})$.

Because the number of arithmetic operations (and iterations) can depend critically on $\epsilon > 0$, the bound on the accuracy of the computed optimal solution, we will use the following definition of a polynomial-time algorithm.

Definition (ϵ -Optimal Polynomial Time): An algorithm to solve a linear program in m rows and n columns is said to have a *polynomial worst-case* running time (measured in seconds, say) if the time to execute it is less than some polynomial expression in \mathcal{L} , m , n , $q = -\log_{10} \epsilon$, where \mathcal{L} is the total number of bits required to store the problem's data in the computer, q is the number of significant decimal digits of accuracy of the optimal solution, and the tolerance ϵ is some measure of how much the calculated solution differs from the true optimal objective value, or an approximate feasible solution to a feasibility problem differs from the right-hand side of a true feasible solution.

3.1 VON NEUMANN'S METHOD

Von Neumann, in a discussion with George Dantzig in 1948, proposed the first interior algorithm for finding a feasible solution to a linear program with a convexity

constraint recast in the form:

$$x \geq 0, \quad \sum_1^n x_j = 1, \quad \sum_1^n P_j x_j = 0, \quad \|P_j\|_2 = 1 \quad \text{for } j = 1, \dots, n. \quad (3.1)$$

He provided no proof of its convergence properties. In a follow-up letter to von Neumann, Dantzig proved that if the problem is feasible, it has the remarkable property that, *independent of the number of rows m and columns n* , it is guaranteed to generate in less than t iterations an approximate feasible solution with a precision

$$\epsilon \leq \frac{1}{\sqrt{t}} \quad (3.2)$$

where ϵ^2 is the sum of the squares of errors of the fit of the left-hand side of the equations to their right-hand side.

In the worst-case scenario, which gives rise to (3.2), all the points P_j lie on, or on one side of, a hyperplane through the origin. This, as we will see, causes the algorithm to have an exponentially slow rate of convergence, namely an upper bound on iterations $t = 10^{2q}$, which is not a polynomial expression in (m, n, q) , to achieve q decimal digits of accuracy. However, when the convex hull of the points P_j contains the origin in its interior, we will prove a very strong result, namely, the algorithm generates in less than t iterations an approximate feasible solution with a precision,

$$\epsilon \leq (1 - r^2)^{t/2}, \quad 0 < r < 1, \quad (3.3)$$

in the worst case, where the *fixed* constant r is the radius of the largest ball centered at the origin that is contained in the convex hull of the P_j s. In this case the polynomial expression for the number of iterations t is linear in q , independent of the m and n , namely:

$$t \leq \frac{2 \log_{10} \epsilon}{\log_{10}(1 - r^2)} = -\frac{2q}{\log_{10}(1 - r^2)}$$

The work per iteration is approximately mn multiplications and additions; see Exercise 3.17.

- ▷ **Exercise 3.1** Show that the general linear program $\min c^T x, Ax \geq b, x \geq 0$ with feasible primal and dual solutions is equivalent to $Ax \geq b, A^T y \leq c, b^T y \geq c^T x, x \geq 0, y \geq 0$.
- ▷ **Exercise 3.2** Show that the number of iterations to attain an approximate feasible solution whose Euclidean distance $\epsilon = 10^{-q}$ from the origin is $t \leq 16q$ if $r = 0.5$. Show that $t \leq 113q$ if $r = 0.2$.
- ▷ **Exercise 3.3** Show that a feasibility problem $Ax = b, x \geq 0$ can be reduced to a feasibility problem with a convexity constraint by adding a relation $\sum_j x_j + x_o = M$ where M is sufficiently large.

Consider the general linear program feasibility problem with a convexity constraint: Find $y = (y_1, y_2, \dots, y_n) \geq 0$ such that

$$\begin{aligned} \sum_{j=1}^n Q_j y_j &= b, \quad \text{where } Q_j \in \mathfrak{R}^m \\ \sum_{j=1}^n y_j &= 1, \\ y_j &\geq 0 \text{ for } j = 1, \dots, n. \end{aligned} \tag{3.4}$$

This system is equivalent to

$$\begin{aligned} \sum_{j=1}^n \widehat{Q}_j y_j &= 0, \quad \text{where } \widehat{Q}_j = Q_j - b \\ \sum_{j=1}^n y_j &= 1, \\ y_j &\geq 0 \text{ for } j = 1, \dots, n, \end{aligned} \tag{3.5}$$

which we will refer to as the *Center of Gravity Problem*, which can be transformed into a *Center of Gravity Problem with Normalized Columns* $\|P_j\| = 1$ for $j = 1, \dots, n$: Find $x = (x_1, x_2, \dots, x_n) \geq 0$ such that

$$\begin{aligned} \sum_{j=1}^n P_j x_j &= 0, \quad \text{where } P_j \in \mathfrak{R}^m, \quad \|P_j\| = 1 \\ \sum_{j=1}^n x_j &= 1. \end{aligned} \tag{3.6}$$

This is done by setting:

$$P_j = \frac{\widehat{Q}_j}{\|\widehat{Q}_j\|}, \tag{3.7}$$

and noting that if $x = x^o$ solves (3.6) then $y = y^o$ solves (3.4) where

$$y_j^o = \frac{x_j^o / \|\widehat{Q}_j\|}{\sum_{k=1}^n x_k^o / \|\widehat{Q}_k\|}. \tag{3.8}$$

Conversely, if $y = y^o$ solves (3.4) then $x = x^o$ solves (3.6) where

$$x_j^o = \frac{\|\widehat{Q}_j\| y_j^o}{\sum_{k=1}^n \|\widehat{Q}_k\| y_k^o}. \tag{3.9}$$

▷ **Exercise 3.4** Verify that if $x = x^o$ solves (3.6), then $y = y^o$ solves (3.4) and conversely.

▷ **Exercise 3.5** Prove that if there exists a nondegenerate basic feasible solution, the convex hull of the P_j s contains a ball of positive radius centered at the origin.

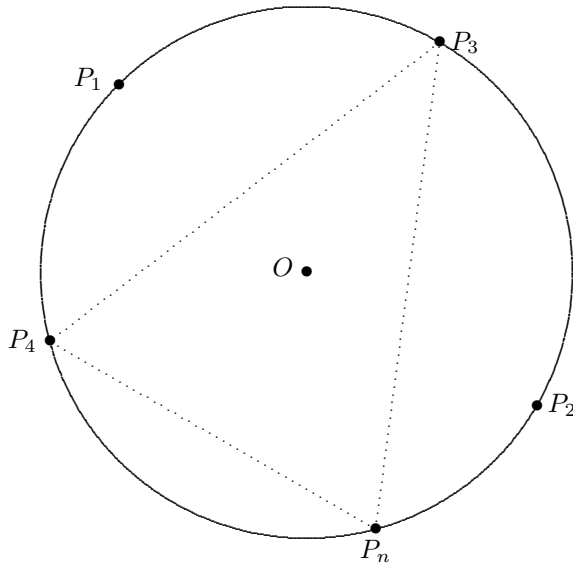


Figure 3-1: The Two-Dimensional Center-of-Gravity Problem: Find a Simplex that Contains the Origin

3.1.1 THE VON NEUMANN ALGORITHM

Given n points $P_j \in \mathfrak{R}^m$ located on the surface of an m -dimensional sphere \mathcal{S} of unit radius centered at the origin O , the problem is to find nonnegative weights $x_j = x_j^*$ to assign to the points P_j such that their weighted center of gravity is the origin, or prove that no such weighting exists; (3.10) states this problem algebraically. See Figure 3-1 for a two-dimensional example.

- ▷ **Exercise 3.6** Given the coordinates of n points P_j on a circle, devise a very efficient algorithm for solving the 2-dimensional center of gravity problem on a computer. Try to generalize your procedure to higher dimensions.

By an *approximate solution* G to a Center of Gravity Problem (3.10) with normalized columns, we mean any nonnegative weighted linear combination of points P_j whose weights $x_j \geq 0$ sum to one.

$$G = \sum_{j=1}^n P_j x_j = 0, \quad \sum_{j=1}^n x_j = 1, \quad P_j^T P_j = 1, \quad \text{for } j = 1, \dots, n. \quad (3.10)$$

The von Neumann Algorithm iteratively finds improving approximations

$$G^1, G^2, \dots, G^t, G^{t+1}, \dots$$

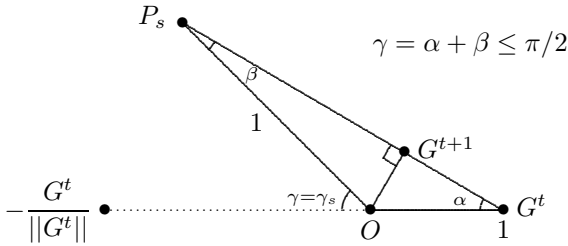


Figure 3-2: Finding an Improved Approximation

whose distances $\|G^t\|$ from the origin O decrease toward zero if the problem is feasible. Given an $\epsilon > 0$, the algorithm terminates in a finite number of iterations t with $0 \leq \|G^t\| < \epsilon$ or terminates on satisfying a certain condition that implies no feasible solution exists.

Von Neumann's algorithm is initiated with the approximate solution $G^1 = P_1$ or any arbitrary convex linear combination:

$$G^1 = \sum_{j=1}^n P_j x_j^1, \quad \sum_{j=1}^n x_j^1 = 1, \quad x_j^1 \geq 0 \quad \text{for } j = 1, \dots, n. \quad (3.11)$$

Given an approximate solution for iteration t ,

$$G^t = \sum_{j=1}^n P_j x_j^t, \quad \sum_{j=1}^n x_j^t = 1, \quad x_j^t \geq 0 \quad \text{for } j = 1, \dots, n, \quad (3.12)$$

the von Neumann algorithm either generates the next approximate:

$$G^{t+1} = \sum_{j=1}^n P_j x_j^{t+1}, \quad \sum_{j=1}^n x_j^{t+1} = 1, \quad x_j^{t+1} \geq 0 \quad \text{for } j = 1, \dots, n, \quad (3.13)$$

with the property

$$\|G^{t+1}\| < \|G^t\| \quad (3.14)$$

or terminates with the condition that implies no feasible solution exists.

On iteration t , von Neumann selects P_s as the direction P_j that makes the sharpest angle γ_j with the direction $-G^t/\|G^t\|$ for $j = 1, \dots, n$.

LEMMA 3.1 (Properties of Improved Solution G^{t+1}) *If $\gamma_s \leq \pi/2$, the improved solution is G^{t+1} (see Figure 3-2), the point closest to the origin O on the line joining P_s to G^t . The point G^{t+1} satisfies $\|G^{t+1}\| < \|G^t\|$.*

Proof. We must show two things. First that G^{t+1} is a convex combination of P_1, P_2, \dots, P_n and second that $\|G^{t+1}\| < \|G^t\|$.

By assumption $\gamma_s \leq \pi/2$. Since γ_s , the exterior angle of the triangle OG^tP_s , is greater than the interior angle α , the angle α is acute implying G^{t+1} is an *interior point* on the line $(P_s)^TG^t$ and hence is a convex combination of P_1, P_2, \dots, P_n . Finally $\|G^{t+1}\| < \|G^t\|$ because

$$\|G^{t+1}\| = \|G^t\| \sin \alpha < \|G^t\|. \quad (3.15)$$

■

LEMMA 3.2 (Infeasibility Condition) *If $\gamma_s = \min_j \gamma_j > \pi/2$ on some iteration t , there exist no feasible solution to the linear program.*

Proof. If $\gamma_s > \pi/2$, then the angles γ_j of all P_j with $-G^t$ are greater than $\pi/2$, implying all points P_j lie strictly on one side of the hyperplane through the origin orthogonal to the direction $-G^t$ and hence every convex combination G of the points P_j lies strictly on the same side as G^t implying there exists no convex combination $G = 0$. ■

▷ **Exercise 3.7** Convert the geometric proof of Lemma 3.2 into an algebraic proof.

THEOREM 3.3 (Convergence to an Optimal) *If $\gamma_t \leq \pi/2$ for all iterations t , then*

$$\|G^t\| \rightarrow 0 \quad \text{as} \quad t \rightarrow +\infty.$$

Proof. Let $\gamma_t = \gamma_s = \pi/2 - \Delta$, $\Delta \geq 0$; see Figure 3-2. Then noting $\gamma = \alpha + \beta$,

$$\begin{aligned} \|G^{t+1}\| &= \sin \beta = \sin(\gamma - \alpha) = \sin(\pi/2 - \Delta - \alpha) = \cos(\Delta + \alpha) \\ &\leq \cos \alpha \end{aligned} \quad (3.16)$$

$$\frac{\|G^{t+1}\|}{\|G^t\|} = \sin \alpha \quad (3.17)$$

Squaring (3.16) and (3.17), and summing

$$\frac{\|G^{t+1}\|^2}{\|G^t\|^2} + \|G^{t+1}\|^2 \leq \sin^2 \alpha + \cos^2 \alpha = 1$$

Therefore

$$\frac{1}{\|G^t\|^2} + 1 \leq \frac{1}{\|G^{t+1}\|^2} \quad \text{for all } t. \quad (3.18)$$

Starting with $G^1 = P_1$ and $\|G^1\| = 1$, summing the preceding relations from $t = 1$ to $t = T - 1$ and cancelling the corresponding terms on each side of the inequality, we obtain

$$T \leq \frac{1}{\|G^T\|^2}; \quad \|G^T\| \leq \frac{1}{\sqrt{T}} \rightarrow 0 \quad \text{as } T \rightarrow +\infty. \quad (3.19)$$

■

THEOREM 3.4 (Convergent Subsequence when $\|G^t\| \rightarrow 0$) If $G^t = \sum P_j x_j^t$, $\sum x_j^t = 1$, $x^t \geq 0$ and $\|G^t\| \rightarrow 0$ as $t \rightarrow \infty$, then there exists a subsequence $t = t_1, t_2, t_3, \dots$ such that $0 = G^* = \sum P_j x_j^*$, where $x_j^{t_i} \rightarrow x_j^* > 0$ for $j = 1, \dots, n$ and $\sum x_j^* = 1$.

▷ **Exercise 3.8** Prove Theorem 3.4.

THEOREM 3.5 (Convergence When Sharpest Angle Always $< \pi/2$) If the sharpest angle is \leq some $\gamma^* < \pi/2$ for all t , then

$$\|G^{t+1}\| \leq (\sin \gamma^*)^{t+1} \rightarrow 0 \quad \text{as } t \rightarrow +\infty.$$

Proof. In Figure 3-2, exterior angle $\gamma^t > \alpha$; therefore

$$\|G^{t+1}\| = \|G^t\| \sin \alpha < \|G^t\| \sin \gamma_t \leq \|G^t\| \sin \gamma^*;$$

whence, assuming $\|G^1\| = 1$, $\|G^{t+1}\| \leq (\sin \gamma^*)^{t+1} \rightarrow 0$ as $t \rightarrow +\infty$. ■

▷ **Exercise 3.9** Prove $\|G^t\| \rightarrow 0$, if an infinite subsequence $t = t_1, t_2, t_3, \dots$ satisfies $\gamma_{t_i} \leq \gamma^*$.

THEOREM 3.6 (Convergence If Ball Contains Origin) Given the class of normalized feasibility problems (3.6) with the property that the convex hull of the P_j s contains a ball \mathcal{B} centered at the origin with positive radius $0 < \rho < 1$ then

$$\|G^t\| \leq (1 - \rho^2)^t \rightarrow 0 \quad \text{as } t \rightarrow +\infty,$$

and the rate of convergence to the origin is linear, i.e., the number of iterations t required for $\|G^t\| \leq 10^{-q}$ is $t \leq (-\log(1 - \rho^2))^{-1} q$.

Proof. In Figure 3-3, the vertical dotted line through Q represents the hyperplane \mathcal{H} through Q orthogonal to direction $-G^t/\|G^t\|$, where Q is a point on the surface of ball \mathcal{B} at a distance ρ from the origin. Because $Q \in \mathcal{B}$, it is a convex combination of the P_j s. We claim for all t , on each iteration t , P_s , the direction P_j that makes the sharpest angle with $-G^t/\|G^t\|$, lies on the opposite side of \mathcal{H} from the origin O . If this is true, this implies distance

$$OR = \cos \gamma_s \geq \rho \quad \text{for all } t > 1. \quad (3.20)$$

Assume, on the contrary, $\cos \gamma_s = \max_j \cos \gamma_j < \rho$. Then all $\cos \gamma_j < \rho$ and all P_j lie on the origin side of \mathcal{H} . This implies that \mathcal{B} , the ball containing all convex combinations of P_j , including Q , lie *strictly* on the origin side of \mathcal{H} , which contradicts the fact that Q lies on the hyperplane \mathcal{H} . We conclude that (3.20) is true. The rest of the proof is straightforward.

$$\|G^{t+1}\| = \|G^t\| \sin \alpha < \|G^t\| \sin \gamma_s = \|G^t\| \sqrt{1 - \cos^2 \gamma_s} \leq \sqrt{1 - \rho^2}.$$

■

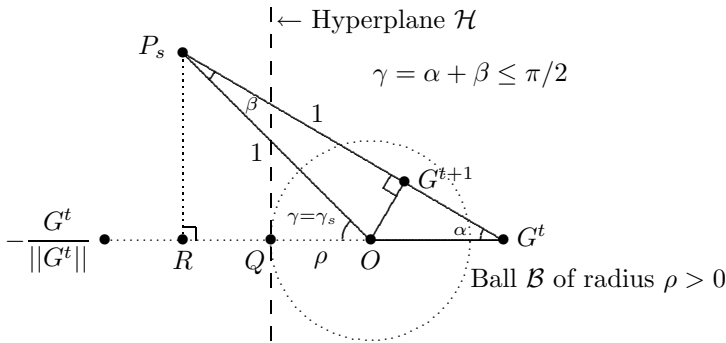


Figure 3-3: Convergence under Existence of a Ball \mathcal{B}

▷ **Exercise 3.10** Convert the geometric proof of Theorem 3.6 into an algebraic proof.

THEOREM 3.7 (Linear Rate of Convergence under Nondegeneracy) *If a nondegenerate basic solution to the feasibility problem (3.6) exists, then the iterates*

$$\|G^t\| \leq (1 - r^2)^{t/2} \rightarrow 0 \quad \text{as } t \rightarrow \infty \quad \text{for all } t > 1$$

where $r > 0$ is the largest ball centered at the origin in the finite set of simplices associated with the nondegenerate basic feasible solutions.

▷ **Exercise 3.11** Prove Theorem 3.7. How is finiteness of the number of simplices used to prove linear rate of convergence?

▷ **Exercise 3.12** If the radius of the largest ball in the convex hull centered at the origin has radius $r = 0$, all basic feasible solutions to the linear program are degenerate.

▷ **Exercise 3.13** Construct a two-dimensional example in which all basic feasible solutions are degenerate but the largest ball centered at the origin contained in the convex hull has positive radius.

▷ **Exercise 3.14** Construct a two-dimensional example whose convex hull contains a ball of positive radius that contains the origin, and every basic solution is degenerate.

We have described the von Neumann algorithm geometrically. We now describe it algebraically. The iterative process is repeated with iteration counter t set equal

to $t + 1$. The direction P_s can be computed by:

$$s = \operatorname{argmax}_{j=1,\dots,n} \{ \cos \gamma_j \} \quad \text{where} \quad \cos \gamma_j = \frac{-(G^t)^T P_j}{\|G^t\| \|P_j\|}. \quad (3.21)$$

Since $\|G^t\|$ and $\|P_j\|$ in the denominator are independent of j (recall that $\|P_j\| = 1$), we can simplify the determination of P_s by

$$s = \operatorname{argmax}_{j=1,\dots,n} \{ -(G^t)^T P_j \}, \quad (3.22)$$

and setting

$$\cos \gamma_s = -\frac{(G^t)^T P_s}{\|G^t\| \|P_s\|}. \quad (3.23)$$

To algebraically determine the point G^{t+1} closest to the origin on the line joining P_s to G^t , set $G^{t+1} = \lambda P_s + (1 - \lambda)G^t$, and determine the value of the scalar $\lambda = \bar{\lambda}$ that minimizes

$$\|G^{t+1}\|^2 = \|G^t\|^2 + 2\lambda(G^t)^T(P_s - G^t) + \lambda^2\|P_s - G^t\|^2, \quad (3.24)$$

by setting $\partial\|G^{t+1}\|^2/\partial\lambda = 0$. This yields

$$(G^t)^T(P_s - G^t) + \bar{\lambda}\|P_s - G^t\|^2 = 0 \quad (3.25)$$

and

$$\bar{\lambda} = \frac{\|G^t\|^2 - (G^t)^T P_s}{\|G^t\|^2 - 2(G^t)^T P_s + 1}. \quad (3.26)$$

▷ **Exercise 3.15** Show that $\lambda = \bar{\lambda}$ yields the minimum and not the maximum of the expression for G^{t+1} in (3.24).

Hence we determine the new approximation

$$G^{t+1} = \bar{\lambda}P_s + (1 - \bar{\lambda})G^t. \quad (3.27)$$

Replacing λ in (3.24) by optimal $\bar{\lambda}$ and subtracting $\bar{\lambda}$ times (3.25) from (3.24) and rearranging terms we obtain

$$\operatorname{Min} \|G^{t+1}\|^2 = (1 - \bar{\lambda})\|G^t\|^2 + \bar{\lambda}(G^t)^T P_s. \quad (3.28)$$

The next step is to update the weights x as follows:

$$\begin{aligned} x_j^{t+1} &= (1 - \bar{\lambda})x_j^t && \text{for all } j \neq s \\ x_s^{t+1} &= (1 - \bar{\lambda})x_s^t + \bar{\lambda}. \end{aligned} \quad (3.29)$$

Comment: Since $-(G^t)^T P_s > 0$, it follows from (3.26) that $1 > \bar{\lambda} > 0$, $(1 - \bar{\lambda}) > 0$ and therefore from (3.29) and $x_j^t \geq 0$, that $x_j^{t+1} \geq 0$ for all j ; also $\sum_{j=1}^n x_j^{t+1} = 1$. We have already shown that $\|G^{t+1}\| < \|G^t\|$. This can also be seen by rewriting (3.28) as

$$\operatorname{Min} \|G^{t+1}\|^2 = \|G^t\|^2 - \bar{\lambda}(\|G^t\|^2 - (G^t)^T P_s) \quad (3.30)$$

and noting $-(G^t)^T P_s \geq 0$.

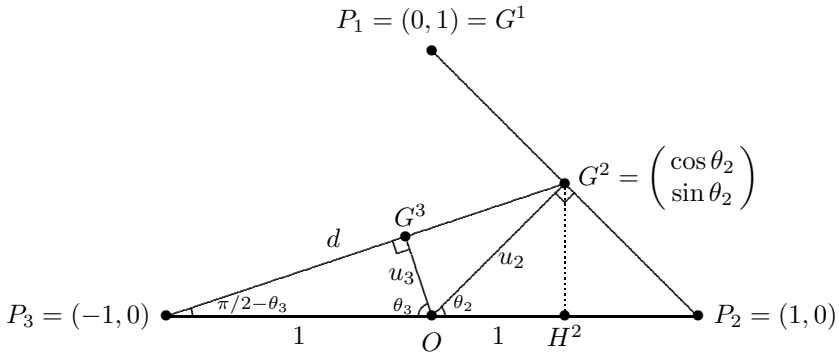


Figure 3-4: Degenerate Two-Dimensional Case

▷ **Exercise 3.16** Prove that as γ_s varies between 0 and $\pi/2$ that

$$\frac{\|G^t\|^2}{\|G^t\|^2 + 1} \leq \bar{\lambda} \leq \frac{\|G^t\|}{\|G^t\| + 1} \quad (3.31)$$

▷ **Exercise 3.17** Define δ to be the *nonzero coefficient density*, i.e., the ratio of the number of nonzeros to the total number of elements in the coefficient matrix. Show that the work of the various steps per iterations is:

	Multiplications	Additions	Comparisons
Step 1	δmn	δmn	n
Step 3	$2m + n + 9$	$m + 8$	1

▷ **Exercise 3.18** Suppose system (3.4) has a nonzero coefficient density of δ and b is 100% dense. Then the coefficient density of system (3.5) will also turn out to be 100% dense. Show, however, that you can replace the computation of $-(G^t)^T P_j$ of the algorithm with one that preserves the sparsity of the original Q_j by replacing P_j by $Q_j - b$ in (3.5).

Convergence Rate in the Degenerate Two-Dimensional Case. Apply the von Neumann algorithm to the Center of Gravity Problem (3.6) with $n = 3$; see Figure 3-4. The P_j s are

$$P_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad P_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad P_3 = \begin{pmatrix} -1 \\ 0 \end{pmatrix}.$$

Starting with

$$G^1 = P_2 \text{ or } G^1 = P_3,$$

the von Neumann algorithm converges in one iteration. However, starting on iteration $t = 1$ at

$$G^1 = P_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

the von Neumann algorithm, on even iterations t , arrives at

$$G^t = u_t \begin{pmatrix} \cos \theta_t \\ \sin \theta_t \end{pmatrix} \quad \text{where } u_t = \cos \theta_t > 0; \quad (3.32)$$

and on odd iterations arrives at

$$G^{t+1} = \cos \theta_t \begin{pmatrix} -\cos \theta_{t+1} \\ \sin \theta_{t+1} \end{pmatrix} \quad \text{where } u_{t+1} = \cos \theta_{t+1} > 0, \quad (3.33)$$

because the iterates have the property that the angle $OG^t P_2 = \pi/2$. On the next odd iteration G^{t+1} is obtained by joining G^t to P_2 and dropping a perpendicular from the origin O . Therefore angle $OG^t P_3 = \pi/2$. Angle $P_2 OG^t$ is denoted by θ_t and angle $P_3 OG^{t+1}$ is denoted by θ_{t+1} .

The distance G^t to P_t is denoted by d_t , where, for $t \geq 2$

$$d_t^2 = (u_t \cos \theta_t + 1)^2 + u_t^2 \sin^2 \theta_t = u_t^2 + 2u_t \cos \theta_t + 1 = 3u_t^2 + 1.$$

The angle θ_{t+1} is related to θ_t by (see the dotted line $G^t H^t = G^2 H^2$ in Figure 3-4):

$$\begin{aligned} G^t H^t &= d_t \sin(\pi/2 - \theta_{t+1}) = u_t \sin \theta_t \\ d_t \cos \theta_{t+1} &= u_t \sin \theta_t \\ \frac{1}{u_{t+1}} &= \frac{(1 + 3u_t^2)^{1/2}}{u_t(1 - u_t^2)^{1/2}}. \end{aligned}$$

From this it follows for $t \geq 2$ that

$$\frac{1}{u_{t+1}^2} = \frac{1}{u_t^2} + \frac{4}{1 - u_t^2}. \quad (3.34)$$

For $t \geq t_0$, because $u_t \leq u_{t_0}$,

$$\frac{1}{u_{t+1}^2} \leq \frac{1}{u_t^2} + \frac{4}{1 - u_{t_0}^2} \quad \text{for } t = t_0, t_0 + 1, \dots \quad (3.35)$$

If we sum the relations (3.34) from $t = t_0$ to $t = T - 1$ for $T \geq t_0$ we obtain

$$\frac{1}{u_T^2} \leq \frac{1}{u_{t_0}^2} + \frac{4(T - t_0)}{1 - u_{t_0}^2}. \quad (3.36)$$

Starting with $G^2 = (1/2, 1/2)^T$, $u_2^2 = 1/2$, from (3.35), $u_3^2 = 1/10$. Letting $t_0 = 3$, $u_{t_0}^2 = 1/10$ and

$$\frac{1}{u_T^2} \leq \frac{40T - 30}{9} \quad \text{or} \quad u_T \geq 3/\sqrt{40T - 30}. \quad (3.37)$$

In other words, u_T is converging to 0 more slowly than $3/\sqrt{40T - 30}$ in T iterations. Letting $\epsilon_T = 3/\sqrt{40T - 30} = 10^{-k}$, to obtain an approximate solution to k decimal digits of accuracy will require the number of iterations T to satisfy

$$10^{-k} > 3/\sqrt{40T - 30}$$

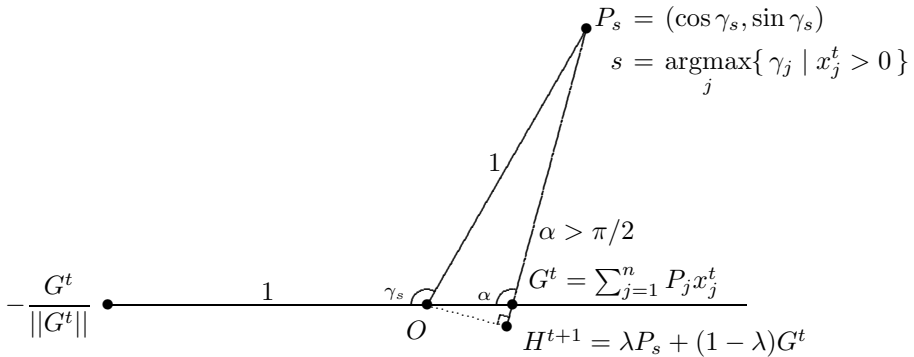


Figure 3-5: Decreasing x_s^t to Improve the Rate of Convergence

implying

$$T > 10^{2k-0.65} \tag{3.38}$$

iterations, which is an exponential expression in (m, n, k) ; and therefore in this degenerate case, starting at P_1 , the von Neumann convergence rate is nonpolynomial.

3.1.2 IMPROVING THE RATE OF CONVERGENCE

In Section 3.1.1, we showed that the von Neumann algorithm, when applied to the degenerate two-dimensional example $P_1 = (0, 1)$, $P_2 = (1, 0)$, $P_3 = (-1, 0)$, and initiated with $G^1 = P_1$, converges exponentially slowly; see Figure 3-4 and Equation (3.38). In general, its convergence rate can be sped up significantly by *decreasing* the weight $x_s^t > 0$ for some term $j = s$ in the approximation $G^t = \sum_{j=1}^n P_j x_j^t$ and adjusting the remaining weights x_j^t proportionally upward so that their weights re-sum to one.

This improvement can be striking in the degenerate case.

Definition (Degenerate Problem): A problem is *degenerate* by definition if there are no points P_j on one side of some hyperplane through the origin.

The previous example is degenerate because letting $P_j = (u_j, v_j)$ there are no points below the line $v = 0$. At the end of iteration 3, $G^3 = P_1 x_1^3 + P_2 x_2^3 + P_3 x_3^3$ where $x_1^3 > 0$, $x_2^3 > 0$, $x_3^3 > 0$. *Reducing* x_1^3 to 0 and *increasing* x_2^3 , x_3^3 to ρx_2^3 , ρx_3^3 , where $\rho = 1/(1 - x_1^3)$, results in $G^4 = 0$ on the next iteration.

Algorithm 3.1 (Improved von Neumann Algorithm) Given $P_j \in \mathbb{R}^n$, $\|P_j\| = 1$,

and ϵ , the algorithm converges to an ϵ -optimal solution, if one exists, to the problem:

$$\sum_{j=1}^n P_j x_j = 0, \quad \sum_{j=1}^n x_j = 1, \quad x_j \geq 0 \quad \text{for } j = 1, \dots, n. \quad (3.39)$$

Initiate the approximation $G^t = \sum_{j=1}^n P_j x_j^t$, at $t = 1$ as follows. Let $E^2 = \|G^t\|^2$ measure the square of the error of the approximation G^t and set

$$t \leftarrow 1, \quad G^t \leftarrow P_1, \quad x_1^t \leftarrow 1, \quad x_j^t \leftarrow 0 \text{ for } j > 1, \quad E^2 \leftarrow \|G^t\|^2 = 1, \quad FLAG \leftarrow 1.$$

1. *Determine s , the Index of x_s .* For $j = 1, \dots, n$ compute

$$\delta_j = -(G^t)^T P_j. \quad (3.40)$$

Set

$$s = \begin{cases} \underset{j}{\operatorname{argmin}}\{\delta_j\} & \text{if } FLAG = 1 \\ \underset{j}{\operatorname{argmax}}\{\delta_j \mid x_j^t > 0\} & \text{if } FLAG = -1 \end{cases} \quad (3.41)$$

2. *Terminate Infeasible.* If $FLAG = +1$ and $\delta_s > 0$, no feasible solution exists. STOP.

3. *Find Adjustments for Next Approximation.* Determine λ that minimizes the norm of the next approximation $\|\lambda P_s + (1 - \lambda)G^t\|$; i.e.,

$$\lambda \leftarrow \frac{E^2 - \delta_s}{E^2 - 2\delta_s + 1}. \quad (3.42)$$

Adjust λ so that the updated $x_s^{t+1} \geq 0$. If $FLAG = -1$ and $\lambda < -x_s^t/(1 - x_s^t)$, then

$$\lambda \leftarrow -\frac{x_s^t}{1 - x_s^t}. \quad (3.43)$$

4. *Update Approximation.*

Set:

$$\begin{aligned} E^2 &\leftarrow (1 - \lambda^2)E^2 + 2\lambda(1 - \lambda)\delta_s + \lambda^2 \\ G^{t+1} &\leftarrow \lambda P_s + (1 - \lambda)G^t \\ x_j^{t+1} &\leftarrow (1 - \lambda)x_j^t \quad \text{for all } j \neq s \\ x_s^{t+1} &\leftarrow (1 - \lambda)x_s^t + \lambda \\ FLAG &\leftarrow -FLAG \\ t &\leftarrow t + 1. \end{aligned}$$

5. *Loop Back.* If $E > \epsilon$ return to Step 1.

6. *Terminate ϵ -Feasible.* If $E \leq \epsilon$, STOP and report the iteration count t , the value $E^2 = \|G^t\|^2$, and the ϵ -feasible solution x^t .

3.1.3 Von Neumann Algorithm as a Variant of the Simplex Algorithm

The Improved von Neumann Algorithm may be interpreted as a variant of the Simplex Algorithm with a separable quadratic objective. We illustrate this for the case $m = 3$.

At iteration t of the Improved von Neumann Algorithm,

$$G^t = \sum_{j=1}^n P_j x_j^t, \quad G^t = \begin{pmatrix} g_1^t \\ g_2^t \\ g_3^t \end{pmatrix}, \quad x_j^t \geq 0 \text{ for } j = 1, \dots, n. \quad (3.44)$$

The equivalent problem for solution by the Simplex Algorithm is:

Find Min $z, v \geq 0, x_j^t v + \Delta x_j \geq 0$, for $j = 1, \dots, n$,
such that

$$\begin{aligned} \frac{1}{2}g_1^2 + \frac{1}{2}g_2^2 + \frac{1}{2}g_3^2 &= z && \text{:Multipliers} \\ -g_1 &+ g_1^t v + \sum_{j=1}^n a_{1j} \Delta x_j = 0 && \text{:}-g_1^t \\ -g_2 &+ g_2^t v + \sum_{j=1}^n a_{2j} \Delta x_j = 0 && \text{:}-g_2^t \\ -g_3 &+ g_3^t v + \sum_{j=1}^n a_{3j} \Delta x_j = 0 && \text{:}-g_3^t \\ &1v + \sum_{j=1}^n \Delta x_j = 1 && \text{:}\|G^t\|^2 \end{aligned} \quad (3.45)$$

where a_{ij} are the coefficients from the original problem.

The current basic variables are $g_1^t = 0, g_2^t = 0, g_3^t = 0$, and $v = 1$, and the nonbasic variables are $\Delta x_j = 0$, for $j = 1, \dots, n$.

▷ **Exercise 3.19** Prove that (3.45) is equivalent to the original problem.

▷ **Exercise 3.20** Prove that the simplex multipliers π associated with the basis is

$$\pi = (g_1^t, g_2^t, g_3^t, \|G^t\|^2).$$

▷ **Exercise 3.21** Assume that all Δx_j except Δx_s are fixed at $\Delta x_j = 0$ and the adjusted values of z, g_i, v are expressed as a function of Δx_s .

1. Determine the range of Δx_s such that $x_j^{t+1} \geq 0$ for all j .
2. Determine Δx_s that minimizes z subject to $x_j^{t+1} \geq 0$ for all j .

- ▷ **Exercise 3.22** The Simplex Algorithm “prices” out the columns to determine which column $j = s$ to increase (or decrease). Show that this results in the criterion that z will decrease with an infinitesimal decrease in x_j if $(P_j)^T G^t < \|G^t\|^2$ or with an infinitesimal increase in x_j if $(P_j)^T G^t > \|G^t\|^2$ and $x_j^t \geq 0$.
- ▷ **Exercise 3.23** Let $B = (P_{j_1}, P_{j_2}, \dots, P_{j_k})$ be k independent columns and let $\pi^T P_{j_i} = 1$ for all j_i . Prove that $\bar{B} = (P_{j_1}, P_{j_2}, \dots, P_{j_k}, P_s)$ is an augmented set of $k + 1$ independent columns if $\pi^T P_s \neq 1$.
- ▷ **Exercise 3.24** Let $B = (P_{j_1}, P_{j_2}, \dots, P_{j_k})$ be k independent columns. Prove that $B^T B$ is a $k \times k$ nonsingular matrix that can be used to find y such that By is closest to b in the least-squares sense.
- ▷ **Exercise 3.25** Let $B = (P_{j_1}, P_{j_2}, \dots, P_{j_k})$ be k independent columns. Show how $B^T B$ can be used to determine By closest to the origin $b = 0$ where $e^T y = 1$, $e = (1, 1, \dots, 1)^T$. Note that if P_{j_i} is very sparse, then BB^T is likely to be sparse and likely to make it computationally efficient to solve a system $BB^T u = e$. Whereas if

$$\bar{B} = \begin{pmatrix} P_{j_1} & P_{j_2} & \cdots & P_{j_k} \\ 1 & 1 & \cdots & 1 \end{pmatrix}$$

then $\bar{B}^T \bar{B}$ is 100% dense; verify that $\bar{B}^T \bar{B}$ is 100% dense.

- ▷ **Exercise 3.26** If P_j are m component vectors and $B = (P_{j_1}, P_{j_2}, \dots, P_{j_m})$ are m independent columns, show how the inverse of B instead of the inverse of $B^T B$ can be used to determine By closest to the origin $b = 0$ where $e^T y = 1$, $e = (1, 1, \dots, 1)^T$.

3.2 DIKIN'S METHOD

We are concerned with the linear program whose primal form is

$$(\mathcal{P}) \quad \begin{array}{ll} \text{Minimize} & z = c^T x \\ \text{subject to} & x \in \{x \in \mathfrak{R}^n \mid Ax = b, x \geq 0\} \end{array} \quad (3.46)$$

where $c \in \mathfrak{R}^n$, $A \in \mathfrak{R}^{m \times n}$, and $0 \neq b \in \mathfrak{R}^m$ are given. Its dual form is

$$(\mathcal{D}) \quad \begin{array}{ll} \text{Maximize} & v = b^T y \\ \text{subject to} & y \in \{y \in \mathfrak{R}^m \mid c - A^T y \geq 0\}. \end{array} \quad (3.47)$$

We denote the dual slack variables by

$$u = c - A^T y \geq 0. \quad (3.48)$$

When feasible solutions exist for both (\mathcal{P}) and (\mathcal{D}) , then feasible solutions $x = x^*$ and $(y, u) = (y^*, u^*)$ exist and are optimal for (\mathcal{P}) and (\mathcal{D}) if and only if (see Section 2.2) the complementary conditions

$$u_j x_j = 0 \quad \text{for } j = 1, \dots, n \quad (3.49)$$

are satisfied. It is convenient to define $D = [D_{ij}]$ as the diagonal matrix

$$D = \text{Diag}(u),$$

where

$$D_{ij} = \begin{cases} 0 & \text{if } i \neq j, \\ u_i & \text{if } i = j. \end{cases} \quad (3.50)$$

It is easy to verify that the condition

$$Dx = 0 \quad (3.51)$$

is a convenient way to write the complementary conditions (3.49) in matrix notation. We therefore seek feasible solutions for (\mathcal{P}) and (\mathcal{D}) , see (3.46) and (3.47), satisfying the complementary conditions (3.51).

Recall some definitions: A *primal solution* is any x satisfying $Ax = b$; it is feasible if $x \geq 0$. A *dual solution* is any (y, u) satisfying $A^T y + u = c$; it is feasible if $A^T y \leq c$ or $u \geq 0$. Neither primal nor dual solutions need be feasible in general; however, all dual solutions we will be generating in Dikin's affine algorithm, will be required to be feasible but primal solutions will not necessarily be required to be feasible. A partition of indices $j = 1, \dots, n$ into two sets consisting of any m indices and the remaining $(n - m)$ indices will be denoted by $\{\mathcal{B}, \mathcal{N}\}$. The set $A_{\mathcal{B}}$ of columns corresponding to \mathcal{B} we will later assume to be nonsingular and hence it will form a basis in the space generated by linear combinations of the columns of A . The *basic* primal and dual solutions associated with some partition $\{\mathcal{B}, \mathcal{N}\}$ will be denoted by \bar{x} and (\bar{y}, \bar{u}) , respectively. In other words, *the bar* over the symbol denotes that the solution is basic.

By definition, the primal solution $\bar{x} = (\bar{x}_{\mathcal{B}}, \bar{x}_{\mathcal{N}})$ associated with $\{\mathcal{B}, \mathcal{N}\}$ is *basic*, if $\bar{x}_{\mathcal{N}} = 0$. The basic primal solution is nondegenerate if

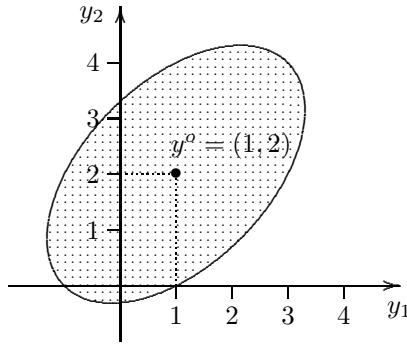
$$|\bar{x}_j| > 0 \quad \text{for all } j \in \mathcal{B}. \quad (3.52)$$

By definition, the dual solution (\bar{y}, \bar{u}) associated with $\{\mathcal{B}, \mathcal{N}\}$ is *basic* if $\bar{u}_{\mathcal{B}} = 0$. The dual basic solution is nondegenerate if $|\bar{u}_j| > 0$ for all $j \in \mathcal{N}$. A dual basic solution is nondegenerate and *feasible*, if and only if

$$\bar{u}_{\mathcal{N}} > 0. \quad (3.53)$$

The primal and dual basic solutions \bar{x} and (\bar{y}, \bar{u}) are called *complementary* because $|u_j| > 0$ implies $x_j = 0$ and $|x_j| > 0$ implies $u_j = 0$; see (3.51).

Definition (Interior Solution): A solution (y, u) is said to be an *interior solution for the dual* (\mathcal{D}) if $u > 0$, implying that interior solutions are strictly in the interior of the u space.

Figure 3-6: Ellipsoid Centered at $(1, 2)$

Recall that quadratic expressions, such as $y_1^2 + y_2^2$ and $y_1^2 + y_1y_2 + y_2^2$ or more generally y^TMy , are positive-definite if and only if $y^TMy > 0$ for all $y \neq 0$.

Definition (Ellipsoid): Let M be any positive-definite matrix; then the set of points

$$\left\{ y \in \mathfrak{R}^m \mid (y - y^o)^T M (y - y^o) \leq k \right\}, \quad k > 0, \quad (3.54)$$

defines an *ellipsoid* with center at y^o . Another way to express an ellipsoidal region is by the set of points

$$\left\{ y \in \mathfrak{R}^m \mid \|A^T(y - y^o)\| \leq \bar{k} \right\}, \quad \bar{k} > 0, \quad (3.55)$$

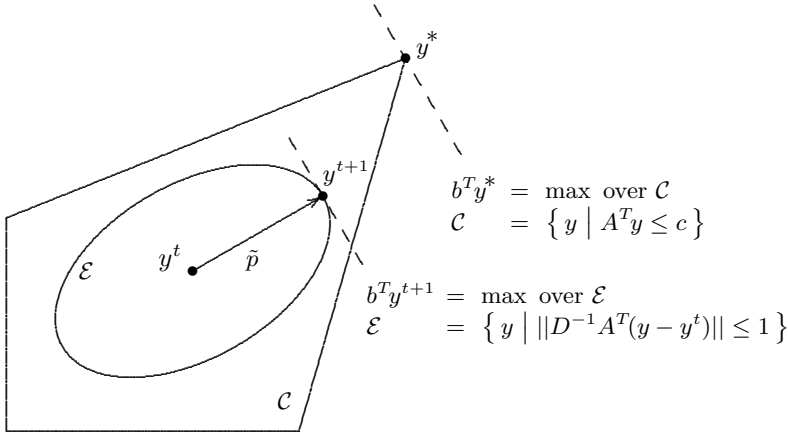
where A is any $m \times n$ matrix of rank m and $m \leq n$.

▷ **Exercise 3.27** Show that the set of y given by (3.55) is the same as the ellipsoidal region defined by (3.54) with $M = AA^T$ and $k = \bar{k}^2$. Prove that if A is of rank m , then M is positive-definite. If $\text{rank}(M) < m$, then prove that M is positive semi-definite; i.e., $y^TMy \geq 0$ for all y .

▷ **Exercise 3.28** Prove that if M in (3.54) is positive semi-definite then (3.54) defines an ellipsoid in a lower-dimensional space than \mathfrak{R}^m .

▷ **Exercise 3.29** Show that the expression

$$(y_1 - 1)^2 - (y_1 - 1)(y_2 - 2) + (y_2 - 2)^2 \leq 4$$

Figure 3-7: Ellipsoid Subproblem Centered at y^t

defines a two-dimensional ellipsoid whose boundary is an ellipse

$$(y_1 - 1)^2 - (y_1 - 1)(y_2 - 2) + (y_2 - 2)^2 = 4$$

having a center $y^o = (1, 2)^T$. The shaded region in Figure 3-6 is a graph of this ellipsoid.

- ▷ **Exercise 3.30** Find the translation and a rotation about the origin that transforms the ellipse in Exercise 3.29 so that its resulting equation is in the form

$$\frac{x_1^2}{a_1^2} + \frac{x_2^2}{a_2^2} = 1.$$

3.2.1 DIKIN'S ALGORITHM

Each iteration t of Dikin's affine algorithm applied to the dual problem starts with an interior solution y^t to the dual (\mathcal{D}), and looks for an interior point $y = y^{t+1}$ (see Figure 3-7) where the objective $b^T y$ is maximized over an *ellipsoidal region* \mathcal{E} centered at y^t , that is,

$$\mathcal{E} = \left\{ y \in \mathfrak{R}^m \mid \|D^{-1} A^T (y - y^t)\| \leq 1 \right\}, \quad (3.56)$$

where $D = D^t = [D_{ij}^t]$ is the diagonal matrix

$$D = \text{Diag}(u^t).$$

The matrix D has a positive diagonal $u^t > 0$ corresponding to the current interior solution (y^t, u^t) . A fundamental property of this ellipsoid, which we will prove later (Lemma 3.9), is that it lies *strictly* in the interior of the feasible dual region

$$C = \{ y \in \mathfrak{R}^m \mid A^T y \leq c \} \quad (3.57)$$

(except under very special circumstances it may not be strictly in the interior, that is, it may touch the boundary; see Exercises 3.31 and 3.37). Once $(y^{t+1}, u^{t+1} = c - A^T y^{t+1} > 0)$ is found, a new ellipsoid is constructed and the process is repeated. This generates a sequence of points y^1, y^2, \dots, y^t , which we will show converges to y^* , the optimal dual solution. We will show that the difference between the optimal value of the objective $v^* = b^T y^*$ and the value of the objective at the current iterate $v_t = b^T y^t$ decreases asymptotically by a factor ρ that approaches $1 - 1/\sqrt{m}$ as $t \rightarrow \infty$.

▷ **Exercise 3.31** Graph the two-dimensional ellipsoid defined by (3.56) and centered at $y^0 = (0, 0)$ for the dual feasible region defined by $y_1 \leq 1$ and $y_2 \leq 1$. Show analytically that this ellipsoid is not strictly in the interior of the dual feasible region but touches its boundaries.

▷ **Exercise 3.32** Graph the two-dimensional ellipsoid defined by (3.56) and centered at $y^0 = (0, 0)$ when the feasible region is the square region defined by $-1 \leq y_1 \leq 1$ and $-1 \leq y_2 \leq 1$.

Algebraically, the ellipsoid subproblem (\mathcal{EP}), shown in (3.58), at iteration t is to find $y = y^{t+1} \in \mathfrak{R}^m$, which

$$(\mathcal{EP}) \quad \begin{array}{ll} \text{Maximizes} & v = b^T y \\ \text{subject to} & \|D^{-1}A^T(y - y^t)\| \leq 1. \end{array} \quad (3.58)$$

The algorithm is iterated with (y^{t+1}, u^{t+1}) replacing (y^t, u^t) . The iterate also computes a primal (not necessarily feasible) solution x^t in addition to the maximizing point y^{t+1} in the ellipsoid. These iterates x^t and y^{t+1} are computed as shown in Algorithm 3.2.

Algorithm 3.2 (Dikin's Algorithm) Given a dual interior solution (y^o, u^o) where $u^o > 0$. Given a convergence tolerance $\epsilon > 0$. Set $t \leftarrow 0$.

1. Find a Search Direction p :

$$p = (AD^{-2}A^T)^{-1}b, \quad D = D^t = \text{diag}(u^t). \quad (3.59)$$

2. Determine a Primal Solution x^t :

$$x^t = D^{-2}A^T p \quad (3.60)$$

3. Update the Dual Solution for the Next Iteration:

$$y^{t+1} = y^t + \tilde{p} = y^t + |(b^T p)^{-1/2}| p, \quad (3.61)$$

$$u^{t+1} = c - A^T y^{t+1}. \quad (3.62)$$

4. Convergence Test: If

$$\|y^{t+1} - y^t\| \leq \epsilon$$

stop and report the primal solution x^t and dual solution (y^{t+1}, u^{t+1}) as “optimal.”

5. Set $t \leftarrow t + 1$ and go to Step 1.

▷ **Exercise 3.33** Show that y^{t+1} given by (3.61) is the optimal solution to the ellipsoid problem (3.58).

▷ **Exercise 3.34** If the dual has an unbounded maximum, i.e., it is primal infeasible, then Step 4 of Algorithm 3.2 will never test optimal. Suggest a modification that detects this.

▷ **Exercise 3.35** Suggest a method for determining an initial (starting) dual interior solution (y^o, u^o) . If the dual were infeasible, how would you detect it in your technique?

We will prove that y^{t+1} lies in the interior of (\mathcal{C}) when $n > m$ and that x^t , the primal solution, satisfies $Ax^t = b$ but not necessarily $x^t \geq 0$. As $t \rightarrow \infty$, we will show that x^t tends toward the unique optimal basic feasible solution under our nondegeneracy assumption and (y^t, u^t) tends towards its complementary optimal-dual basic feasible solution. Until convergence the dual solution is strictly in the interior and is given by:

$$u^t = c - A^T y^t > 0. \quad (3.63)$$

Comment: In practical applications, in order to speed up the rate of convergence one often takes bigger steps than $\tilde{p} = |(b^T p)^{-1/2}|p$. Instead of (3.61),

$$\text{adjusted } y^{t+1} = y^t + \alpha |(b^T p)^{-1/2}|p, \quad \alpha \geq 1,$$

where α is chosen so that the adjusted y^{t+1} is, say, .9 of the way in the direction p from the unadjusted y^{t+1} to the boundary of \mathcal{C} ; see (3.57). The proof of convergence is almost exactly the same. For our development, however, we will assume $\alpha = 1$.

3.2.2 CONVERGENCE OF DIKIN'S ALGORITHM

In this section, to simplify the proof, we prove that Dikin's algorithm converges under the following rather restrictive *assumptions*:

A1 $b \neq 0$, $c \neq 0$, $n > m$, and every subset of m columns from A has rank m .

A2 An interior feasible dual solution y^o is given.

A3 A feasible primal solution exists but is not specified.

A4 Every feasible dual basic solution is nondegenerate.

A5 Every primal basic solution, feasible or not, is nondegenerate.

We first show that the steps of Dikin's algorithm are valid; we do this by using two lemmas that show several interesting properties of the steps of Dikin's algorithm. Next we show that the algorithm converges to an optimal feasible solution by proving several interesting lemmas.

THE STEPS OF DIKIN'S ALGORITHM ARE VALID

LEMMA 3.8 (Properties) *In Dikin's algorithm the following hold:*

$$D^{-1} \text{ exists.} \quad (3.64)$$

$$(AD^{-2}A^T)^{-1} \text{ exists and is positive-definite.} \quad (3.65)$$

$$Ax^t = b. \quad (3.66)$$

$$\|Dx^t\|^2 = p^T b > 0. \quad (3.67)$$

$$\tilde{p} = |(b^T p)^{-1/2}| p \text{ is an improving direction, i.e., } b^T y^{t+1} > b^T y^t. \quad (3.68)$$

$$u^{t+1} = u^t - \|Dx^t\|^{-1}(D^2 x^t). \quad (3.69)$$

Proof. We first show (3.64). We are given inductively by (3.63) that y^t is an interior point meaning $u^t = c - A^T y^t > 0$. Thus $D = \text{Diag}(u^t)$ has a positive diagonal and therefore D^{-1} and D^{-2} exist and have positive diagonals.

We next show (3.65). The rank of A by (A1) is m and so the rank of AD^{-1} (which rescales the columns j of A by $(1/u_j^t) > 0$) is also of rank m ; hence $AD^{-2}A^T$ is an m by m symmetric positive-definite matrix of rank m (see Exercise 3.27), implying $(AD^{-2}A^T)^{-1}$ exists.

To show (3.66), we note that the iterates satisfy certain important relationships. First

$$Ax^t = b, \quad (3.70)$$

which is obtained by substituting the expression for p from (3.59) into (3.60) and multiplying by A , thus:

$$Ax^t = (AD^{-2}A^T)(AD^{-2}A^T)^{-1}b = b \quad (3.71)$$

which are valid steps since D^{-1} and $(AD^{-2}A^T)^{-1}$ exist.

To show (3.67), we note that $x^t \neq 0$ because if $x^t = 0$ in (3.70) it would imply $b = 0$ contrary to (A1). Since $u^t > 0$ and $x^t \neq 0$:

$$Dx^t = (u_1^t x_1^t, u_2^t x_2^t, \dots, u_n^t x_n^t)^T \neq 0, \quad \Rightarrow \quad \|Dx^t\| > 0. \quad (3.72)$$

Also note that

$$\begin{aligned} 0 < \|Dx^t\|^2 &= (x^t)^T D^2 x^t \\ &= (p^T A D^{-2}) D^2 (D^{-2} A^T p) \end{aligned} \quad (3.73)$$

$$\begin{aligned} &= p^T (A D^{-2} A^T) p \\ &= b^T (A D^{-2} A^T)^{-1} (A D^{-2} A^T) p \end{aligned} \quad (3.74)$$

$$= b^T p, \quad (3.75)$$

where (3.73) follows from (3.60), and (3.74) follows from (3.59).

To show that (3.68) is true, note that (3.67) implies that it is legal to use $|(p^T b)^{-1/2}|$ as a factor in (3.68) and in the computation of y^{t+1} by (3.61) since $p^T b > 0$. Multiplying (3.61) on the left by b^T and noting (3.67), we get

$$b^T y^{t+1} - b^T y^t = |(b^T p)^{1/2}| = \|Dx^t\| > 0, \quad (3.76)$$

that is, $\tilde{p} = |(b^T p)^{-1/2}|p$ is a strictly improving direction.

Finally, to show (3.69), multiply (3.61) on the left by A^T and substitute $c - u^t$ for $A^T y^t$ and $c - u^{t+1}$ for $A^T y^{t+1}$, to obtain from (3.63):

$$u^{t+1} = u^t - A^T |(b^T p)^{-1/2}|p \quad (3.77)$$

$$= u^t - \|Dx^t\|^{-1} A^T p \quad (3.78)$$

$$= u^t - \|Dx^t\|^{-1} (D^2 x^t) \quad (3.79)$$

where (3.78) follows from (3.75), and (3.79) follows by multiplying (3.60) by D^2 . Hence (3.69) is true. \blacksquare

LEMMA 3.9 (The Ellipsoid is Strictly in the Dual Interior) *The dual iterates are feasible and lie strictly in the interior and satisfy:*

$$0 < u^{t+1} \leq 2u^t. \quad (3.80)$$

Proof. We first show that $0 < u^{t+1} \leq 2u^t$, by showing first $0 \leq u^{t+1} \leq 2u^t$ and next that the feasible region of (\mathcal{E}) is contained in the interior of the feasible region of the dual feasible region (\mathcal{C}) .

Let $\hat{y} \neq y^t$ be any point that lies in the ellipsoid \mathcal{E} and let $\hat{u} = c - A^T \hat{y}$. Then by the definition of the ellipsoid \mathcal{E} given by (3.56):

$$\sum_{j=1}^n \frac{(A_{\bullet j}^T (\hat{y} - y^t))^2}{(u_j^t)^2} \leq 1. \quad (3.81)$$

From (3.81) and the equation for u^t , (3.63), it follows that

$$\sum_{j=1}^n \frac{(\hat{u}_j - u_j^t)^2}{(u_j^t)^2} \leq 1. \quad (3.82)$$

This implies for each component j :

$$\frac{(\hat{u}_j - u_j^t)^2}{(u_j^t)^2} \leq 1, \quad (3.83)$$

$$-u_j^t \leq \hat{u}_j - u_j^t \leq u_j^t, \quad (3.84)$$

$$0 \leq \hat{u}_j \leq 2u_j^t. \quad (3.85)$$

We next show that $\hat{u} > 0$. Assume on the contrary, that \hat{y} is on the boundary of the dual feasible region (\mathcal{C}) , then there exists a component $j = s$ of $\hat{u} = c - A^T \hat{y}$ such that

$$0 = \hat{u}_s = c_s - A_{\bullet s}^T \hat{y}. \quad (3.86)$$

But $\hat{u}_s = 0$ holding implies

$$\frac{(\hat{u}_s - u_s^t)^2}{(u_s^t)^2} = \frac{(0 - u_s^t)^2}{(u_s^t)^2} = 1. \quad (3.87)$$

Since the sum of all the terms in (3.82) must be less than or equal to 1, and, from (3.87) the term for $j = s$ equals 1 this implies that all the terms except for $j = s$ must vanish. Therefore, applying (3.63):

$$0 = u_j^t - \hat{u}_j = A_{\bullet j}^T (\hat{y} - y^t), \quad \forall j \neq s. \quad (3.88)$$

However, choosing B to be any basic set of m indices j , which excludes $j = s$, we have by the nonsingularity of A_B (see assumption (A1)):

$$A_B^T (\hat{y} - y^t) = 0 \quad \Rightarrow \quad \hat{y} = y^t.$$

However, our contrary assumption that \hat{y} is a boundary point of (\mathcal{C}) implies that $y^t = \hat{y}$ is also a boundary point, which contradicts the assumption that y^t is an interior point of (\mathcal{C}) . Therefore \hat{y} , which is any point that lies in the ellipsoid (\mathcal{E}) is an interior point of (\mathcal{C}) , i.e., $\hat{u} > 0$. In particular, $\hat{y} = y^{t+1}$ is not on the boundary of (\mathcal{C}) implying $u^{t+1} > 0$. This, together with (3.85), implies:

$$0 < u^{t+1} = c - A^T y^{t+1} \leq 2u^t. \quad (3.89)$$

This establishes what we set out to prove. ■

▷ **Exercise 3.36** Prove in Lemma 3.9 that in fact $u^{t+1} < 2u^t$.

▷ **Exercise 3.37** Since we just proved that the ellipsoid (\mathcal{E}) lies strictly in the interior of the dual feasible region and we know from Exercise 3.31 that if $n = m$ this may not be true, discover where in the proof for Lemma 3.9 we made use of the assumption that $m < n$.

▷ **Exercise 3.38** Note that in the proof of Lemma 3.9 we have not made use of the primal nondegeneracy assumption (A5). Suppose in assumption (A1) we allow $m \leq n$. Prove that the optimal solution y^{t+1} to (3.58) would be strictly in the interior of the dual feasible region if assumption (A5) holds.

THEOREM 3.10 (Steps of Dikin's Algorithm are Legal) *Steps (3.59), (3.60), and (3.61) of Dikin's algorithm are legal and hence can be executed iteratively.*

▷ **Exercise 3.39** Apply Lemmas 3.8 and 3.9 to prove Theorem 3.10.

PROOF OF CONVERGENCE

Having completed the proof that the detailed steps of Dikin's algorithm are legal, we will now use Lemmas 3.8 and 3.9 to prove convergence of the algorithm. Before formally proving the theorem we will sketch the proof.

We start by showing that the primal-dual iterates generated by Dikin's algorithm are complementary. We next use this to show that, given a basic/nonbasic partition, if convergence of the primal basic variables occurs, or of the primal nonbasic variables, or of the dual nonbasic variables, or of the dual basic variables, then in fact convergence occurs to a basic primal solution and a corresponding basic dual feasible solution. Next we show that convergence does in fact occur and that, in fact, the primal basic solution and basic dual feasible solutions are complementary. What is left to show is that the basic primal solution is feasible and that the solution is optimal.

About each of the dual basic feasible solutions (extreme points) we construct a ball of radius δ , where δ is chosen so small that none of the balls overlap. We claim that only a finite number of iterates y^t lie outside all of the balls because if an infinite number lay outside the balls a subset of them would converge to a point not in any of the balls, but, as we have already shown, convergence of the subset must be to an extreme point, which is a contradiction. Therefore every convergent subsequence must be to the extreme point centers of some balls. However, we show that there cannot be two subsequences converging to two or more different extreme points. We, next show that the extreme point it converges to is in fact primal feasible and primal/dual optimal.

THEOREM 3.11 (Complementarity of Primal-Dual Iterates) *The primal-dual iterates $\{u^t, x^t\}$ tend toward complementarity, i.e., for each $j = 1, \dots, n$*

$$u_j^t x_j^t \rightarrow 0 \quad \text{as } t \rightarrow \infty. \quad (3.90)$$

Proof. Note that $b^T y^t$ is strictly monotonically increasing by (3.68), Lemma 3.9, and has a finite upper bound because by assumption (A3) primal feasible solutions exist. Therefore from (3.76) we have

$$b^T y^{t+1} - b^T y^t = \|Dx^t\| \rightarrow 0 \quad \text{as } t \rightarrow \infty. \quad (3.91)$$

where $\|Dx^t\|^2 = \sum_{j=1}^n (u_j^t x_j^t)^2$ from (3.72). This implies that $u_j^t x_j^t \rightarrow 0$ for all components j . ■

Note that our assumptions (A1)–(A5) do not imply that the dual space is bounded; and, that we do not require boundedness of the dual feasible space for our proofs as do some proofs (of Dikin's algorithm) in the literature.

Definition (Using t in S): Let $S = \{t_1, t_2, \dots, t_i, \dots\}$ be an increasing infinite subsequence of iterations $t = \{1, 2, \dots\}$. By the phrase *using t in S* , we mean choosing some increasing infinite subsequence of the subsequence S .

THEOREM 3.12 (Convergence to Basic Solutions) *Given some partition $(\mathcal{B}, \mathcal{N})$ of the column indices of A . Using t in S , if $x_B^t \rightarrow \bar{x}_B$, or $x_N^t \rightarrow \bar{x}_N$, or $u_B^t \rightarrow \bar{u}_B$, or $u_N^t \rightarrow \bar{u}_N$, then (\bar{x}_B, \bar{x}_N) is a basic primal solution and (\bar{u}_B, \bar{u}_N) is a basic dual feasible solution.*

Proof. The proof consists in showing the implications outlined in the following figure:

$$\begin{array}{ccc} x_B^t \rightarrow \bar{x}_B & \longleftarrow & x_N^t \rightarrow \bar{x}_N = 0 \\ \Downarrow & \boxed{\text{Using } t \text{ in } S} & \Uparrow \\ u_B^t \rightarrow \bar{u}_B = 0 & \Longrightarrow & u_N^t \rightarrow \bar{u}_N > 0 \end{array}$$

We will show each implication separately.

1. Suppose $x_N^t \rightarrow \bar{x}_N = 0$ using t in S ; then we get

$$x_B^t = B^{-1}(b - A_N x_N^t) \rightarrow B^{-1}b = \bar{x}_B \text{ using } t \text{ in } S,$$

where $B = A_B$.

2. Suppose $x_B^t \rightarrow \bar{x}_B$ using t in S . By the nondegeneracy assumption A5, $|\bar{x}_i| > 0$ for all $i \in \mathcal{B}$. Hence $|\bar{x}_i^t| > \epsilon > 0$ for all $t > \text{some fixed } t^o$. Then by the Complementarity Theorem 3.11 we get $u_B^t \rightarrow 0 = \bar{u}_B$.
3. Suppose that $u_B^t \rightarrow \bar{u}_B = 0$ using t in S , then

$$u_N^t = c_N - ((c_B - u_B^t)^T B^{-1} A_N)^T \rightarrow c_N - ((c_B^T B^{-1}) A_N)^T = \bar{u}_N.$$

Since $u_N^t > 0$, this implies $\bar{u}_N \geq 0$ and this in turn implies by nondegeneracy of the dual solution (assumption A4), that $\bar{u}_N > 0$ and $u_N^t > \epsilon > 0$ for some ϵ for $t > t^o$ in S .

4. Suppose that $u_N^t \rightarrow \bar{u}_N > 0$ using t in S , then by the Complementarity Theorem 3.11 we get $x_N^t \rightarrow \bar{x}_N = 0$.

This completes the proof. ■

LEMMA 3.13 (Convergence of Dual Iterates) *Let $u^1, u^2, \dots, u^t, \dots$ be the infinite sequence of u^t generated by Dikin's algorithm. Let $S_0 = \{1, 2, \dots, t, \dots\}$. Then there exists an infinite subsequence $S_* = \{t_1^*, t_2^*, \dots\} \supseteq S_0$ such that for each component j using t in S_* either $u_j^t \rightarrow 0$ or $u_j^t \geq \epsilon_*$ for some fixed $\epsilon_* > 0$.*

Proof. Let n be the number of columns of A . Define inductively for $j = 1, 2, \dots, n$ the nested set of infinite subsequences $S_0 \supseteq S_1 \supseteq S_2 \cdots \supseteq S_n$ as follows. For the first component $j = 1$ of the u^t , let $\epsilon_1 = \inf u^t$ using t in S_0 . Clearly $\epsilon_1 \geq 0$ since $u^t > 0$. If $\epsilon_1 > 0$ then $u_1^t \geq \epsilon_1$ for all t and we set $S_1 = S_0$. If, on the other hand, $\epsilon_1 = 0$, then there exists a subsequence $S_1 \subseteq S_0$ such that $u_1^t \rightarrow 0$ using t in S_1 . The process is then repeated for component $j = 2$ with S_1 instead of S_0 generating $\epsilon_2 = \inf u_2^t$ using t in S_1 . If $\epsilon_2 > 0$, then we set $S_2 = S_1$; if $\epsilon_2 = 0$ using t in S_1

then we set $S_2 \subseteq S_1$ such that $u_j^t \rightarrow 0$ using t in S_2 . In this manner, we obtain for components $j = 1, \dots, n$, $\epsilon_1, \epsilon_2, \dots, \epsilon_n$ and subsequence $S_1 \supseteq S_2 \supseteq \dots \supseteq S_n$. Letting $\epsilon_* = \min \epsilon_j$ for $\epsilon_j > 0$ and letting $S_* = S_n$, we obtain the desired result. ■

Comment: The proof is an existence proof. Since it involves finding the inf (infimum) of an infinite sequence, it cannot be executed as an algorithm on a computer in a finite lifetime.

THEOREM 3.14 (Convergence to Complementary Basic Solutions) *Let S_0 be any infinite subsequence of iterations. There exists a partition $\{\mathcal{B}, \mathcal{N}\}$ and infinite subsequence $S_* = \{t_1^*, t_2^*, \dots\}$ contained in S_0 such that $x^t = (x_B^t, x_N^t)$ tends to the primal basic solution $(\bar{x}_B, \bar{x}_N = 0)$ and $(y^t; u_B^t, u_N^t)$ tends to the complementary dual basic feasible solution $(\bar{y}; \bar{u}_B = 0, \bar{u}_N > 0)$ using t in S_* .*

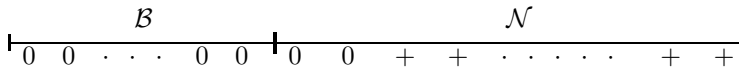
Proof. In Lemma 3.13, let \bar{m} be the number of indices j such that $u_j^t \rightarrow 0$ using t in S_* . For the remaining $n - \bar{m}$ indices j , $u_j^t \geq \epsilon_* > 0$ for all t in S_* . One of the following two cases holds.

Case 1: $\bar{m} \geq m$ of the u_j s tend to 0.

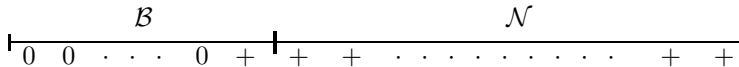
Case 2: $n - \bar{m} > n - m$ of the u_j s are bounded away from 0.

We will show that in either case the (x^t, u^t) converge to complementary basic feasible solutions using t in S_* where the basic partition $\{\mathcal{B}, \mathcal{N}\}$ is defined as follows.

Case 1: Since $\bar{m} \geq m$, there exists a subset \mathcal{B} of m indices such that for $j \in \mathcal{B}$, $u_j^t \rightarrow 0$ using t in S_* . The basic partition $\{\mathcal{B}, \mathcal{N}\}$ is then defined by letting \mathcal{N} denote the remaining indices j .



Case 2: Since $n - \bar{m} > n - m$, there exists a subset \mathcal{N} of $n - m$ indices j such that for $j \in \mathcal{N}$, $u_j^t \geq \epsilon_*$ for all t in S_* . The basic partition $\{\mathcal{B}, \mathcal{N}\}$ is then defined by letting \mathcal{B} denote the remaining indices j .



In Case 1 there is a basic partition $\{\mathcal{B}, \mathcal{N}\}$ such that $u_B^t \rightarrow 0 = \bar{u}_B$ using t in S_* . The result follows from Theorem 3.12.

In Case 2 there is a basic partition $\{\mathcal{B}, \mathcal{N}\}$ such that for $j \in \mathcal{N}$ that $u_j^t \geq \epsilon_* > 0$ for all t in S_* . The result follows from Theorem 3.12. ■

Convergence as $t \rightarrow \infty$: Letting S_0 be any arbitrary infinite subset $t_1 < t_2 < \dots < \infty$, we have shown convergence to a basic primal solution \bar{x}^* and its complementary dual basic feasible solution (\bar{y}^*, \bar{u}^*) using t in some infinite subset $S_* \subset S_0$. *Our goal (see Theorem 3.19) will be to show that $x^t, (y^t, u^t)$ converge to $\bar{x}^*, (\bar{y}^*, \bar{u}^*)$ and \bar{x}^* is basic feasible and primal optimal as is (\bar{y}^*, \bar{u}^*) dual basic feasible and optimal using $\{t\} = \{1, 2, \dots\}$.*

But first some definitions and easy to prove lemmas.

Definition (Lying in a Ball): In Theorem 3.14, choose $S_0 = \{1, 2, \dots, \infty\}$ and let $T_* = S_*$ be the subsequence obtained for this choice of S_0 where $x^t, (y^t, u^t)$ converge to a primal basic solution and a complementary dual feasible solution. Let (\bar{x}^*, \bar{y}^*) be the basic (which we will show is not necessarily feasible) primal solution and complementary basic dual feasible solution that (x^t, y^t) converge to using t in T_* .

Denote by $(\bar{x}^\nu, \bar{y}^\nu)$ for $\nu = 1, \dots, q$ the finite set q of all other basic (not necessarily feasible) primal and complementary basic dual feasible solutions. Let $2\delta_0$ be the shortest distance between any two distinct extreme points \bar{y} ; i.e.,

$$\delta_0 = (1/2) \min \|\bar{y}^i - \bar{y}^j\|, \quad i \neq j, \quad (3.92)$$

where $i, j \in \{*, 1, \dots, q\}$.

We say that all points y that are within a $\delta > 0$ distance of an extreme point \bar{y} are *lying in a ball of radius δ with a center at \bar{y} or lying in a δ -neighborhood of \bar{y}* denoted by $N_\delta(\bar{y})$:

$$N_\delta(\bar{y}) = \{y : \|y - \bar{y}\| \leq \delta\}. \quad (3.93)$$

An iterate y^t is said to *lie outside the nonoverlapping balls $N_\delta(\bar{y}^\nu)$* if

$$\|y^t - \bar{y}^\nu\| > \delta \text{ for all } \nu = *, 1, \dots, q.$$

▷ **Exercise 3.40** Show that for $0 < \delta < \delta_0$ defined in (3.92) the δ -neighborhoods of the extreme points \bar{y}^* and \bar{y}^ν for $\nu = 1, \dots, q$ have no points in common.

LEMMA 3.15 (Count of y^t that are Outside of the Balls is Finite.) *Given any δ , $0 < \delta < \delta_0$, where δ_0 defined by (3.92) and given balls $N_\delta(\bar{y}^\nu)$ where \bar{y}^ν is one of the dual basic feasible solutions for $\nu = *, 1, \dots, q$. Then the count of y^t , for $t = 1, 2, \dots, \infty$, that are outside of the balls is finite.*

Proof. Let $Y_0 = \{y^{t_1}, y^{t_2}, \dots\}$ be the subsequence of all y^t lying outside the balls and let $S_0 = \{t_1, t_2, \dots\}$ be the corresponding subsequence of t . If, on the contrary, the count of t in S_0 is infinite, then by Theorem 3.14 there would be an infinite subsequence $S_* \subset S_0$ such that, using t in S_* , $y^t \rightarrow \bar{y}^{\nu_0}$, for some dual basic

feasible solution \bar{y}^{ν_0} . This implies an infinity of t in S_0 whose y^t are in the ball $\|y - \bar{y}^{\nu_0}\| \leq \delta$, contrary to the construction of Y_0 as the set of y^t lying outside all the balls $N_\delta(\bar{y}^\nu)$, for $\nu = *, 1, \dots, q$. ■

LEMMA 3.16 (Iterates y^t Converge to an Extreme Point \bar{y}^{ν_0}) *If the count of y^t is infinite in the ball $N_{\delta_0}(\bar{y}^{\nu_0})$ for some extreme point \bar{y}^{ν_0} , then $y^t \rightarrow \bar{y}^{\nu_0}$ using y^t in the ball.*

Proof. Let Y_0 be the subsequence of y^t lying in the ball $N_{\delta_0}(\bar{y}^{\nu_0})$ for some extreme point \bar{y} , but outside some smaller concentric ball $N_\delta(\bar{y}^{\nu_0})$, $0 < \delta < \delta_0$. From Lemma 3.15 the subsequence Y_0 must be finite whatever be $\delta < \delta_0$. It follows that if the count of y^t in $N_{\delta_0}(\bar{y}^{\nu_0})$ is infinite, then given any δ , $0 < \delta < \delta_0$, there exists a t_δ such that for all $t > t_\delta$ all y^t in $N_{\delta_0}(\bar{y}^{\nu_0})$ are also in $N_\delta(\bar{y}^{\nu_0})$. By definition, this is what we mean when we say $y^t \rightarrow \bar{y}^{\nu_0}$ using y^t in $N_{\delta_0}(\bar{y}^{\nu_0})$. ■

LEMMA 3.17 (Iterates y^t Converge to a Primal Basic Solution and Dual Basic Feasible Solution \bar{y}^*) *If for t in S_0 , y^t does not converge as $t \rightarrow \infty$ to the same limit \bar{y}^* using t in T_* then for some $\nu \neq *$ there exists an infinite subsequence $T_1 = \{t_1, t_2, \dots\}$ and a successor subsequence $T_2 = \{t_1 + 1, t_2 + 1, \dots\}$ such that $y^t \rightarrow \bar{y}^*$ using $t = t_k$ in T_1 and $y^{t+1} \rightarrow \bar{y}^\nu$ using $t = t_k + 1$ in T_2 .*

Proof. Assume that $y^t \rightarrow \bar{y}^*$ is not true. Generate infinite subsequences $T'_1 = \{t_k\}$ and $T'_2 = \{t_k + 1\}$ and $\{\nu_k\}$ as follows:

```

initialize  $k := 1$ ;  $s_1 :=$  first  $t$  such that  $y^t \in N_{\delta_0}(\bar{y}^*)$ ;
while  $j < \infty$  do begin
     $t_k :=$  first  $t \geq s_k$  such that  $y^t \in N_{\delta_0}(\bar{y}^*)$  and  $y^{t+1} \in N_{\delta_0}(\bar{y}^\nu)$  for some  $\nu \neq *$ ;
     $\nu_k := \nu$ ;
     $s_k :=$  first  $t > t_k$  such that  $y^t \in N_{\delta_0}(\bar{y}^*)$ ;
     $k := k + 1$ ;
end while;
```

Referring to the definition of T_* (see Page 96), note that s_k always exists since $y^t \rightarrow \bar{y}^*$ using t in T_* . Under the contrary assumption that $y^t \not\rightarrow \bar{y}^*$ it follows that t_k must also exist; otherwise $s_k, s_{k+1}, s_{k+2}, \dots$ would all belong to $N_{\delta_0}(\bar{y}^*)$, implying by Lemma 3.15 that $y^t \rightarrow \bar{y}^*$ as $t \rightarrow \infty$. Therefore, since the subsequence $\{\nu_k\}$ is infinite and there are at most q different values that the ν_k can assume, there exists a $\nu \neq *$ such that there is an infinite subsequence of $t \in T'_1$ whose y^t are in $N_{\delta_0}(\bar{y}^*)$ and whose successor subsequence y^{t+1} are in $N_{\delta_0}(\bar{y}^\nu)$. Hence, there exists a subsequence $T_1 \subset T'_1$ of t_k and successor sequence $T_2 \subset T'_2$ of $t_k + 1$ such that $y^t \rightarrow \bar{y}^*$ using t in T_1 and $y^{t+1} \rightarrow \bar{y}^\nu$, where $\nu \neq *$ using t in T_2 . ■

THEOREM 3.18 (Iterates y^t Converge to the Same Extreme Point) *A subsequence y^t and a successive subsequence y^{t+1} cannot tend to different limits \bar{y}^* and \bar{y}^ν , $\nu \neq *$ implying as $t \rightarrow \infty$, (x^t, y^t) converge to primal basic and complementary dual basic feasible solutions (\bar{x}^*, \bar{y}^*) .*

Proof. Let $\{\mathcal{B}, \mathcal{N}\}$ be the basic partition associated with \bar{y}^* , $\bar{u}^* = c - A^T \bar{y}^*$ and let $(\widehat{\mathcal{B}}, \widehat{\mathcal{N}})$ be the basic partition associated with \bar{y}^ν , $\bar{u}^\nu = c - A^T \bar{y}^\nu$ where $\nu \neq *$ as implied under the contrary assumption that y^t does not converge. Since $\mathcal{B} \neq \widehat{\mathcal{B}}$, let $j = r$ be an index in \mathcal{B} that is not in $\widehat{\mathcal{B}}$. Then, since for t_k in T_1 (from Lemma 3.17), $y^{t_k} \rightarrow \bar{y}^*$ as $k \rightarrow \infty$, and $y^{t_{k+1}} \rightarrow \bar{y}^\nu$, we have

$$u_r^{t_k} = c_r - A_{\bullet r}^T y^{t_k} \rightarrow c_r - A_{\bullet r}^T \bar{y}^* = \bar{u}_r^* = 0, \quad (3.94)$$

$$u_r^{t_{k+1}} = c_r - A_{\bullet r}^T y^{t_{k+1}} \rightarrow c_r - A_{\bullet r}^T \bar{y}^\nu = \bar{u}_r^\nu > 0, \quad (3.95)$$

where $\bar{u}_r^\nu > 0$ since r is nonbasic with respect to $(\widehat{\mathcal{B}}, \widehat{\mathcal{N}})$ and the dual basic solution is nondegenerate by assumption (A4). Since $u_r^{t_k} \rightarrow 0$, as $k \rightarrow \infty$, it follows from $u_r^{t_{k+1}} \leq 2u_r^{t_k}$ (see Equation (3.80)) that $u_r^{t_{k+1}} \rightarrow 0$, contrary to (3.95). Since the contrary assumption $y^t \rightarrow \bar{y}^*$ and $y^{t+1} \rightarrow \bar{y}^\nu$, $\nu \neq *$ leads to a contradiction, we conclude that $y^t \rightarrow \bar{y}^*$ from which it follows from Theorem 3.12 that $(x^t, y^t, u^t) \rightarrow (\bar{x}^*, \bar{y}^*, \bar{u}^*)$, where $\bar{u}_N^* > 0$. ■

THEOREM 3.19 (Convergence to Optimal Solution) *Dikin's algorithm converges to optimal basic primal and dual solutions.*

Proof. If we can prove that $x^t \rightarrow \bar{x}^*$ is *feasible*, then by Theorem 3.18 we have convergence in the limit to optimal solutions to the primal and dual problems. Since $Ax^t = b$, we only need to prove that $x_j^t \rightarrow x_j^* \geq 0$ for all j . At the t th iteration, we have from (3.79)

$$u_j^{t+1} = u_j^t - \frac{(u_j^t)^2 x_j^t}{\|Dx^t\|} = u_j^t \left(1 - \frac{u_j^t x_j^t}{\|Dx^t\|} \right), \quad (3.96)$$

$D = D^t = \text{diag}(u^t)$. On the contrary, let us assume for some basic index r , that $x_r^t \rightarrow \bar{x}_r^* < 0$ as $t \rightarrow \infty$. There exists a finite \bar{t} such that for all $t > \bar{t}$, $x_r^t < 0$ and therefore, because $u_r^t > 0$:

$$1 - \frac{u_r^t x_r^t}{\|Dx^t\|} > 1 \quad \text{for all } t > \bar{t}. \quad (3.97)$$

Hence, from (3.96), for all $t > \bar{t}$

$$u_r^{t+1} > u_r^t. \quad (3.98)$$

Thus, we have u_r^t strictly increasing for $t = \bar{t} + 1, \bar{t} + 2, \dots, \infty$, contradicting $u_r^t \rightarrow 0$ by Theorem 3.11 for basic index r . This completes the proof. ■

The next theorem tells us how fast Dikin's algorithm converges to an optimum.

THEOREM 3.20 (Ratio of Convergence) *Let $v_t = b^T y^t$ and $v^* = b^T \bar{y}^*$ be the values of the dual objective on iteration t and in the limit as $t \rightarrow \infty$. Then the ratio of convergence ρ_t satisfies*

$$\rho_t = \frac{v^* - v_{t+1}}{v^* - v_t} \leq 1 - \left(\frac{1}{m} + \epsilon_t \right)^{1/2}, \quad v_t < v^*, \quad D = \text{Diag}(u^t), \quad (3.99)$$

where $\epsilon_t \rightarrow 0$ as $t \rightarrow \infty$, i.e., ρ_t is asymptotically $\leq 1 - (1/m)^{1/2}$.

Proof. Let $\{\mathcal{B}, \mathcal{N}\}$ be the basic partition associated with optimal (\bar{x}^*, \bar{y}^*) . We rewrite (3.99) as

$$\rho_t = 1 - \frac{b^T y^{t+1} - b^T y^t}{b^T \bar{y}^* - b^T y^t} = 1 - \frac{\|Dx^t\|}{(u_B^t)^T \bar{x}_B^*}, \quad (u_B^t)^T \bar{x}_B^* > 0, \quad (3.100)$$

where the numerator follows from (3.76) and the denominator follows from $A_B \bar{x}_B^* = b$, $A_B^T y^t = (c_B - u_B^t)$ and $A_B^T \bar{y}^* = c_B$; that is,

$$\begin{aligned} b^T (\bar{y}^* - y^t) &= (\bar{x}_B^*)^T A_B^T (\bar{y}^* - y^t) = (\bar{x}_B^*)^T (c_B - (c_B - u_B^t)) \\ &= (u_B^t)^T \bar{x}_B^* > 0 \end{aligned} \quad (3.101)$$

because $u_B^t > 0$ by (3.80) and $\bar{x}_B^* > 0$ by nondegeneracy. Relabeling the indices $\mathcal{B} = (1, \dots, m)$ then

$$\|Dx^t\| = \left(\sum_{j=1}^n (u_j^t x_j^t)^2 \right)^{1/2} \geq \left(\sum_{i=1}^m (u_i^t x_i^t)^2 \right)^{1/2} \quad (3.102)$$

$$(u_B^t)^T \bar{x}_B^* = \sum_{k=1}^m u_k^t \bar{x}_k^*. \quad (3.103)$$

Thus,

$$\rho_t \leq 1 - \frac{(\sum_{i=1}^m (u_i^t x_i^t)^2)^{1/2}}{\sum_{k=1}^m u_k^t \bar{x}_k^*}. \quad (3.104)$$

Noting that $u_i^t > 0$ and $\bar{x}_k^* > 0$, define λ_i by

$$u_i^t \bar{x}_i^* = \lambda_i \sum_{k=1}^m u_k^t \bar{x}_k^*, \quad \sum_{i=1}^m \lambda_i = 1, \quad \lambda_i > 0, \quad i = 1, \dots, m \quad (3.105)$$

If, in (3.104), we were to replace x_i^t in the numerator by its limit \bar{x}_i^* and then substitute $u_i^t \bar{x}_i^* = \lambda_i \sum_{k=1}^m u_k^t \bar{x}_k^*$ then the factor $\sum_{k=1}^m u_k^t \bar{x}_k^*$ will cancel out with the denominator obtaining:

$$\text{asymptotically } \rho_t \leq 1 - \left(\sum_{i=1}^m (\lambda_i^t)^2 \right)^{1/2} < 1 - (1/m)^{1/2} \quad (3.106)$$

because $\sum_{i=1}^m \lambda_i = 1$.

To make this "proof" a little more rigorous, we rewrite $u_i^t x_i^t$ in the numerator of (3.104) as

$$\begin{aligned} u_i^t x_i^t &= u_i^t \bar{x}_i^* \left(1 + \frac{x_i^t - \bar{x}_i^*}{\bar{x}_i^*} \right) = u_i^t \bar{x}_i^* (1 + \delta_i^t) \\ &= \lambda_i \left(\sum_{k=1}^m u_k^t \bar{x}_k^* \right) (1 + \delta_i^t) \end{aligned}$$

where λ_i is defined by (3.105), $x_i^* > 0$, and $\delta_i^t = (x_i^t - \bar{x}_i^*)/\bar{x}_i^* \rightarrow 0$ as $t \rightarrow \infty$. This yields

$$\begin{aligned} \rho_t &\leq 1 - \left[\sum_{i=1}^m (\lambda_i^t)^2 (1 + \delta_i^t)^2 \right]^{\frac{1}{2}} \\ &\leq 1 - \left[\sum_{i=1}^m (\lambda_i^t)^2 + \sum_{i=1}^m (\lambda_i^t)^2 \delta_i^t (2 + \delta_i^t) \right]^{\frac{1}{2}} \\ &\leq 1 - \left(\frac{1}{m} + \epsilon_t \right)^{1/2} \end{aligned}$$

where $\sum_{i=1}^m (\lambda_i^t)^2 > 1/m$ because $\sum_{i=1}^m \lambda_i^t = 1$, $\lambda_i^t > 0$, and

$$\epsilon_t = \sum_{i=1}^m (\lambda_i^t)^2 \delta_i^t (2 + \delta_i^t) \rightarrow 0 \text{ as } t \rightarrow \infty$$

because $0 < \lambda_i^t \leq 1$. ■

3.3 KARMARKAR'S METHOD

We will present a variation of Karmarkar's method and a simpler proof. His algorithm, like Dikin's, moves through the interior of a polytope. However in deciding at each iteration t how to make a move from x^t to x^{t+1} , in Karmarkar's method, the coordinate system is temporarily changed so that the current iterate x^t becomes y^t at the polytope's "center" in the transformed space and the ellipsoid becomes a hypersphere. The algorithm has a polynomial bound of $O(n^{7/2}\mathcal{L})$ on the number of arithmetic operations, where n is the number of variables and \mathcal{L} is the number of bits required to represent the problem data on the computer.

The rationale for the approach is based on the following observations. When minimizing, one is first tempted to move from the current solution x^t in the direction of *steepest descent* of the objective function (i.e., in the negative gradient of the objective function, which is the same as moving orthogonal to the objective hyperplane $c^T x = \text{constant}$). If the current iterate x^t is an interior point so that $x^t > 0$, such a move will in general violate the constraints $Ax = b$. To adjust for this, one typically moves instead in the direction given by the projection of the negative gradient of the objective onto the hyperplanes $Ax = b$. However, if x^t is close to the boundary hyperplanes, as $x^t = \bar{x}^t$ is in Figure 3-8, very little improvement will occur. On the other hand, if the current iterate happens to be near the "center," such as $x^t = \hat{x}^t$ in Figure 3-8, there could be a big improvement.

3.3.1 DEVELOPMENT OF THE ALGORITHM

One of Karmarkar's key ideas is to view the current iterate $x^t > 0$ in a different coordinate system as being at the center of the polytope. This is done by trans-

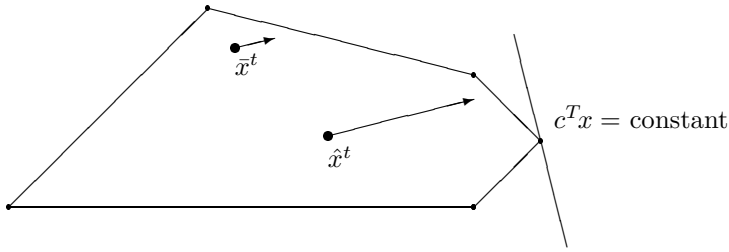


Figure 3-8: Comparison of a Move from a Point \hat{x}^t Near the “Center” Versus a Point \bar{x}^t Near the Boundary

forming the coordinates $x = (x_1, x_2, \dots, x_n)$ to $y = (y_1, y_2, \dots, y_n)$ by means of the mapping \mathcal{T} defined by:

$$y_j = n \frac{x_j/a_j}{\sum_{i=1}^n x_i/a_i} \quad \text{for } j = 1, \dots, n, \quad (3.107)$$

where the current iterate $x^t = a > 0$. Transformations such as (3.107) above or its inverse (3.109) below are called projective transformations; for this reason Karmarkar’s algorithm is often referred to as a *projective algorithm*. If we sum (3.107) for $j = 1, \dots, n$, we obtain, whatever x is,

$$\sum_{j=1}^n y_j = n. \quad (3.108)$$

The inverse mapping can be written

$$x_j = n \frac{y_j a_j}{\sum_{i=1}^n y_i a_i} \quad \text{for } j = 1, \dots, n, \quad (3.109)$$

provided the original x_j s of the linear program always satisfy

$$\sum_{j=1}^n x_j = n. \quad (3.110)$$

In matrix notation, letting $D = \text{Diag}(x^t) = \text{Diag}(a)$, the transformation \mathcal{T} and its inverse are defined by:

$$y = n \frac{D^{-1}x}{e^T D^{-1}x} \quad \text{and} \quad x = n \frac{Dy}{e^T Dy}. \quad (3.111)$$

▷ **Exercise 3.41** Verify that (3.107) and (3.109) are inverse maps provided (3.108) and (3.110) hold.

- ▷ **Exercise 3.42** Prove (see Section 1.2 for the definition of a simplex) that the set of relations:

$$(S) \quad \sum_{j=1}^n x_j = n, \quad x_j \geq 0 \text{ for } j = 1, \dots, n. \quad (3.112)$$

defines an $(n - 1)$ -dimensional simplex $S \subset \mathbb{R}^n$. What are the coordinates of the vertices of S ?

- ▷ **Exercise 3.43** Show that the center of the simplex S defined earlier is $e = (1, 1, \dots, 1)^T$. Show that the transformation (3.111) maps the simplex onto itself and maps the feasible interior point $x^t \in S$ into the center $y^t = e \in S$.

- ▷ **Exercise 3.44** Let S be a simplex defined by (3.112). Show that the radius of the smallest circumscribed sphere is $R = \sqrt{n(n-1)}$ and the radius of the largest inscribed sphere is $r = \sqrt{n/(n-1)}$. Show that the vertices of the simplex S are of the form ne^k where $e^k = (0, \dots, 0, 1, 0, \dots, 0)^T$ is the k th unit vector. Prove that the furthest points from the center e are the vertices ne^k and their distances from the center are $R = \sqrt{n(n-1)}$.

Karmarkar assumes that the original linear program is of the form $\bar{c}^T \bar{x} = \bar{z}$ (min), $e^T \bar{x} = 1$, $A\bar{x} = \bar{b}$, $\bar{x} \geq 0$, which can always be written in the equivalent special form (see Exercise 3.45):

$$\begin{aligned} \text{Minimize} \quad & c^T x = z \\ \text{subject to} \quad & Ax = 0, \quad A \in \mathbb{R}^{m \times n} \\ & e^T x = n, \\ & x \geq 0. \end{aligned} \quad (3.113)$$

He also makes certain additional assumptions:

- A1** A minimizing solution x^* exists and is such that the minimum value of the objective function is *zero*, i.e., $c^T x^* = 0$.
- A2** A starting feasible interior solution $x^o > 0$, $Ax^o = 0$, $e^T x^o = n$ is given. We denote $c^T x^o = z_0$.
- A3** The solution process generates feasible interior iterates x^t ; a feasible solution $x = x^t$ will be declared an “optimal” solution if

$$\frac{z_t}{z_0} = \frac{c^T x^t}{c^T x^o} \leq \frac{e^{-q}}{g^o} \quad (3.114)$$

where $g^o = (x_1^o x_2^o \cdots x_n^o)^{1/n}$ is the geometric mean of $x_1^o, x_2^o, \dots, x_n^o$, q is a given positive constant, and e is the base of the natural logarithm.

One way to transform the general linear program without a convexity constraint to (3.113) and obtain a starting feasible solution will be discussed in Section 3.3.4. It assumes that we know a finite upper bound on the sum of the variables at an optimal solution, which is often known for a particular problem.

- ▷ **Exercise 3.45** Show that a linear program with a convexity constraint, namely, $c^T \bar{x} = \bar{z}$ (min), $\bar{A}\bar{x} = \bar{b}$, $e^T \bar{x} = 1$, $\bar{x} \geq 0$ is equivalent to the linear program of the form $c^T x = z$ (min), $Ax = 0$, $e^T x = n$, $x \geq 0$, see (3.113), where $x = n\bar{x}$, $c = \bar{c}/n$, $A_{\bullet j} = n(\bar{A}_{\bullet j} - \bar{b})$ for $j = 1, \dots, n$.

Applying \mathcal{T} at iteration t , problem (3.113) is transformed to:

$$\begin{aligned} &\text{Minimize} && n\hat{c}^T y / a^T y = z \\ &\text{subject to} && \hat{A}y = 0, \\ & && e^T y = n, \\ & && y \geq 0, \end{aligned} \tag{3.115}$$

where $a = x^t$, $\hat{c} = Dc$, $\hat{A} = AD$, and $D = \text{Diag}(x^t) = \text{Diag}(a)$. Iterate x^t becomes $y^t = e > 0$, where $e = (1, 1, \dots, 1)^T$. Letting the map of optimal x^* be y^* , note that, by assumption A1 on Page 102, $\hat{c}^T y^* = 0$.

- ▷ **Exercise 3.46** Show that $x = x^t$ is mapped into $y^t = e$ satisfying $\hat{A}y^t = 0$, $e^T y^t = n$, $\hat{c}^T y^t = \hat{c}^T e = z_t$, and $y^t > 0$.

However, Karmarkar's next key idea is to avoid solving (3.115) with its fractional objective by considering the alternative problem (3.116), which has the same optimal solution $y^* \geq 0$ and the same initiating solution $y = e > 0$:

$$\begin{aligned} &\text{Minimize} && \hat{c}^T y = \hat{z} \\ &\text{subject to} && \hat{A}y = 0, \\ & && e^T y = n, \\ & && y \geq 0. \end{aligned} \tag{3.116}$$

- ▷ **Exercise 3.47** Show that $\hat{z}_t = z_t$.

- ▷ **Exercise 3.48** *Ph.D. Comprehensive Exam, September 26, 1992, at Stanford.* Prove if x^* is optimal for (3.113), then its map is optimal for (3.116) and conversely if y^* is optimal for (3.116) then its inverse map x^* is optimal for (3.113).

Problem (3.116) is never completely solved to obtain $y = y^*$ instead only an improved solution $y = y^{t+1}$ is obtained by moving some distance ρ from $y^t = e$ in the direction of steepest descent of $\hat{c}^T y$ subject to the equality constraints, i.e.,

$$y^{t+1} = y^t + \rho \frac{p}{\|p\|} = e + \rho \frac{p}{\|p\|} \tag{3.117}$$

where the steepest direction p is the negative of the gradient \hat{c} projected onto the null space of $F = \begin{pmatrix} \hat{A} \\ e^T \end{pmatrix}$ and $\rho = \rho_t > 0$ may be chosen arbitrarily so that $y^{t+1} > 0$. Later, for theoretical reasons, we choose $\rho = 1/2$.

The equality constraints for (3.116) can be written as:

$$Fy = \begin{pmatrix} \widehat{A} \\ e^T \end{pmatrix} y = \begin{pmatrix} 0 \\ n \end{pmatrix}. \quad (3.118)$$

In order for $y = e + \alpha p$ to be feasible, we need, in addition to $e + \alpha p \geq 0$,

$$Fe + \alpha Fp = \begin{pmatrix} 0 \\ n \end{pmatrix}$$

or

$$Fp = 0.$$

This implies that the direction p must be in the null space of F . Any vector can be projected into the null space of F by multiplying it on the left by the projection matrix:

$$P_F = I - F^T(FF^T)^{-1}F. \quad (3.119)$$

The steepest descent direction projected onto the null space of F is

$$\hat{p} = -P_F Dc. \quad (3.120)$$

Hence, the steepest descent direction is computed by:

$$\begin{aligned} p &= -\left[I - \widehat{A}^T(\widehat{A}\widehat{A}^T)^{-1}\widehat{A} - \frac{1}{n}ee^T \right] \hat{c} \\ &= -\hat{c} - \widehat{A}^T(\widehat{A}\widehat{A}^T)^{-1}\widehat{A}\hat{c} + e(z_t/n). \end{aligned} \quad (3.121)$$

▷ **Exercise 3.49** Verify that $P_F x$ projects x into the null space of F . Show that $P_F = P_F^T$ and that $P_F^T P_F = P_F P_F^T = P_F$.

▷ **Exercise 3.50** Show that

$$P_F = I - \widehat{A}^T(\widehat{A}\widehat{A}^T)^{-1}\widehat{A} - \frac{1}{n}ee^T. \quad (3.122)$$

▷ **Exercise 3.51** Prove (3.121).

▷ **Exercise 3.52** Prove $\widehat{A}p = 0$ and $e^T p = 0$.

Iterative Step: The solution y^{t+1} is then mapped back to the x space by $x^{t+1} = nDy^{t+1}/e^T D y^{t+1}$ and the iterative process is repeated with $t \leftarrow t + 1$.

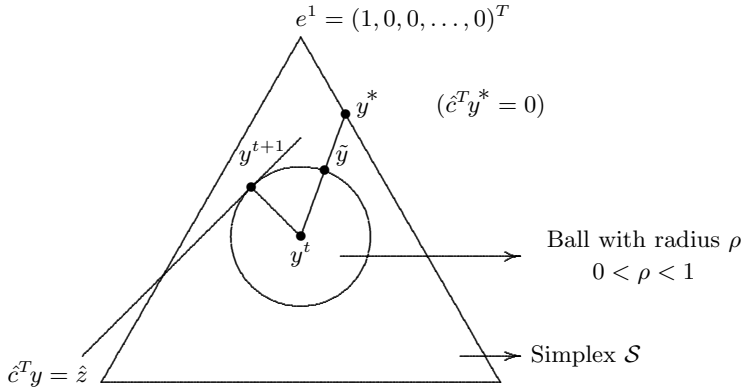


Figure 3-9: Bound for $\hat{c}^T y^{t+1}$

3.3.2 PROOF OF CONVERGENCE

Because the current iterate $y^t = e$ is at the center e of the simplex

$$(\mathcal{S}) \quad \sum_{j=1}^n y_j = n, \quad y_j \geq 0, \quad \text{for } j = 1, \dots, n, \quad (3.123)$$

we can inscribe a hypersphere of radius $\rho < \sqrt{n/(n-1)}$ (see Exercise 3.44) as our *ellipsoid* in \mathcal{S} and consider the *hypersphere* problem

$$(\mathcal{H}) \quad \begin{array}{ll} \text{Minimize} & \hat{c}^T y = \hat{z} \\ \text{subject to} & \hat{A} y = 0, \\ & e^T y = n, \\ & \|y - e\| \leq \rho. \end{array} \quad (3.124)$$

This problem is a variant of the classical least-squares problem subject to constraints and is trivial to solve; the optimal solution is

$$y^{t+1} = y^t + \frac{\rho}{\|p\|} p \quad (3.125)$$

where the steepest direction p is the negative of the gradient \hat{c} projected onto the null space of $\begin{pmatrix} \hat{A} \\ e^T \end{pmatrix}$ and $0 < \rho \leq \sqrt{n/(n-1)}$. As long as $\rho < \sqrt{n/(n-1)}$, we know that the ball $\|y - e\| \leq \rho$ lies strictly in the interior of \mathcal{S} , i.e., $y > 0$. Therefore all y satisfying (3.124) are strictly interior feasible solutions of (3.116).

▷ **Exercise 3.53** Prove (3.125).

- ▷ **Exercise 3.54** The “ball” shown in Figure 3-9 is the intersection of the n -dimensional ball $\|y - e\| \leq \rho$ with the hyperplane $\sum_j y_j = n$. Show that this intersection is an $(n - 1)$ -dimensional hyperspheroid with radius ρ .
- ▷ **Exercise 3.55** Assume \hat{A} is of full rank. Show that the intersection of the n -dimensional ball $\|y - e\| \leq \rho$ with the hyperplanes $\hat{A}y = 0$ and $\sum_j y_j = n$ is an $(n - m - 1)$ -dimensional hyperspheroid with radius ρ .

The iterative step is to move to y^{t+1} , the optimal solution to (\mathcal{H}) for some choice of $\rho < \sqrt{n/(n-1)}$. For reasons that will become clear later we will choose $\rho = 1/2$. This minimizing point y^{t+1} lies on the boundary surface of the hypersphere. To see why, note that an optimal point y^* of (3.116) exists that is an extreme point and hence lies on the boundary of the simplex \mathcal{S} and hence lies outside the interior of the sphere. If on the contrary, y^{t+1} were an interior point of the hypersphere, an improved solution lying on the boundary of the sphere could be found along the line joining y^{t+1} to y^* , a contradiction. Without explicitly computing p by (3.121) and $\hat{c}^T y^{t+1}$ by (3.125), it is easy to prove the following important result.

LEMMA 3.21 (Bound on $\hat{c}^T y^{t+1}$)

$$\hat{c}^T y^{t+1} \leq \left(1 - \rho/\sqrt{n(n-1)}\right) z_t < (1 - \rho/n) z_t. \quad (3.126)$$

Proof. It follows from y^{t+1} being a minimizing point on the boundary of the ball that $\hat{c}^T y^{t+1} \leq \hat{c}^T \tilde{y}$ where \tilde{y} is any other point on this boundary. In particular, this is true for \tilde{y} defined as the intersection of the ball’s boundary with the line segment joining the center of the ball e to y^* where y^* corresponds to an optimal extreme point x^* (see Figure 3-9). By assumption A1 on Page 102 and Exercise 3.48, $\hat{c}^T x^* = 0$ implies $\hat{c}^T y^* = 0$. Because y^* , \tilde{y} , and e are on a straight line (see Figure 3-9), we have

$$\begin{aligned} \tilde{y} &= \lambda e + (1 - \lambda)y^* && \text{where } \lambda = \|y^* - \tilde{y}\|/\|y^* - e\| \\ \hat{c}^T \tilde{y} &= \lambda \hat{c}^T e + (1 - \lambda)\hat{c}^T y^* = \lambda \hat{c}^T e \end{aligned}$$

where $\hat{c}^T y^* = 0$ and $\hat{c}^T e = \hat{z}_t$. Therefore

$$\hat{c}^T \tilde{y} = \frac{\|y^* - \tilde{y}\|}{\|y^* - e\|} (\hat{c}^T e) = \frac{\|y^* - e\| - \|\tilde{y} - e\|}{\|y^* - e\|} \hat{z}_t \quad (3.127)$$

$$= \left(1 - \frac{\|\tilde{y} - e\|}{\|y^* - e\|}\right) \hat{z}_t = \left(1 - \frac{\rho}{\|y^* - e\|}\right) \hat{z}_t \quad (3.128)$$

$$\leq \left(1 - \frac{\rho}{\|ne^k - e\|}\right) \hat{z}_t, \quad (3.129)$$

where $\|ne^k - e\|$ is the distance of a vertex ne^k of the simplex \mathcal{S} from its center e , see Exercise 3.44. Since $\|ne^k - e\| = \sqrt{n(n-1)}$ we have finally

$$\hat{c}^T y^{t+1} \leq \hat{c}^T \tilde{y} \leq \left(1 - \rho/\sqrt{n(n-1)}\right) \hat{z}_t < (1 - \rho/n) \hat{z}_t,$$

where $\hat{z}_t = z_t$ (see Exercise 3.47). This completes our proof. \blacksquare

Having fixed ρ at some value, say $\rho = 1/2$, and found y^{t+1} by (3.125), the next step is to determine x^{t+1} as the inverse map by (3.109):

$$x^{t+1} = n \frac{Dy^{t+1}}{e^T Dy^{t+1}} = Dy^{t+1} \left(\frac{n}{\sum_{i=1}^n x_i^t y_i^{t+1}} \right), \quad D = \text{Diag}(x^t) \quad (3.130)$$

and to repeat the iterative process with $x^{t+1} > 0$, $z_{t+1} = c^T x^{t+1}$.

At this point, we ask in what way the solution x^{t+1} is an improvement over x^t ? Just because $\hat{c}^T y^{t+1} / \hat{c}^T y^t < 1 - \rho/n$ does not imply $c^T x^{t+1} < c^T x^t$. To see why, multiply (3.130) by $c^T = \hat{c}^T D^{-1}$ and then divide by $c^T x^t$ on the left and $\hat{c}^T y^t = c^T x^t$ on the right obtaining

$$\frac{c^T x^{t+1}}{c^T x^t} = \frac{\hat{c}^T y^{t+1}}{\hat{c}^T y^t} \left(\frac{n}{\sum_{i=1}^n x_i^t y_i^{t+1}} \right). \quad (3.131)$$

It is possible for the factor $n / \sum_{i=1}^n x_i^t y_i^{t+1}$ to be so large that $c^T x^{t+1} > c^T x^t$, in which case $z_{t+1} > z_t$ and there would be no improvement.

While this possibility may happen on some iterations, Karmarkar proved that $c^T x^t \rightarrow c^T x^* = 0$ by finding a function that bounds z_t , namely,

$$\frac{c^T x^t}{c^T x^o} \leq \frac{e^{-\gamma t/n}}{g^o} \quad \text{for some fixed } \gamma > 0, \quad (3.132)$$

where $g^o = (x_1^o x_2^o \cdots x_n^o)^{1/n}$ is the geometric mean of $x_1^o, x_2^o, \dots, x_n^o$. To this end, he defines

$$\begin{aligned} u(x) &= n \log(c^T x) - \sum_{j=1}^n \log(x_j) \\ v(y) &= n \log(\hat{c}^T y) - \sum_{j=1}^n \log(y_j) \end{aligned} \quad (3.133)$$

as corresponding *potential functions* if x maps into y by the projective transformation (3.111) for iteration t . (Functions of type (3.133) are used in physics and chemistry to measure the potential energy of physical systems, hence the term *potential function*.) He uses $v(y^t) - v(y^{t+1})$ to measure the “improvement” of the solution (y^{t+1}, \hat{z}_{t+1}) over that of (y^t, \hat{z}_t) , and similarly he uses $u(x^t) - u(x^{t+1})$ to measure the corresponding “improvement” of (x^{t+1}, z_{t+1}) over that of (x^t, z_t) .

LEMMA 3.22 (Improvement in Transformed Space Equals Improvement in Original Space) *If y is the map of x , then an improvement measured by a*

decrease in the potential function $v(y)$ in the transformed space corresponds to an equal improvement measured by a decrease in the potential function $u(x)$ in the original space.

Proof. We will first show that the value of the potential function $u(x)$ differs from $v(y)$ by a constant that depends on x^t , i.e.,

$$v(y) - u(x) = \sum_{j=1}^n \log(x_j^t). \quad (3.134)$$

Indeed, from the definitions (3.133) of the potential functions $u(x)$ and $v(y)$ and from (3.111) the transformation \mathcal{T} of $x \rightarrow y$,

$$\begin{aligned} v(y) &= n \log(\hat{c}^T y) - \sum_{j=1}^n \log(y_j) \\ &= n \log\left(n \frac{\hat{c}^T D D^{-1} x}{e^T D^{-1} x}\right) - \sum_{j=1}^n \log\left(n \frac{x_j/x_j^t}{e^T D^{-1} x}\right) \\ &= n \log(\hat{c}^T x) - \sum_{j=1}^n \log(x_j) + \sum_{j=1}^n \log(x_j^t) \\ &= u(x) + \sum_{j=1}^n \log(x_j^t). \end{aligned}$$

If (y^{t+1}, v_{t+1}) is any solution to (3.115) with improvement $v_{t+1} - v_t = \psi < 0$, it follows that it maps back into a solution (x^{t+1}, u^{t+1}) that equally improves (x^t, u^t) , i.e., $u^{t+1} - u^t = \psi < 0$. ■

Iterative Cycle: Assuming the move from (y^t, v_t) to (y^{t+1}, v_{t+1}) is made by (3.125), (3.121) for some fixed ρ then the change of coordinates $(x^t, u^t) \rightarrow (y^t, u^t)$ and the change of coordinates back of the improvement (y^{t+1}, v_{t+1}) satisfy:

$$\begin{array}{ccc} (x^t, u^t) & \longrightarrow & (y^t = e, v_t = n \log \hat{c}^T e) \\ & & \downarrow \\ (x^{t+1}, u^{t+1}) & \longleftarrow & (y^{t+1}, v_{t+1}) \end{array}$$

where $u^{t+1} - u^t = v_{t+1} - v_t$.

▷ **Exercise 3.56** Prove $v_t = v(e) = n \log \hat{c}^T e = n \log \hat{z}_t = n \log z_t$.

THEOREM 3.23 (Bounds on Potential Function Decrease) *The corresponding iterates x^{t+1} , y^{t+1} and the corresponding potential decreases $u_{t+1} - u_t = v_{t+1} - v_t$ satisfy the following inequalities at each iteration*

$$x^{t+1} > 0, \quad y^{t+1} > 0, \quad (3.135)$$

$$u_{t+1} - u_t = v_{t+1} - v_t \leq -\gamma = -1 + \log(2) \approx -0.307, \quad (3.136)$$

where the iterate y^{t+1} is defined to be the point that minimizes $\hat{c}^T y$ subject to $e^T y = n$, $\hat{A}y = 0$, and $\|y - e\| = \rho$ for $\rho = 1/2$.

Proof. Since $y^t = e$ and $\hat{c}^T y^t = \hat{z}_t$ we get $v_t = n \log \hat{z}_t$. Therefore

$$\begin{aligned} v_{t+1} - v_t &= n \log(\hat{c}^T y^{t+1}) - \sum_{j=1}^n \log(y_j^{t+1}) - n \log \hat{z}_t \\ &< n \log((1 - \rho/n)\hat{z}_t) - \sum_{j=1}^n \log(y_j^{t+1}) - n \log \hat{z}_t \end{aligned} \quad (3.137)$$

$$\begin{aligned} &= n \log(1 - \rho/n) - \sum_{j=1}^n \log(y_j^{t+1}) \\ &= n \log(1 - \rho/n) - \sum_{j=1}^n \log(1 - \epsilon_j), \end{aligned} \quad (3.138)$$

where (3.137) follows from Lemma 3.21 and we have set $y_j^{t+1} = 1 - \epsilon_j$. Since y^{t+1} satisfies $e^T y^{t+1} = n$ and belongs to the boundary of the ball of radius ρ , ϵ_j must satisfy the following:

$$\sum_{j=1}^n \epsilon_j = 0, \quad \sum_{j=1}^n \epsilon_j^2 = \rho^2 < 1, \quad |\epsilon_j| \leq \rho. \quad (3.139)$$

From the definition of $\log(1 - \theta)$ for $|\theta| < 1$, i.e.,

$$\log(1 - \theta) = -\left(\theta + \frac{1}{2}\theta^2 + \frac{1}{3}\theta^3 + \dots\right) \text{ for } |\theta| < 1 \quad (3.140)$$

and (3.139) we get

$$\begin{aligned} v_{t+1} - v_t &< n \log(1 - \rho/n) + \sum_{j=1}^n \left(\epsilon_j + \frac{1}{2}\epsilon_j^2 + \frac{1}{3}\epsilon_j^3 + \dots\right) \\ &\leq n \log(1 - \rho/n) + \sum_{j=1}^n \left(\frac{1}{2}\epsilon_j^2 + \frac{1}{3}\epsilon_j^2 \rho + \frac{1}{4}\epsilon_j^2 \rho^2 + \dots\right) \\ &= n \log(1 - \rho/n) + \left(\frac{1}{2}\rho^2 + \frac{1}{3}\rho^3 + \frac{1}{4}\rho^4 + \dots\right) \\ &= n \log(1 - \rho/n) - \rho - \log(1 - \rho) \\ &< -2\rho - \log(1 - \rho). \end{aligned} \quad (3.141)$$

Finally, we see that the minimum for the upper bound (3.141) for $v_{t+1} - v_t$ is achieved if we choose the radius $\rho = \rho^*$ by setting the derivative of the right-hand side to zero:

$$-2 + \frac{1}{1 - \rho^*} = 0. \quad (3.142)$$

Clearing fractions, we get $\rho^* = 1/2$. Substituting $\rho^* = 1/2$ into (3.141), we obtain

$$u_{t+1} - u_t = v_{t+1} - v_t < -1 + \log(2) = -\gamma \approx -0.307. \quad (3.143)$$

This γ results in a somewhat sharper bound than that found by Karmarkar. ■

▷ **Exercise 3.57** Prove $n \log(1 - \rho/n) < -\rho$.

▷ **Exercise 3.58** Prove that a sharper bound can be obtained by setting the derivative of $n \log(1 - \rho/n) - \rho$ to zero and substituting the value of $\rho = \rho^*$ so obtained to compute γ .

THEOREM 3.24 (Bound on Objective Function Ratio) *Letting $g^o = (x_1^o x_2^o \cdots x_n^o)^{1/n}$ be the geometric mean of the coordinates $x_1^o, x_2^o, \dots, x_n^o$,*

$$\frac{c^T x^t}{c^T x^o} < \frac{e^{-\gamma t/n}}{g^o}, \quad (3.144)$$

where $\gamma = 1 - \log(2) \approx 0.307$.

Proof. Summing $u_{k+1} - u_k < -\gamma$ from $k = 0$ to $k = t-1$, we obtain $u_t - u_o < -\gamma t$. Hence we have from (3.133) the definition of u_t , that:

$$-\gamma t > u_t - u_o = n \log(c^T x^t) - \sum_{j=1}^n \log(x_j^t) - n \log(c^T x^o) + \sum_{j=1}^n \log(x_j^o).$$

Because $\log(\phi)$ is a concave function, we have

$$(1/n) \sum_{j=1}^n \log(x_j^t) \leq \log\left(\sum_{j=1}^n (x_j^t/n)\right) = 0;$$

thus

$$-\gamma t > n \log(c^T x^t) - n \log(c^T x^o) + \sum_{j=1}^n \log(x_j^o).$$

Dividing by n and rearranging terms this implies that

$$\log\left(\frac{c^T x^t}{c^T x^o}\right) < -\frac{\gamma t}{n} - \sum_{j=1}^n \log(x_j^o),$$

from which (3.144) follows. ■

COROLLARY 3.25 (Polynomial Time Bound) *The algorithm is guaranteed to stop in $t < n(q - \log(g^o))/\gamma$ iterations.*

Proof. Apply (3.114), the termination condition, assumption A3 on Page 102, and Theorem 3.24. ■

If some of the components x_j^o are very small then it is clear that the factor $1/g^o$ can be very large, implying poor convergence. One way to avoid this is as follows. Suppose that the original problem is $\min \bar{c}^T \bar{x}$, s.t. $\bar{A}\bar{x} = 0$, $e^T \bar{x} = n$, $\bar{x} \geq 0$ and we are given a starting a feasible solution $\bar{x} = \bar{x}^o > 0$. Instead of optimizing this problem, we first convert the problem by applying a projective transformation \mathcal{T} to the constraints of the original problem. This mapping of \bar{x} to x transforms the linear objective function $\bar{c}^T \bar{x}$ to a fractional objective function $n\bar{c}^T \bar{D}x / e^T \bar{D}x$. Suppose we “throw away” the denominator divided by n and solve instead the problem $\min c^T x$ s.t. $Ax = 0$, $e^T x = n$, $x \geq 0$ where $c = \bar{D}\bar{c}$, $A = \bar{A}\bar{D}$ with $\bar{D} = \text{Diag}(\bar{x}^o)$. Then we solve this converted problem with the interior-point method just described with starting solution $x^o = e > 0$. Let $x = x^*$ be optimal. Since the geometric mean of the components of x^o is now $g^o = 1$, (3.144) now reduces to

$$\frac{c^T x^t}{c^T x^o} \leq e^{-\gamma t/n}. \quad (3.145)$$

It would appear that we could obtain a much better upper bound for $u_{t+1} - u_t = v_{t+1} - v_t$ if we had a sharp upper bound for $-\sum_{j=1}^n \log(1 - \epsilon_j)$ in (3.138). Lemma 3.26 will provide such a bound but, as we will see later, it differs very little from $-2\rho - \log(1 - \rho)$.

LEMMA 3.26 (Sharp Upper Bound) Given $\sum_{j=1}^n \epsilon_j = 0$, $\sum_{j=1}^n \epsilon_j^2 = \rho^2$,

$$\begin{aligned} F &= -\sum_{j=1}^n \log(1 - \epsilon_j) \\ &\leq -\log\left(1 - \rho\sqrt{\frac{n-1}{n}}\right) - (n-1)\log\left(1 + \rho\sqrt{\frac{1}{n(n-1)}}\right). \end{aligned}$$

This upper bound is sharp and is attained by setting $\epsilon_n = \rho\sqrt{(n-1)/n}$ and $\epsilon_1 = \epsilon_2 = \dots = \epsilon_{n-1} = -\rho/\sqrt{n(n-1)}$.

Proof. To find $\max F$ subject to $\sum_{j=1}^n \epsilon_j = 0$, $\sum_{j=1}^n \epsilon_j^2 = \rho^2$ we form the Lagrangian

$$L = -\sum_{j=1}^n \log(1 - \epsilon_j) - \lambda \sum_{j=1}^n \epsilon_j - \frac{1}{2}\mu \sum_{j=1}^n \epsilon_j^2 \quad (3.146)$$

and set its partials $\partial L / \partial \epsilon_j = 0$, obtaining for some fixed choices of λ and μ :

$$\frac{1}{1 - \epsilon_j} - \lambda - \mu\epsilon_j = 0, \quad \text{for } j = 1, \dots, n. \quad (3.147)$$

Multiplying by $(1 - \epsilon_j)$, we see that each ϵ_j is equal to either the positive or the negative root of the quadratic

$$\mu\epsilon_j^2 + (\lambda - \mu)\epsilon_j + (1 - \lambda) = 0. \quad (3.148)$$

Thus ϵ_j are functions of λ and μ and the fixed values of λ and μ are adjusted so that $\sum_j \epsilon_j = 0$ and $\sum_j \epsilon_j^2 = \rho^2$. To this end we sum (3.148) for $j = 1, \dots, n$ and set $\sum_j \epsilon_j^2 = \rho^2$ and $\sum_j \epsilon_j = 0$ obtaining $1 - \lambda = -\rho^2 \mu / n$. Dividing (3.148) by μ :

$$\epsilon_j^2 + \left(\frac{\lambda - \mu}{\mu} \right) \epsilon_j - \frac{\rho^2}{n} = 0. \quad (3.149)$$

At this point we know that some integer k of the ϵ_j all have the same value $\epsilon_j = u > 0$ and that the remaining $l = n - k$ of the ϵ_j have the same value $\epsilon_j = -\bar{u} < 0$ and that the product of the roots $(-\bar{u})u = -\rho^2/n$. Therefore we wish to find $u > 0$, $\bar{u} > 0$ and integers $k > 0$, $l > 0$ to obtain

$$\max F = k[-\log(1 - u)] + l[-\log(1 + \bar{u})]$$

where

$$k + l = n, \quad (3.150)$$

$$ku - l\bar{u} = 0, \quad \text{since } \sum_{j=1}^n \epsilon_j = 0 \quad (3.151)$$

$$ku^2 + l\bar{u}^2 = \rho^2, \quad \text{since } \sum_{j=1}^n \epsilon_j^2 = \rho^2 \quad (3.152)$$

$$u\bar{u} = \rho^2/n. \quad (3.153)$$

Although it appears that we have four relations (3.150), (3.151), (3.152), and (3.153) with which to determine $(u, \bar{u}) > 0$ and integers $(k, l) > 0$, it turns out that (3.153) is redundant, which can be seen by solving (3.150), (3.151), and (3.152) for u and \bar{u} :

$$u = \frac{\rho}{\sqrt{n}} \sqrt{\frac{l}{k}}, \quad \bar{u} = \frac{\rho}{\sqrt{n}} \sqrt{\frac{k}{l}}, \quad u\bar{u} = \frac{\rho^2}{n}. \quad (3.154)$$

We are thus left with one degree of freedom with which to maximize F . The relation $ku - l\bar{u} = 0$ implies for some α that $k = \alpha\bar{u}$, $l = \alpha u$. Substituting into the relation $k + l = n$ we obtain $\alpha = n/(u + \bar{u})$ and therefore we wish to choose u and \bar{u} so that

$$\frac{F}{n} = \frac{-\bar{u} \log(1 - u) - u \log(1 + \bar{u})}{u + \bar{u}}, \quad u\bar{u} = \frac{\rho^2}{n}, \quad (3.155)$$

where the range of u and \bar{u} are restricted by the range of integers k and l in (3.154), i.e.,

$$\frac{\rho}{\sqrt{n(n-1)}} \leq u = \frac{\rho}{\sqrt{n}} \sqrt{\frac{l}{k}} \leq \rho \sqrt{\frac{n}{n-1}}. \quad (3.156)$$

We will prove that F in (3.155) monotonically increases with u and therefore at $\max F$ we have $u = \rho\sqrt{(n-1)/n}$, $\bar{u} = \rho/\sqrt{n(n-1)}$, which corresponds to $k = 1$

and $l = n - 1$. To get a better separation of variables, we divide (3.155) on the left by ρ^2/n and the right side by $u\bar{u} = \rho^2/n$ obtaining

$$\frac{F}{\rho^2} = \left[-\frac{\log(1-u)}{u} - \frac{\log(1+\bar{u})}{\bar{u}} \right] / (u + \bar{u}). \quad (3.157)$$

Instead of trying to show that $dF/du > 0$, we get a more symmetric treatment of the terms involving u and \bar{u} if we let $w = \log u$ and show that $dF/dw > 0 \Leftrightarrow dF/du > 0$. To see this, note that $du/dw = u$, $d\bar{u}/dw = -\bar{u}$, and $dF/du = (dF/dw)(dw/du) = (dF/dw)(1/u) > 0$ when $dF/dw > 0$ since $u > 0$. Differentiating (3.157) with respect to w , setting $du/dw = u$, $d\bar{u}/dw = -\bar{u}$, and then multiplying both sides by the resulting denominator $(u + \bar{u})^2$ we obtain

$$\begin{aligned} \frac{dF}{dw} \left(\frac{(u + \bar{u})^2}{\rho^2} \right) &= \left[\frac{1}{1-u} + \frac{1}{1+\bar{u}} \right] (u + \bar{u}) \\ &\quad + \left[\frac{\log(1-u)}{u} - \frac{\log(1+\bar{u})}{\bar{u}} \right] (u + \bar{u}) \\ &\quad + \left[\frac{\log(1-u)}{u} + \frac{\log(1+\bar{u})}{\bar{u}} \right] (u - \bar{u}) \\ &= \left[\frac{1}{1-u} + \frac{1}{1+\bar{u}} \right] [(1+\bar{u}) - (1-u)] \\ &\quad + 2 \log(1-u) - 2 \log(1+\bar{u}) \\ &= 2 \log \left(\frac{1-u}{1+\bar{u}} \right) + \frac{1+\bar{u}}{1-u} - \frac{1-u}{1+\bar{u}} \\ &= 2 \log \theta + \frac{1}{\theta} - \theta = g(\theta), \end{aligned}$$

where $\theta = (1-u)/(1+\bar{u})$. Note for any u such that $\rho/\sqrt{n(n-1)} \leq u \leq \rho\sqrt{(n-1)/n}$ that the corresponding θ satisfies $0 < \theta < 1$. For any θ , $0 < \theta < 1$ the function $g(\theta) = 2 \log \theta - \theta + 1/\theta$ is positive because first of all its derivative $2/\theta - 1 - 1/\theta^2 = -(1/\theta - 1)^2 < 0$, implying $g(\theta)$ is decreasing in the range $(0 < \theta < 1)$, and second, at $\theta = 1$, $g(\theta) = 0$. Finally $g(\theta) > 0$ implies $dF/dw = (\rho^2/(u + \bar{u})^2)g(\theta) > 0$ for any u in its admissible range.

Thus F achieves its maximum at the maximum of $u = (\rho/\sqrt{n})\sqrt{l/k}$, which is when $k = 1$ and $l = n - 1$. This completes the proof of the Lemma. \blacksquare

To improve the bound obtained in Theorem 3.23, we have just shown

$$\begin{aligned} v_{t+1} - v_t &< n \log(1 - \rho/n) - \log \left(1 - \rho \sqrt{(n-1)/n} \right) \\ &\quad - (n-1) \log \left(1 + \rho / \sqrt{n(n-1)} \right). \end{aligned} \quad (3.158)$$

Again we are free to choose any radius $\rho < 1$ so we choose $\rho = \rho^*$, which yields the smallest upper bound for $v_{t+1} - v_t$. We find $\rho = \rho^*$ by setting to 0 the derivatives

of the bound with respect to ρ obtaining:

$$\frac{-1}{1 - \rho^*/n} + \frac{\sqrt{(n-1)/n}}{1 - \rho^* \sqrt{(n-1)/n}} - \frac{\sqrt{(n-1)/n}}{1 + \rho^* / \sqrt{n(n-1)}} = 0.$$

Clearing of fractions and solving for ρ^* we obtain

$$\rho^* = \frac{1}{1 + (n-2)/\sqrt{n(n-1)}}. \quad (3.159)$$

In the limit

$$\rho^* \approx 0.5 \quad \text{as } n \rightarrow \infty, \quad (3.160)$$

which yields the same approximation as that obtained from (3.142).

▷ **Exercise 3.59** Show (3.160).

▷ **Exercise 3.60** Compute ρ^* using (3.160) with $n = 1000$. Substitute this value of ρ^* in (3.158) and compute the upper bound. How does this compare with -0.307 , the bound in (3.136)?

3.3.3 THE ALGORITHM SUMMARIZED

According to the theory developed in the previous section, to decrease the potential $u(x) = \log(c^T x) - \sum_{j=1}^n x_j$ by at least $\gamma = 0.307$, we can do so by moving in the transformed space from e in the direction \hat{p} , where \hat{p} is the projection of \hat{c} onto the null space of the equality constraints (3.116). The steps of the algorithm applied to (3.113) are then as follows.

Algorithm 3.3 (Karmarkar's Algorithm) Given a linear program in the standard form (3.113) that satisfies assumptions A1–A3 (see Page 102) and has a feasible interior point $x^o = e$.

1. Initiate with $t = 0$ and the feasible interior point $x^o = e$.
2. If $c^T x^o = 0$ stop with $x^* = x^o$ as optimal.
3. *Check for Convergence.* Given the current point x^t , test for termination:

$$\frac{c^T x^t}{c^T x^o} \leq 2^{-q}.$$

If the test is satisfied stop with $x^* = x^t$ declared as optimum.

4. *Compute the Search Direction.*

(a) Let

$$D = \text{Diag}(x^t).$$

(b) Let

$$F = \begin{pmatrix} AD \\ e^T \end{pmatrix}$$

(c) Compute

$$\hat{p}^t = -[I - F^T(FF^T)^{-1}F]Dc$$

as the direction of decrease for $c^T D y$ from the center of the simplex S in the transformed space.

(d) Set $p^t = \hat{p}^t / \|\hat{p}^t\|$.

5. Compute the New Point y^{t+1} in the Transformed Space. Set

$$y^{t+1} = y^t + \alpha r p^t,$$

where $y^t = e$, $r = \sqrt{n/(n-1)}$ is the radius of the largest inscribed sphere in the simplex in the transformed space, and $\alpha \in (0, 1)$ is a fixed constant, $\alpha r = \rho$ may be fixed at $\rho = 0.5$ according to the proof of Theorem 3.23. In practice $\alpha r = \rho$ can be chosen even larger, for example, $\alpha r = 0.9$.

6. Compute the New Point x^{t+1} in the Original Space: Finally, transform the coordinates back to the original space using (3.111), i.e.,

$$x^{t+1} = n \frac{D y^{t+1}}{e^T D y^{t+1}}.$$

7. Set $t \leftarrow t + 1$ and return to Step 3.

Comment: Since we want to maximize the decrease of the potential function it is advantageous to determine the $\alpha r = \rho$ that minimizes the potential function along the direction p^t in Step 5. An efficient technique for doing this is based on developing a line search using a cubic fit.

3.3.4 CONVERTING A STANDARD LP TO A STARTING FORM FOR THE ALGORITHM

One way to transform the general linear program to (3.113) and obtain a starting feasible solution is as follows. Consider the linear program in standard form

$$\begin{aligned} \text{Minimize} \quad & \tilde{c}^T y = z \\ \text{subject to} \quad & \tilde{A} y = b, \quad \tilde{A}: (m-1) \times (n-3), \\ & y \geq 0. \end{aligned} \tag{3.161}$$

We do not know whether a feasible solution exists, and if it exists, whether there exists an optimal solution that is bounded.

For the purpose of transforming (3.161) to a starting form, assume that a feasible solution exists and an optimal feasible solution exists that satisfies

$$e^T y < \sigma(n-1), \tag{3.162}$$

for some prespecified σ , a large number. (If it turns out that this assumption is incorrect we will discover this in the course of solving the problem; see Section 3.3.5.)

First we rescale y by defining

$$\bar{x} = \left(\frac{1}{\sigma}\right) y, \quad (3.163)$$

where $\bar{x} \in \Re^{n-3}$. This gives:

$$\begin{aligned} \text{Minimize} \quad & \sigma \tilde{c}^T \bar{x} = z \\ \text{subject to} \quad & \tilde{A} \bar{x} = (1/\sigma)b \\ & \bar{x} \geq 0 \end{aligned} \quad (3.164)$$

which can in turn be written in the form

$$\begin{aligned} \text{Minimize} \quad & \tilde{c}^T \bar{x} & & = z \\ \text{subject to} \quad & \tilde{A} \bar{x} - (1/\sigma)bx_{n-2} & & = 0 \\ & & x_{n-2} & = 1 \\ & e^T \bar{x} + x_{n-2} + x_{n-1} & = & n \\ & (\bar{x}, x_{n-2}, x_{n-1}) & \geq & 0. \end{aligned} \quad (3.165)$$

Note that $x_{n-2} = 1$ and x_{n-1} is the value of the *slack* in the equation $e^T \bar{x} < n - 1$. Furthermore, the objective function has been rescaled by dividing by σ .

In order to have a starting feasible solution we introduce an artificial variable x_n and assign to it a large cost coefficient M in the objective and assign the remaining coefficients as shown in (3.166).

$$\begin{aligned} \text{Minimize} \quad & \tilde{c}^T \bar{x} & + & Mx_n = z \\ \text{subject to} \quad & \tilde{A} \bar{x} - (1/\sigma)bx_{n-2} & + & ((1/\sigma)b - \tilde{A}e)x_n = 0 \\ & & x_{n-2} & = 1 \\ & e^T \bar{x} + x_{n-2} + x_{n-1} & + & x_n = n \\ & (\bar{x}, x_{n-2}, x_{n-1}, x_n) & \geq & 0. \end{aligned} \quad (3.166)$$

Note that $x = (\bar{x}, x_{n-2}, x_{n-1}, x_n) = e = (1, 1, \dots, 1)^T$ is a starting feasible solution to (3.166). As a final step, subtracting $1/n$ times the last equation from the second-from-last equation puts the problem in the required form except for the optimal objective being 0.

3.3.5 COMPUTATIONAL COMMENTS

Comment 1: If at the optimal solution of (3.166) it turns out that x_n is zero (i.e., very small in practice), the original linear program (3.161) is feasible. If it turns out that, in addition to x_n being zero, $x_{n-1} = 0$ at the optimal solution to (3.166) then this implies that (3.162) is satisfied as an equality implying that σ was not chosen sufficiently large. If σ was in fact chosen extremely large it could be interpreted that the linear program is unbounded for all practical purposes.

Comment 2: If at the optimal solution of (3.166) it turns out that x_n is not zero (i.e., not very small in practice), then either M was not chosen large enough or our assumption that (3.165) is feasible is incorrect. At this point we can decide either to quit or to try again by increasing σ , or to try again by increasing M , or to try again by increasing both M and σ .

▷ **Exercise 3.61** Construct an example that shows that if σ is not chosen large enough, problem (3.165) may be infeasible even if the original linear program (3.161) is feasible.

Comment 3: The requirement that the objective function be zero is equivalent to assuming that an optimal value $z = z^*$ is known because it is easy to convert an objective whose minimum is z^* to one whose minimum is zero. All we have to do is replace $z = c^T x$ by:

$$\hat{z} = (c - (z^*/n)e)^T x$$

as the new objective.

However, since we do not know the minimum z^* in general, a method needs to be devised to handle this situation. One approach is to use an estimate of z^* of the original problem, namely, \bar{z} . The objective after the transformation is

$$\hat{z} = (c - (\bar{z}/n)e)^T x.$$

One way to update this estimate as the iterations proceed is to identify a dual solution at each iteration and let \bar{z} be the objective value of the dual problem, which we know is a lower bound for the primal objective value. It turns out that it is very easy to find a feasible dual solution when $Ax = 0$ and $e^T x = n$. Suppose that the primal problem is of the form:

$$\begin{aligned} \text{Minimize} \quad & c^T y = z \\ \text{subject to} \quad & Ay = 0, \\ & e^T y = n, \\ & y \geq 0. \end{aligned} \tag{3.167}$$

The dual of (3.167) is to

$$\begin{aligned} \text{Maximize} \quad & nv \\ \text{subject to} \quad & A^T \pi + ve \leq c \end{aligned} \tag{3.168}$$

One way to find a feasible dual solution (π, v) is to choose *any* π and set $v = \min_j \{(c_j - \pi^T A_{\bullet j})\}$. It is equally straightforward to examine the dual in the transformed space, i.e., for (3.116).

However for some variants of Karmarkar's method to converge, one needs to find a π such that (π, v) s converge to an optimal dual feasible solution. One approach to do this is shown next without proof. The dual problems

considered here are generated in the transformed space. Start by obtaining π^o as the least-squares solution to:

$$DA^T\pi^o = Dc, \quad (3.169)$$

where $D = \text{Diag}(x^o)$. Set $v_o = \min_j \{(c_j - A_{\bullet j}^T \pi^o)\}$. Then on subsequent iterations, say at the start of iteration $k + 1$, update the dual variables. First solve the least squares problem obtained from the dual to the primal problem whose objective has been replaced by $(c - v_k e)^T x$, i.e.,

$$DA^T\pi^{k+1} = D(c - v_k e), \quad (3.170)$$

where $D = \text{Diag}(x^k)$ and $e = (1, 1, \dots, 1)^T$. Set $\bar{v} = \min_j \{(c_j - A_{\bullet j}^T \pi^k)\}$. If $\bar{v} \leq v_k$ then we have not improved our solution and we set $v_{k+1} = v_k$. Otherwise, if $\bar{v} > v_k$, set $v_{k+1} = \bar{v}$ and revise the estimate of π^{k+1} by solving:

$$DA^T\pi^{k+1} = D(c - v_{k+1} e), \quad (3.171)$$

where $D = \text{Diag}(x^k)$ and $e = (1, 1, \dots, 1)^T$.

Comment 4: In most practical cases it appears that the number of iterations performed by interior-point methods is remarkably small and grows slowly with problem size. Note, however, that each iteration requires solving a large linear least-squares problem (see, for example, Step 3(b) of Algorithm 3.3). It is clear that the key to computational efficiency is being able to solve such least-squares problem quickly. Certain classes of practical models give rise to least-squares problems that can be solved very quickly because the linear programs have specially structured coefficient matrices.

Comment 5: A practical numerical difficulty is that the least-squares problems become more and more ill-conditioned as the optimal solution is approached. This is especially true in the degenerate case, which almost always occurs in practice.

Comment 6: Finally, note that interior-point methods almost never find a vertex solution when the linear program does not have a unique optimum solution. This can have serious implications for sensitivity analysis and when the dual solution is important.

3.3.6 COMPLEXITY OF VON NEUMANN VERSUS KARMARKAR ALGORITHMS

To attain a precision of ϵ , Karmarkar's Algorithm has an upper bound of

$$n(-\log_e \epsilon)/\gamma$$

iterations where $\gamma = 1 - \log_e 2 \approx 0.3$. The work per iteration for his algorithm is considerably higher than a von Neumann iteration. To load the dice in favor of the Karmarkar algorithm, we will assume the work per iteration is the same. The comparison is therefore for the same precision ϵ :

<i>Algorithm</i>	<i>Upper Bound on Operations</i>
Karmarkar	$\left(\frac{1}{0.3}\right) (-\log_e \epsilon)n(mn + 2m + n + 9)$
Von Neumann	$\left(\frac{1}{\epsilon^2}\right) (mn + 2m + n + 9)$

For a precision ϵ , say $\epsilon = 10^{-10}$, the von Neumann method has an upper bound on the number of arithmetic operations which is lower than that for Karmarkar when the number of variables

$$h > .3(-\log_e \epsilon)/\epsilon^2 = 1.3 \times 10^{18} \text{ for } \epsilon = 10^{-10}.$$

The polynomial complexity of Karmarkar as simplified here is 3 (actually 3.5) while that of von Neumann is 2, but the latter has a 10^{18} constant factor to obtain a precision of 10^{-10} .

3.4 NOTES & SELECTED BIBLIOGRAPHY

Interior-point methods are not recent; they have been around for a very long time. For example, von Neumann [1947] (see Dantzig [1992a]), Hoffman, Mannos, Sokolowsky, & Wiegmann [1953], Tompkins [1955, 1957], Frisch [1957], Dikin [1967]. (Fiacco & McCormick [1968] further developed Frisch's Barrier Method approach to nonlinear programming.) None of these earlier methods, up to and including Khachian's [1979] ellipsoidal polynomial-time method, turned out to be competitive in speed to the Simplex Method on practical problems.

Von Neumann, in a private communication with Dantzig in 1948, proposed the first interior algorithm for finding a feasible solution to a linear program with a convexity constraint. Dantzig [1992a] proved it has the remarkable property that *independent of the number of rows m and columns n* , it generates in less than $1/\epsilon^2$ iterations a feasible solution with a precision ϵ (where ϵ^2 is the sum of the squares of errors of the fit of the left-hand side of the equations to the right-hand side) when the general problem is recast in the form:

$$x \geq 0, \quad \sum_1^n x_j = 1, \quad \sum_1^n P_j x_j = 0, \quad \|P_j\|_2 = 1 \quad \text{for all } j.$$

For a comparison of the complexity of von Neumann's versus Karmarkar's algorithms see Section 3.3.6.

The modification to improve the rate of convergence was developed by Dantzig in 1997 and not published. Another approach that attempts to improve the rate of convergence is one that exploits a "bracketing" idea. Analogous to gunners firing trial shots to bracket a target in order to adjust direction and distance, we demonstrate that it is sometimes faster not to apply an algorithm directly, but to approximately solve several perturbations of the problem and then combine these rough approximations to get an exact solution. This is described in Dantzig [1992b]. Such an approach may also be a way to speed up other infinitely converging methods.

Dikin's [1967] method has the distinction of having been rediscovered by many; for example, the primal affine method is the same as Dikin's method. Later, Dikin [1974] proved

convergence of his method under primal nondegeneracy. Proofs of convergence of Dikin's iterates discussed in Section 3.2 can also be found in Adler, Resende, Veiga, & Karmakar [1989], Barnes [1986], Dantzig [1988a], Dantzig & Ye [1990], Monma & Morton [1987], Vanderbei, Meketon, & Freedman [1986], and, under somewhat weaker assumptions, in Vanderbei & Lagarias [1988].

In 1979, Khachian presented an algorithm, based on a nonlinear geometry of shrinking ellipsoids, with a worst-case polynomial-time bound of $O(n^6 \mathcal{L}^2)$, (where \mathcal{L} is the number of bits required to represent the input data on a computer). Given an open set of inequalities of the form $Ax < b$, where A is $m \times n$ with $m \geq 2$, $n \geq 2$, Khachian's algorithm either finds a feasible point if the system is nonempty or demonstrates that no feasible point exists. Assuming that the inequalities have a feasible solution, the method starts by defining a ball that is assumed to have a radius large enough to contain a sufficiently large volume of the feasible space defined by the inequalities $Ax < b$. If the center of the ball is within the open set of inequalities, a feasible solution has been found and the algorithm terminates. If a feasible solution is not obtained, the method proceeds to the next iteration by constructing an ellipsoid of smaller volume which contains the feasible space of the inequalities contained in the previously drawn ball. If the center of the ellipsoid is in the feasible space of $Ax < b$ we have found a feasible solution; otherwise the method proceeds to the next iteration by constructing another ellipsoid of smaller volume, and so on.

The theory developed by Khachian states that if a feasible solution exists, then the center of some ellipsoid will lie in the feasible space within a number of iterations bounded by some polynomial expression in the data. Although Khachian's ellipsoid method has nice theoretical properties, unfortunately, it performs poorly in practice. First, the number of iterations tends to be very large, and second, the computation associated with each iteration is much more than needed with the Simplex Method. Khachian's work specialized to linear programming is based on earlier work done by Shor [1971a, 1971b, 1972a, 1972b, 1975, 1977a, 1977b] for the more general case of convex programming. Other work that was influenced by Shor and preceded Khachian was due to Judin & Nemirovskii [1976a, 1976b, 1976c]. Predating all this was an article by Levin [1965] for convex programming. For detailed proofs of Khachian's polynomial-time complexity results see, for example, Papadimitriou & Steiglitz [1982] and Grötschel, Lovász, & Schrijver [1988].

In 1984, Karmarkar presented his interior-point ellipsoid method with a worst-case polynomial-time bound of $O(n^{3.5} \mathcal{L}^2)$, where \mathcal{L} , as defined here, is the number of bits required to represent the input data on a computer. Claims by Karmarkar that his method is much faster (in some cases 50 times faster) than the Simplex Method stimulated improvements in the simplex-based algorithms and the development of alternative interior-point methods. More than a thousand papers on interior methods have been published during the period 1984–2003. See Kranich [1991] for a bibliography, and M. Wright [1992] for a review of interior-point methods. Also see Lustig, Marsten, & Shanno [1994] for a review of the computational aspects of interior-point methods. Until 2003 no method has been devised to our knowledge that is superior for all problems encountered in practice.

The upper bound of $\gamma = -0.307$ developed in Section 3.3 is a sharper bound than that found by Karmarkar. However, an even sharper bound is claimed by Anstreicher [1989] and independently by McDiarmid [1990], namely:

$$u_{t+1} - u_t = v_{t+1} - v_t < -\log(2) \approx -0.69 = -\bar{\gamma}. \quad (3.172)$$

Anstreicher [1989] and McDiarmid [1990] further showed that the bound on the potential decrease approaches 0.7215 in the limit as the size of the problem tends towards ∞ when

a step-size close to 1 is taken. Karmarkar's algorithm and its variants, when applied to practical problems, take much fewer iterations than the theoretical worst-case polynomial-time bound.

Powell [1991] showed a potentially serious computational difficulty by constructing an example with n variables where the actual number of iterations is $O(n)$. Gonzaga [1991] devised a potential-reduction method that avoids the necessity of using projective transformations; variants of this method allow large steps to be taken. For some other theoretical results, see Freund [1988a, 1988b, 1991a].

Tomlin [1987] proposed an approach in 1985 to convert a linear program in standard form to a standard form for Karmarkar's algorithm, see (3.113). His approach, as described in this chapter, also provides a way to generate an initial starting feasible strictly interior solution. Gay [1987] and de Ghellinck & Vial [1986] also show how to apply Karmarkar's method to a linear program in standard form. Another approach to determine an initial feasible interior point is described in Section 4.4.2.

The method of estimating and updating the estimate \bar{z} of the optimal z^* described through equations (3.169), (3.170), and (3.171) is due to Todd & Burrell [1986]; their paper describes a variant of Karmarkar's algorithm. They prove that the dual feasible solutions converge and thus the estimate \bar{z} converges to z^* . Todd and Burrell also show that instead of using $\alpha = 0.9$, one could perform a line search on the potential function

$$v = n \log(\bar{c}^T e + \alpha p) - \sum_{j=1}^n \log(1 + \alpha p_j),$$

i.e., along the direction $p = \hat{p}/\|\hat{p}\|$ computed in Step 5 of Algorithm 3.3.

Bahn, Goffin, Vial, & Merle [1994], Goffin & Vial [1990], and Mitchell & Todd [1992] have used a projective interior-point algorithm for cutting plane algorithms. Yamashita [1986] implemented a projective algorithm applied to the dual. Additional computational results are reported by Anstreicher & Watteyne [1993], Fraley & Vial [1989], and Todd [1990a]. Anstreicher & Watteyne [1993] indicate that the reason projective methods do not perform as well as affine-scaling and primal-dual methods (see Chapter 4) is because of the necessity of generating upper and lower bounds on the optimal value. The poor performance is particularly noticeable when the dual has no feasible interior. Todd [1994a] draws the same conclusion.

3.5 PROBLEMS

- 3.1 Construct examples in $m = 2$ and $m = 3$ dimensions to show that the convex hull contains the center in its interior but there exists no nondegenerate basic feasible solution.
- 3.2 Solve, by hand, the following problem by the variant of Karmarkar's method described in this chapter:

$$\begin{aligned} &\text{Minimize} && 4x_1 - 2x_2 + 3x_3 = z \\ &\text{subject to} && x_1 + x_2 + x_3 = 3 \\ &&& 2x_1 - x_2 - x_3 = 0 \\ &&& \text{and } x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{aligned} \tag{3.173}$$

- 3.3 Solve (3.173) by hand by Dikin's Method as described in this chapter.
- 3.4 *Ph.D. Comprehensive Exam, Fall 1985, at Stanford.* This question is based on issues raised in Karmarkar's "nonlinear algorithm" for linear programming. Let $N(x) = \frac{x}{e^T x}$, $e = (1, 1, \dots, 1)^T$ be the operator that

$$\text{maps } \{x \in \mathbb{R}^n \mid e^T x > 0\} \text{ to } \{x \in \mathbb{R}^n \mid e^T x = 1\}.$$

- (a) Is N a linear or affine function? Justify your answer.
- (b) Let Λ be a convex subset of \mathbb{R}^1 containing at least two points. Let

$$L = \{b + \lambda d \mid \lambda \in \Lambda\}$$

be a line segment in the domain of N . What is $N(L)$ like? Justify your answer.

- (c) Does N carry convex sets to convex sets? Justify your answer.

- 3.5 *Based on Ph.D. Comprehensive Exam, Fall 1985, at Stanford.* This question is based on issues raised in Karmarkar's "nonlinear algorithm" for linear programming. Let A and B be $m \times n$ and $k \times n$ matrices of full row rank. Assume $AB^T = 0$. Define the sets

$$\begin{aligned} \mathcal{A} &= \{x \mid Ax = 0\}, \\ \mathcal{B} &= \{x \mid Bx = 0\}, \\ \mathcal{C} &= \{x \mid Ax = 0, Bx = 0\}. \end{aligned}$$

Given $y \in \mathbb{R}^n$ let $u \in \mathcal{A}$, $v \in \mathcal{B}$, and $w \in \mathcal{C}$, be the three points closest to y in the least-squares sense. In addition, let U , V , and W be the three functions that send y to u , v , and w respectively.

- (a) Does $A^T A$ have an inverse?
- (b) Does AA^T have an inverse?
- (c) Is U a linear or affine function of y ?
- (d) Derive an explicit formula for U .
- (e) Does $UU = U$?
- (f) Does $VU = W$?
- (g) Does $UV = VU$?
- 3.6 Consider a "generalized" ellipsoid with coordinates (y_1, y_2, \dots, y_n) defined by:

$$\sum_{j=1}^n |y_j - 1|^k = \rho^k. \quad (3.174)$$

- (a) Inscribe the largest such generalized ellipsoid into the simplex

$$(S) \quad \sum_{j=1}^n y_j = n, \quad y_j \geq 0 \text{ for } j = 1, \dots, n, \quad (3.175)$$

and find the point on the generalized ellipsoid closest to a vertex.

- (b) Prove that this point is invariant whatever power of k is used.
- (c) Does the generalized ellipsoid for power k include the generalized ellipsoid for power $k - 1$?

INTERIOR-POINT METHODS

In this chapter, we concentrate on path-following methods. We describe the primal logarithm barrier method and the primal-dual algorithm for solving linear programs. These methods are based on Newton's method, a logarithm barrier method, and the methods used to solve least-squares problems. Hence, we start by briefly describing Newton's method, the barrier function method, and the least-squares problem.

4.1 NEWTON'S METHOD

In this section our discussion will concern the minimization of a specified scalar-valued function $f(x)$, usually called an *objective function*. The following notation will be used throughout. Let $g(x)$, a column vector, denote the gradient vector of $f(x)$, that is,

$$g(x) = \nabla f(x) = \begin{pmatrix} \partial f(x)/\partial x_1 \\ \partial f(x)/\partial x_2 \\ \vdots \\ \partial f(x)/\partial x_n \end{pmatrix}.$$

If $f(x)$ has continuous second derivatives (i.e., it is twice-continuously differentiable), the symmetric matrix $G(x)$ of second partial derivatives will denote the *Hessian* matrix of $f(x)$. That is, $G(x)$ is the symmetric matrix of second partial derivatives of f , whose ij th element is given by

$$G_{ij}(x) = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}.$$

Newton's method appears in a variety of forms depending on the applications. The model algorithm that follows is a generalized outline for minimizing an unconstrained nonlinear function. Certain details are omitted such as tolerances for convergence, method for defining a descent search direction, and description of an efficient method for determining the steplength. After the model algorithm is described, we explain how to obtain a Newton search direction.

Algorithm 4.1 (Outline of an Algorithm to Solve $\min_{x \in \mathbb{R}^n} f(x)$) In order to minimize a nonlinear function $f(x)$, the algorithm starts with some x^0 and produces a series of iterates $x^1, x^2, \dots, x^k, \dots$.

1. *Test for When to Stop.* If, on iterate k , the conditions for stopping are satisfied, the algorithm terminates with x^k declared a local minimizer of $f(x)$. The tests are divided into two categories:
 - (a) *Standard Test.* Three tests usually must be passed simultaneously to decide when to stop: (i) the norm of the gradient of $f(x)$ at x^k , denoted by $g^k = g(x^k)$, is sufficiently small; (ii) the function values $f(x^{k-1}), f(x^k)$ are getting sufficiently close together; and (iii) the iterates x^{k-1}, x^k are getting sufficiently close together.
 - (b) *Pathological Case Test.* To handle the case where x^0 is the minimizer or x^k happens to be very close to a local minimizer but x^{k-1} is not, replace the preceding three tests by a test to terminate if the norm of the gradient is less than the bound on the absolute accuracy in the computation of the function value at any point.
2. *Compute a Search Direction.* Compute a nonzero vector p^k , satisfying $(p^k)^T g^k < 0$. Along such a direction the function value is decreasing in the neighborhood of x_k , and therefore $f(x^k + \alpha p^k) < f(x^k)$ for some $\alpha > 0$. Such a direction is called a *descent direction*.
3. *Compute a Steplength.* Compute a positive scalar $\alpha = \alpha_k$, the *steplength*, that is a minimizer of the one-dimensional problem $\min_{\alpha} f(x^k + \alpha p^k)$. This usually requires too much computational effort; instead α_k is chosen to satisfy somewhat looser criteria. An easy to implement, but not very efficient, way to estimate α_k is to start with $\alpha = 1$ and then keep dividing by 2 until the gradient of $f(x^k + \alpha p^k)$ with respect to α is sufficiently reduced from that at $\alpha = 0$ and $f(x^k + \alpha_k p^k) < f(x^k)$.
4. *Update the Estimate of the Minimum.* Set $x^{k+1} \leftarrow x^k + \alpha_k p^k$; $k \leftarrow k + 1$; and go back to Step 1.

CLASSICAL STEEPEST DESCENT

A classical search direction is the *steepest descent* direction $p^k = -g^k$, where g^k is the gradient of $f(x)$ at x^k . Methods that use it have the very nice theoretical property of guaranteed convergence from any starting point x^0 to a local minimizer if one exists. Unfortunately, in practice, these methods have an extremely poor rate of convergence and hence are not recommended.

NEWTON DIRECTION

A much better way, than the steepest descent direction, is a *Newton direction*, which is based on a quadratic approximation of the objective function at the current point x^k . We use the Taylor series expansion of the function to three terms to obtain a quadratic model of the function around x^k , i.e.,

$$f(x^k + p) \approx f(x^k) + p^T g(x^k) + \frac{1}{2} p^T G(x^k) p \quad (4.1)$$

which has an error term of $O(\|p\|^3)$. The matrix $G^k = G(x^k)$ is symmetric and is called the *Hessian matrix*. The minimum of the right hand side of (4.1) will be achieved if p^k is the solution of the problem

$$\underset{p}{\text{Minimize}} \quad Q(p) = f_k + p^T g^k + \frac{1}{2} p^T G^k p, \quad (4.2)$$

where $f_k = f(x^k)$ is a fixed scalar, $g^k = g(x^k)$ is a fixed vector and $G^k = G(x^k)$ is a fixed symmetric matrix. A stationary point of $Q(p)$ is given by the solution of $\nabla Q(p) = 0$, i.e., the solution of the linear system of equations:

$$G^k p^k = -g^k. \quad (4.3)$$

An algorithm for minimization that uses (4.3) to define a search direction p^k is classified as a *Newton's method* algorithm.

HESSIAN IS POSITIVE-DEFINITE

If G^k is positive-definite, so is $(G^k)^{-1}$, and hence the search direction p^k obtained using the system of equations (4.3) is a *descent direction*, because $g^k \neq 0$ and

$$(g^k)^T p^k = -(g^k)^T (G^k)^{-1} g^k < 0.$$

If G^k is positive-definite, $x^{k+1} = x^k + p^k$ solves (4.3), as the exact minimizer of the right-hand side of (4.1).

Even if G^k is positive-definite, the quadratic model may be a very poor approximation to the objective function. In particular, it is a poor approximation if $f(x^k + p^k)$ exceeds $f(x^k)$, which violates the descent condition. Thus, a steplength procedure must be included according to some acceptable criteria to construct a convergent algorithm. When Newton's Method is used with a steplength algorithm it is sometimes termed a *damped Newton's method* because the "natural" steplength of unity is not taken. Finally, note that the steepest descent direction is the vector p that solves the minimization problem

$$\min_p \frac{(g^k)^T p}{\|p\|_2}. \quad (4.4)$$

It turns out that the Newton direction defined by (4.3) is a “steepest descent” direction when the norm is defined with respect to G^k , the positive-definite symmetric Hessian matrix, i.e., p solves

$$\min_p \frac{(g^k)^T p}{\|p\|_{G^k}}. \quad (4.5)$$

▷ **Exercise 4.1** Let C be any symmetric positive-definite matrix. Then $\|x\|_c^2 = x^T C x$ is a norm.

1. The Cauchy-Schwartz inequality for any two n -vectors x, y is $|x^T y|^2 \leq (x^T x)(y^T y)$. Use this to derive the inequality

$$(g^T p)^2 \leq (g^T C^{-1} g)(p^T C p).$$

2. Use (1) to show that the solution of the problem

$$\text{Minimize}_{p \in \mathbb{R}^n} \frac{g^T p}{\|p\|_c}$$

where $g^T p < 0$, is given by $p = -C^{-1}g$.

HESSIAN IS NOT POSITIVE-DEFINITE

So far, we have considered the case when G^k is positive-definite. Difficulties arise if G^k is not positive-definite, because it is no longer clear what the quadratic model is telling us. In particular, if G^k is indefinite, the quadratic model indicates that $Q(\alpha p) \rightarrow -\infty$ as $\alpha \rightarrow \infty$, indicating that an infinite step should be taken when applied to $f(x)$ at x^k . Unfortunately, there is no universal agreement on how to define a Newton direction when the Hessian matrix G^k is indefinite. Various strategies exist that seem “reasonable” in this case, such as replacing p^k by $-p^k$, finding a direction of negative curvature, i.e., $(p^k)^T G^k p^k < 0$, or modifying the Hessian matrix, if necessary, to ensure that the modified Hessian matrix is positive-definite. In the latter case, if G^k is not “sufficiently” positive-definite, construct a “related” positive-definite matrix \bar{G}^k , and solve for the search direction using

$$\bar{G}^k p^k = -g^k.$$

In this way the resulting search direction is a descent direction; furthermore, the search direction will not be altered if the Hessian matrix is sufficiently positive definite.

CONVERGENCE

We expect good convergence from Newton’s method, when the quadratic function is a good approximation to the nonlinear function being minimized. In fact, for a general nonlinear function $f(x)$, Newton’s method has a quadratic rate of convergence to x^* if x^0 is sufficiently close to x^* , if the Hessian matrix is positive-definite

at x^* , and if the steplengths $\{\alpha_k\}$ converge to unity. A method is said to have a *quadratic rate of convergence* to x^* if there exists a constant $0 < \gamma < \infty$ such that

$$\gamma = \lim_{k \rightarrow \infty} \frac{\|x^{k+1} - x^*\|}{\|x^k - x^*\|^2}$$

where, $\lim_{k \rightarrow \infty} x^k = x^*$.

The outstanding local convergence properties of Newton's method make it a potentially attractive algorithm for unconstrained minimization. A further benefit of having the second derivatives available is that sufficient conditions for a minimum can be verified, namely, that the Hessian is positive-definite. In fact, Newton's method is often regarded as a standard against which other algorithms are measured. However, difficulties and even failure can occur if the quadratic model is not a good-enough approximation of $f(x)$.

If a modification \bar{G}^k of the Hessian matrix is used to ensure positive-definiteness, then starting at x^o (not necessarily "close" to a local minimizer x^*) and using \bar{G}^k coupled with an appropriate steplength approach, Newton's method converges to some x^* , a local minimizer, provided that $f(x)$ is twice continuously differentiable, $g(x)$ is uniformly continuous on the level set $\mathcal{L}(f(x_0))$ and that $\|G(x)\|$ is uniformly bounded on the level set $\mathcal{L}(f(x_0))$.

Definition (Level Set): For any function f and scalar β , the *level set* $\mathcal{L}(\beta)$ is the set of points x such that $f(x) \leq \beta$.

4.2 THE LINEAR LEAST-SQUARES PROBLEM

The linear least-squares problem is to

$$\text{Minimize}_{\pi \in \mathbb{R}^m} \|d - A^T \pi\|_2^2, \quad (4.6)$$

where A is an $m \times n$ matrix, d is an n -vector, and subscript 2 refers to the 2-norm. The least-squares solution π^* satisfies

$$AA^T \pi^* = Ad. \quad (4.7)$$

If we let

$$r^* = d - A^T \pi^* \quad (4.8)$$

denote the optimal residual then from (4.7) and (4.8) we obtain

$$Ar^* = Ad - AA^T \pi^* = 0. \quad (4.9)$$

Note that $A^T \pi^*$ is the projection of $d \in \mathbb{R}^n$ onto the space of the columns of A^T and Equation (4.9) states that the residual r^* lies in the null space of A . Rewriting

equations (4.8) and (4.9) in matrix notation we get

$$\begin{pmatrix} I & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} r^* \\ \pi^* \end{pmatrix} = \begin{pmatrix} d \\ 0 \end{pmatrix}. \quad (4.10)$$

▷ **Exercise 4.2** Show that the optimal residual r^* is given by

$$r^* = Pd \quad \text{where } P = I - A^T(AA^T)^{-1}A$$

is called the *projection matrix* because it projects any n -vector into the null space of A . Show that an alternative expression for the optimal residual r^* is given by:

$$r^* = ZZ^T d, \quad (4.11)$$

where Z is an $n \times (n - m)$ orthonormal matrix whose columns form a basis for the null space of the matrix A .

4.3 BARRIER FUNCTION METHODS

An idea for solving a minimization problem with inequalities is to replace the inequalities by a term, called a *barrier* term, appended to the objective function with a weight on it. The barrier term, as a function of x , has the property that it approaches $+\infty$ as any feasible interior point x approaches the boundary of the feasible region. Because we are minimizing, this property prevents the feasible iterates from crossing the boundary and becoming infeasible. However, the optimal solution to the original problem is typically a point on the boundary. To obtain such a boundary point solution, it is necessary to keep decreasing the parameter μ of the barrier function to 0 in the limit.

In this section we shall only consider the *logarithmic barrier function*.

4.3.1 THE LOGARITHMIC BARRIER FUNCTION

Consider the nonlinear inequality constrained problem

$$\begin{aligned} &\text{Minimize} && f(x) = z \\ &\text{subject to} && c_i(x) \geq 0, \quad i = 1, \dots, m, \end{aligned} \quad (4.12)$$

where $c_i(x) \geq 0$ are nonlinear constraints. The *logarithmic barrier function* is defined by

$$B(x, \mu) = f(x) - \mu \sum_{i=1}^m \ln(c_i(x)), \quad \mu > 0. \quad (4.13)$$

Note that $\ln(c_i(x))$ is not defined for $c_i(x) \leq 0$ and that for $c_i(x) > 0$,

$$-\ln(c_i(x)) \rightarrow +\infty \quad \text{as } c_i(x) \rightarrow 0.$$

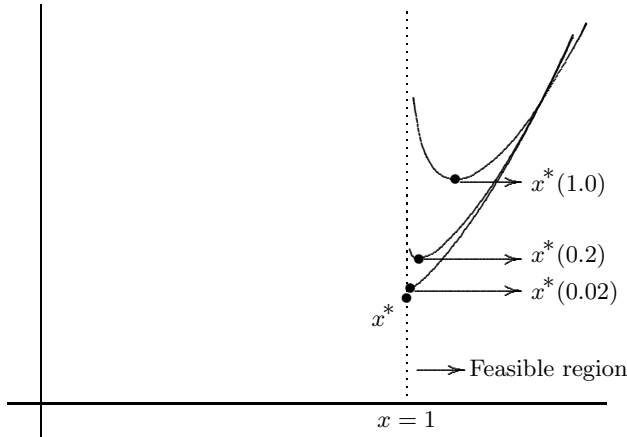


Figure 4-1: Barrier Function Method: Approach of $x^*(\mu)$ to x^*

If we let $x^*(\mu)$ denote an unconstrained minimizer of $B(x, \mu)$, then under mild conditions it can be shown that for $\mu > 0$ there exists a nonempty set \mathcal{S} such that

$$\lim_{\mu \rightarrow 0} x^*(\mu) = x^*, \quad \mu \in \mathcal{S}, \quad (4.14)$$

where x^* can be shown to be a local minimizer of (4.12).

Example 4.1 Consider the following univariate problem

$$\begin{aligned} \text{Minimize} \quad & x^2 = z \\ \text{subject to} \quad & x \geq 1 \end{aligned}$$

which has the unique solution $x^* = 1$. The barrier function for fixed μ for this problem is given by

$$B(x, \mu) = x^2 - \mu \ln(x - 1)$$

and the unconstrained minimizer of $B(x, \mu)$ for fixed μ is

$$x^*(\mu) = 1/2 + 1/2\sqrt{1 + 2\mu}.$$

The approach of the barrier function minimizers $x^*(\mu)$ to the unique solution $x^* = 1$ is illustrated in Figure 4-1.

Definition (Stationary Point): A *stationary point* of a function $f(x)$ is the point $x = \bar{x}$ where the $\nabla f(\bar{x}) = 0$.

▷ **Exercise 4.3** Show in Example 4.1 that $x(\mu) = 1/2 \pm 1/2\sqrt{1 + 2\mu}$ are stationary points of $B(x, \mu)$ for fixed μ and that $x^*(\mu) = 1/2 + 1/2\sqrt{1 + 2\mu}$.

4.3.2 PROPERTIES OF BARRIER FUNCTION METHODS

Successive minima $x^*(\mu)$ of the Barrier function $B(x, \mu)$ can be shown to have the following properties. Let $\bar{\mu} < \mu$ for sufficiently small μ , then

1. $B(x^*(\bar{\mu}), \bar{\mu}) < B(x^*(\mu), \mu)$.
2. $f(x^*(\bar{\mu})) \leq f(x^*(\mu))$.
3. $-\sum_{i=1}^m \ln(c_i(x^*(\bar{\mu}))) \geq -\sum_{i=1}^m \ln(c_i(x^*(\mu)))$.

See the Notes & Selected Bibliography Section (Section 4.8) where references to proofs can be found.

Given a fixed μ , at each iteration k with iterate $x^k(\mu)$, the Newton direction is given by

$$\nabla^2 B(x^k(\mu), \mu)p = -\nabla B(x^k(\mu), \mu).$$

Unfortunately, as μ goes to 0, the Hessian matrix at $x^k(\mu)$ becomes increasingly more *ill-conditioned* with singularity occurring in the limit. This makes it necessary to use some modification of the Hessian matrix to ensure positive-definiteness.

▷ **Exercise 4.4** Determine the Hessian matrix of $B(x, \mu)$ in Example 4.1 and show that it becomes increasingly ill-conditioned as $\mu \rightarrow 0$.

Furthermore, because of ill-conditioning in this approach, a specialized linesearch is required to take care of the case of an ill-conditioned Hessian matrix near the boundary. Thus, the unconstrained problems in general become more and more ill-conditioned and hence more and more difficult to solve.

If the feasible region is bounded and $f(x)$ is bounded from below over the feasible region then the barrier function is bounded from below over the feasible region. However, if the objective is unbounded over the feasible region, there is danger of the barrier function being unbounded, possibly leading to further complications.

At x^* , a minimizer of (4.12),

$$g(x^*) = \sum_{i=1}^m a_i(x^*)\lambda_i^*, \quad (4.15)$$

where λ_i^* are the Lagrange multipliers. Estimates of the Lagrange multipliers can be easily obtained by observing that at the solution $x^*(\mu)$ of $B(x, \mu)$, the gradient

$$g(x^*(\mu)) = \sum_{i=1}^m a_i(x^*(\mu)) \frac{\mu}{c_i(x^*(\mu))}, \quad (4.16)$$

where $a_i(x^*(\mu))$ is the gradient of $c_i(x)$ evaluated at $x^*(\mu)$. Thus, comparing (4.15) and (4.16) estimates of the Lagrange multipliers are given by

$$\lambda_i(\mu) = \frac{\mu}{c_i(x^*(\mu))}. \quad (4.17)$$

Under mild conditions, it can be shown that

$$\lim_{\mu \rightarrow 0} \frac{\mu}{c_i(x^*(\mu))} = \lambda_i^*, \quad \mu \in \mathcal{S}, \quad (4.18)$$

where \mathcal{S} is the same subset as that defined in (4.14).

4.4 THE PRIMAL LOGARITHMIC BARRIER METHOD FOR SOLVING LINEAR PROGRAMS

In this section, we apply a *barrier function* approach (combined with Newton's method for computing a search direction) to solve a linear programming problem in standard form.

$$\begin{aligned} &\text{Minimize} && c^T x = z \\ &\text{subject to} && Ax = b \\ &&& x \geq 0, \end{aligned} \quad (4.19)$$

where $x \in \Re^n$, A is an $m \times n$ matrix, and b is an m -vector, with $m < n$. This method is called the *primal logarithmic barrier method*. Each iteration requires the solution of a linear least-squares problem.

4.4.1 DETAILS OF THE METHOD

Replacing the objective in (4.19) by the logarithmic barrier function, we get the linearly constrained problem.

$$\begin{aligned} &\text{Minimize} && f(x) = c^T x - \mu \sum_{j=1}^n \log(x_j) \\ &\text{subject to} && Ax = b. \end{aligned} \quad (4.20)$$

Note that the conditions $x_j \geq 0$ are no longer needed because $\ln(x_j)$ is not a real-valued function if $x_j < 0$ and because, starting with $x_j > 0$, $\ln(x_j) \rightarrow -\infty$ as $x_j \rightarrow 0$. This latter property means that $\ln(x_j)$ serves as a barrier discouraging x_j from going to 0.

The first and second derivatives of the barrier function are given by:

$$g(x) = c - \mu D_x^{-1} e \quad \text{and} \quad G(x) = \mu D_x^{-2}, \quad (4.21)$$

where

$$D_x = \text{Diag}(x) \quad (4.22)$$

and $e = (1, 1, \dots, 1)^T$. Note that the gradient $g(x)$ and Hessian $G(x)$ are defined only for $x_j > 0$ for all j .

▷ **Exercise 4.5** Show that $G(x)$ is positive-definite.

Let $x = \bar{x} > 0$ satisfying $Ax = b$ be an initial feasible solution to (4.20). As we have seen, in Equation (4.1), to obtain a Newton search direction Δx from \bar{x} we first approximate $f(x)$ by a quadratic at \bar{x} , i.e.,

$$Q(\Delta x) = g(\bar{x})^T \Delta x + \frac{1}{2} \Delta x^T G(\bar{x}) \Delta x \quad (4.23)$$

where $g(x)$ and $G(x)$ are given by (4.21), and we minimize this quadratic to obtain Δx . However, (4.20) requires we minimize $Q(\Delta x)$ subject to Δx satisfying $A(\bar{x} + \Delta x) = b$. Hence we obtain Δx as the solution to

$$\begin{aligned} \text{Minimize} \quad & Q(\Delta x) = g(\bar{x})^T \Delta x + \frac{1}{2} \Delta x^T G(\bar{x}) \Delta x \\ \text{subject to} \quad & A \Delta x = 0, \end{aligned} \quad (4.24)$$

with $g(x)$ and $G(x)$ defined by (4.21). If we let π_x denote the Lagrange multipliers for the constraints $A \Delta x = 0$, then the Lagrangian is

$$L(x, \pi_x) = g(\bar{x})^T \Delta x + \frac{1}{2} \Delta x^T G(\bar{x}) \Delta x - \pi_x^T A \Delta x. \quad (4.25)$$

Because the Hessian matrix $G(\bar{x})$ is positive-definite, we obtain the conditions for optimality by setting the partials with respect to Δx of the Lagrangian to zero, namely,

$$g(\bar{x}) + G(\bar{x}) \Delta x = A^T \pi_x. \quad (4.26)$$

Thus, at an optimum, the gradient $\nabla Q(\Delta x) = g(\bar{x}) + G(\bar{x}) \Delta x$ is a linear combination π_x of the rows of A . Substituting for $g(\bar{x})$ and $G(\bar{x})$ from (4.21), we obtain

$$c - \mu D_{\bar{x}}^{-1} e + \mu D_{\bar{x}}^{-2} \Delta x = A^T \pi_x,$$

which, after rearranging, becomes

$$\mu D_{\bar{x}}^{-2} (-\Delta x) + A^T \pi_x = c - \mu D_{\bar{x}}^{-1} e. \quad (4.27)$$

Thus the solution of (4.24) satisfies the following equations:

$$\begin{pmatrix} \mu D_{\bar{x}}^{-2} & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} -\Delta x \\ \pi_x \end{pmatrix} = \begin{pmatrix} c - \mu D_{\bar{x}}^{-1} e \\ 0 \end{pmatrix} \quad (4.28)$$

If we define a vector r_x by

$$D_{\bar{x}} r_x = -\mu \Delta x, \quad (4.29)$$

we get

$$\begin{pmatrix} I & D_{\bar{x}} A^T \\ A D_{\bar{x}} & 0 \end{pmatrix} \begin{pmatrix} r_x \\ \pi_x \end{pmatrix} = \begin{pmatrix} D_{\bar{x}} c - \mu e \\ 0 \end{pmatrix}. \quad (4.30)$$

This system may be too large to solve directly in practice. Comparing (4.30) with (4.6) and (4.10), we see that an alternative way to obtain r_x is by solving the least-squares problem

$$\text{Minimize}_{\pi} \quad \|D_{\bar{x}}c - \mu e - D_{\bar{x}}A^T\pi\|_2^2 \quad (4.31)$$

for π and setting

$$r_x = D_xc - \mu e - D_xA^T\pi. \quad (4.32)$$

An efficient and numerically stable way to solve (4.31), the least-squares problem, is by using the QR factorization.

The Newton barrier direction is then given by

$$\Delta x = -(1/\mu)D_{\bar{x}}r_x. \quad (4.33)$$

Thus, the search direction at each iteration is obtained as the solution of the linear least-squares problem defined by (4.31). Letting $x^k = \bar{x}$, the new approximation to the solution of (4.20) is then given by

$$x^{k+1} = x^k + \alpha\Delta x, \quad (4.34)$$

where α , the steplength, is obtained by a steplength algorithm (such as the one described in Step 3 of Algorithm 4.1). Finally, note that because we could use a steplength algorithm to determine α , we could redefine the search direction (4.33) by

$$\bar{\Delta}x = -D_xr_x \quad (4.35)$$

for $\mu > 0$. In this case, the iterative process is terminated when the iterates are getting “close” or when the duality gap is “sufficiently small.”

- ▷ **Exercise 4.6** Show that the Newton search direction for the primal logarithmic barrier method can be written as

$$\Delta x = -\frac{1}{\mu}D_xPD_xc + D_xPe, \quad (4.36)$$

where $e = (1, 1, \dots, 1)^T$ and where P , called the projection matrix, is

$$P = I - D_xA^T(AD_xA^T)^{-1}AD_x. \quad (4.37)$$

Definition (Affine Transformation): Let M be an $n \times n$ nonsingular matrix and let $d \in \Re^n$. The transformation $y = Mx + d$ is called an *affine transformation* of x into y .

Instead of (4.36), some authors have proposed the search direction

$$\hat{\Delta}x = -D_xPD_xc, \quad (4.38)$$

that is the direction of Δx in the limit $\mu \rightarrow 0$. In the literature, a method that uses $\hat{\Delta}x$ is called a *primal affine* method or Dikin’s method.

- ▷ **Exercise 4.7** Develop a Newton barrier method for the dual of the linear program in standard form. What is the dual affine direction, i.e., the direction as $\mu \rightarrow 0$.

4.4.2 INITIAL FEASIBLE SOLUTION

So far we have assumed that an initial feasible interior-point solution is available. If an initial feasible interior solution is not available, we can easily generate one by picking an $x^o > 0$ and setting up the following linear program with one artificial variable x_a and associated large cost M :

$$\begin{array}{ll} \text{Minimize} & c^T x + Mx_a = z \\ \text{subject to} & Ax + (b - Ax^o)x_a = b \quad x \geq 0, x_a \geq 0. \end{array} \quad (4.39)$$

Then $x = x^o > 0$ and $x_a = 1$ is clearly a feasible solution to (4.39). On application of an algorithm, if M is sufficiently large, then at an optimal solution we must have $x_a = 0$. Refer to the comments in Section 3.3.5 to see how the choice of M plays a role in practice.

▷ **Exercise 4.8** Show that for M sufficiently large, assuming a feasible solution to $Ax = b$, $x \geq 0$ exists, then the set of optimal solutions to (4.39) is the same as that for the corresponding linear program in standard form.

While any $x^o > 0$ can be chosen it would be nice to choose a “good” starting x^o . If something is known about the problem, then it is possible that a very good educated guess can be made for a choice of x^o . On the other hand, if this is not at hand, then one possibility might be to choose \hat{x}^o by

$$\hat{x}^o = A^T(AA^T)^{-1}b, \quad (4.40)$$

which has the property that $A\hat{x}^o = b$ but not necessarily $\hat{x}^o > 0$. If $x^o > 0$ then set $x^o = \theta$ where $\theta > 0$ is chosen arbitrarily. That is, to ensure an $x^o > 0$, set the components

$$x_j^o = \begin{cases} \hat{x}_j^o & \text{if } \hat{x}_j^o \geq \theta \\ \theta & \text{if } \hat{x}_j^o < \theta \end{cases} \quad (4.41)$$

Of course, if all $\hat{x}_j^o \geq \theta$ then the artificial variable x_a is not needed.

▷ **Exercise 4.9** Show how to generate an artificial initial feasible interior solution for the dual of a linear program in standard form if an initial feasible interior solution is not given.

4.5 PRIMAL-DUAL LOGARITHMIC BARRIER METHODS

Primal-dual logarithmic barrier algorithms have been reported to outperform the Simplex Algorithm on very large-scale linear programs. To derive the algorithm we apply the barrier method to the dual of the linear program in standard form. Recall that an LP in standard form is $\min c^T x$ subject to $Ax = b$, $x \geq 0$, and its dual

is to find $\max b^T y$, subject to $A^T y \leq c$. Letting $\mu > 0$ be a given fixed parameter (which is decreased during the course of the algorithm) and inserting slack variables $s = (s_1, \dots, s_j, \dots, s_n)^T \geq 0$, the barrier method formulation of the dual is:

$$\begin{aligned} \text{Maximize} \quad & b^T y + \mu \sum_{j=1}^n \ln(s_j) \\ \text{subject to} \quad & A^T y + \quad \quad \quad s = c. \end{aligned} \quad (4.42)$$

Note that the conditions $s_j \geq 0$ are no longer needed because $\ln(s_j)$ is not a real-valued function if $s_j < 0$ and because, starting with $s_j > 0$, $\ln(s_j) \rightarrow -\infty$ as $s_j \rightarrow 0$. This latter property means that $\ln(s_j)$ serves as a barrier discouraging s_j from going to 0.

Denoting by x the vector of Lagrange multipliers, the Lagrangian of the preceding problem is:

$$L(x, y, s, \mu) = b^T y + \mu \sum_{j=1}^n \ln(s_j) - x^T (A^T y + s - c). \quad (4.43)$$

To write the first-order necessary conditions for a minimum, we set the partial derivatives of the Lagrangian with respect to x , y , and s to zero. This results in the following three sets of equations:

$$A^T y + s = c \quad (4.44)$$

$$Ax = b \quad (4.45)$$

$$D_x D_s e = \mu e \quad (4.46)$$

where (4.46) is the same as $\mu D_s^{-1} e = D_x e$ rewritten with

$$D_s = \text{Diag}(s) \quad \text{and} \quad D_x = \text{Diag}(x). \quad (4.47)$$

Note that (4.44) is the usual dual feasibility condition with $s \geq 0$ omitted and (4.45) is the usual primal feasibility condition with $x \geq 0$ omitted. Note that (4.46) states that $x_i s_i = \mu$ for $i = 1, \dots, n$ and thus, in the limit as $\mu \rightarrow 0$, results in the usual complementary slackness conditions.

To initiate we pick $\mu > 0$, say $\mu = 1$, and assume we have a starting interior dual feasible solution $(y, s) = (y^o, s^o)$ with $s^o > 0$ available, as well as a primal feasible solution $x = x^o$ available. Since μ is arbitrarily chosen equal to 1, the solution will in general not satisfy (4.46). In this case we determine a search direction $(\Delta x, \Delta y, \Delta s)$ satisfying (4.44), (4.45) and (4.46). Any movement $\alpha \geq 0$ along such a direction satisfies the feasibility conditions (4.44) and (4.45) but does not necessarily maintain the nonnegativity of $x^o + \Delta x$ and $s^o + \Delta s$. In place of the current solution (x, y, s) , we substitute the proposed new solution $(x + \Delta x, y + \Delta y, s + \Delta s)$ in (4.44) and (4.45), to obtain the Newton equations:

$$A^T \Delta y + \Delta s = 0 \quad (4.48)$$

$$A \Delta x = 0 \quad (4.49)$$

Substituting the new solution into (4.46) we obtain

$$(x_i + \Delta x_i)(s_i + \Delta s_i) = \mu \quad \text{for } i = 1, \dots, n. \quad (4.50)$$

In order to satisfy the computations we ignore the second-order term $\Delta x_i \Delta s_i$ for $i = 1, \dots, n$, we obtain the first-order approximation to (4.50) as

$$x_i^o s_i^o + x_i^o \Delta s_i + s_i^o \Delta x_i = \mu,$$

which in matrix notation is

$$D_{s^o} \Delta x + D_{x^o} \Delta s = \mu e - D_{x^o} D_{s^o} e. \quad (4.51)$$

We next solve this linear system of equations (4.48), (4.49), and (4.51) for $(\Delta x, \Delta y, \Delta s)$. From (4.48) we obtain

$$\Delta s = -A^T \Delta y. \quad (4.52)$$

From (4.51) we obtain, after noting that $D_{s^o}^{-1} D_{x^o} = D_{x^o} D_{s^o}^{-1}$,

$$\Delta x = D_{s^o}^{-1} (\mu e - D_{x^o} D_{s^o} e) - D_{x^o} D_{s^o}^{-1} \Delta s. \quad (4.53)$$

Next, noting $A \Delta x = 0$, we multiply (4.53) on the left by A and substitute $\Delta s = -A^T \Delta y$, from (4.52), in it to obtain:

$$0 = A \Delta x = A D_{s^o}^{-1} (\mu e - D_{x^o} D_{s^o} e) + A D_{x^o} D_{s^o}^{-1} A^T \Delta y$$

Hence

$$\Delta y = -(A D_{x^o} D_{s^o}^{-1} A^T)^{-1} A D_{s^o}^{-1} (\mu e - D_{x^o} D_{s^o} e). \quad (4.54)$$

Because we have a separate search direction Δx in the primal space and a separate search direction $(\Delta y, \Delta s)$ in the dual space we can have a separate steplength α_p for Δx and α_d for $(\Delta y, \Delta s)$ respectively. One possible way to compute the steplengths is to compute the maximum step α_p^{\max} that maintains $x + \alpha_p^{\max} \Delta x \geq 0$ and the maximum step α_d^{\max} that maintains $s + \alpha_d^{\max} \Delta s \geq 0$. Next choose a multiple $\rho < 1$ of the maximum steplength

$$\alpha_p = \rho \alpha_p^{\max} \quad \text{if } \alpha_p^{\max} \leq 1, \quad \text{else } \alpha_p = 1. \quad (4.55)$$

$$\alpha_d = \rho \alpha_d^{\max} \quad \text{if } \alpha_d^{\max} \leq 1, \quad \text{else } \alpha_d = 1. \quad (4.56)$$

In practice ρ is chosen very close to 1, for example, 0.99995. Note that $\alpha_p = 1$ or $\alpha_d = 1$ corresponds to an unconstrained Newton step.

The only items that remain to be discussed are when to terminate and when to decrease the parameter μ . At some iteration, assume that we have a feasible primal solution $\bar{x} \geq 0$ satisfying $A\bar{x} = b$ and a dual feasible interior solution (\bar{y}, \bar{s}) with $s > 0$ and $A^T \bar{y} + \bar{s} = c$. If the duality gap $c^T \bar{x} - \bar{b}^T \bar{y}$ is deemed sufficiently close to 0 we terminate. (Note that, because μ is a user-defined parameter a duality gap equal to 0 does not imply satisfaction of (4.46) because μ may not be small;

however, satisfaction of (4.46) for very small μ always implies a duality gap that is very small.) If, on the other hand, the duality gap is not close to 0 and (4.46) holds with μ not sufficiently close to zero, then we are at a stationary point of the Lagrangian but not at an optimal solution to the linear program. Then, in order to continue, we decrease μ by some specified factor, for example, $\mu \leftarrow 0.1 \times \mu$; this means that (4.46) is no longer satisfied and the iterative process can be continued.

Note that an affine variant of the algorithm can be developed by setting $\mu = 0$ in the equations for $(\Delta x, \Delta y, \Delta s)$. In computing steplengths, affine variants typically use a somewhat smaller value of ρ , for example $\rho = 0.95$.

▷ **Exercise 4.10** Show how to generate an initial feasible interior solution for the primal-dual approach using artificial variables (see Section 4.4.2).

▷ **Exercise 4.11** In the preceding discussion, do not assume that we have a feasible solution (x^o, y^o, s^o) to (4.44) and (4.45). Derive the Newton search directions $(\Delta x, \Delta y, \Delta s)$ in this case.

How are the primal log barrier, dual log barrier, and primal-dual log barrier methods related to the first-order condition (4.46)? See Exercise 4.12.

▷ **Exercise 4.12** Notice that the condition $D_x D_s = \mu e$ can also be written as $D_s e = \mu D_x^{-1} e$, $D_x e = \mu D_s^{-1} e$, or $e = \mu D_x^{-1} D_s^{-1} e$. Therefore the second-order terms can be dropped in four different ways leading to four different algorithms: The primal-dual logarithmic method we have just seen and the other three:

1. If we use the form $D_s e = \mu D_x^{-1} e$, show that we obtain the primal log barrier method of Section 4.4.
2. If we use the form $D_x e = \mu D_s^{-1} e$, show that we obtain the dual log barrier method of Exercise 4.7.
3. Develop a barrier method associated with $e = \mu D_x^{-1} D_s^{-1} e$.

4.6 RECOVERING A BASIC FEASIBLE SOLUTION

When the optimal solution is unique, an interior-point method will terminate with the unique extreme-point optimal solution. However, in the more likely case of non-uniqueness (dual degeneracy), an interior-point method will typically not terminate at an extreme-point (basic feasible) solution; instead it will typically terminate at a point somewhere on the optimal face of the feasible region.

Although a nonextreme point solution may be acceptable in some applications, there are other situations when it is desirable to obtain a basic feasible solution. For example, when attempting to obtain integer solutions by using linear program relaxations, it is desirable to have a basic feasible solution because it will have *fewer*

nonzeros than an interior-point solution. In commercial applications, too, it is desirable to have plans that require fewer nonzero activity levels. In some commercial applications several activities are bundled together to reduce the number of potentially different activities that may result with nonzero levels. Many commercial applications require the solution of several related linear programs; in this case the optimal solution of one linear program can be used to start the optimization process for a subsequent linear program. Typically, in such situations, the Simplex Method can solve such problems very efficiently and quickly. If so, it may make sense to solve the initial problem with an interior-point method; convert the solution to a basic feasible solution, and use the corresponding basis to solve subsequent linear programs by the Simplex Method.

Hence it is important to know how to recover an optimal basic feasible solution from an optimal interior-point solution, assuming one exists. The recovery of a basic feasible solution can be done by an approach discussed by Dantzig as an exercise in his 1963 book. Assuming the linear program is in standard form and an interior-point solution has been found, start by throwing away all the columns that have zero activity levels. Next pick a basic set of columns (or create a basis with all artificial variables). Then, for each of the remaining columns j not already in the basis attempt to bring it into the basis. It will either enter the basis and drive a current basic column to zero or be driven to zero. Clearly the number of iterations in this procedure is bounded by the number of positive activity level columns not in the basis (at most n). The approach is summarized in the following algorithm.

Algorithm 4.2 (Converting an Optimal Solution to a Basic Optimal Solution)

Given an optimal solution $\hat{x}^* \geq 0$ satisfying $A\hat{x}^* = b$, delete the columns $\hat{A}_{\bullet j}$ corresponding to $\hat{x}_j^* = 0$. Let the resulting coefficient matrix be denoted by A and the corresponding solution by x^* , where $x^* > 0$ satisfies $Ax^* = b$.

1. *Initialize.* Set $k \leftarrow 0$, $x^k \leftarrow x^*$.
2. *Pick a basis B* from among the columns of A assuming that A is of full row rank. Let $B^k = (A_{\bullet j_1}, A_{\bullet j_2}, \dots, A_{\bullet j_m})$. (See Exercise 4.13 for how to handle the degenerate case and less-than-full-rank-case.)
3. *Pick Nonbasic Columns.* Let \mathcal{N}^k be the set of nonbasic columns of A .
4. *Check for Termination.* If $\mathcal{N}^k = \emptyset$, stop and report x^k as the optimal basic feasible solution and report B^k as the corresponding optimal basis.
5. *Select Incoming Column.* Pick any $s \in \mathcal{N}^k$ such that $x_s^k > 0$.
6. *Select Outgoing Column.* Decrease $x_s^k \geq 0$ as much as possible and adjust the basic variables to maintain feasibility while all remaining nonbasics are held fixed. The detailed steps are as follows.
 - (a) Determine y as the solution to $B^k y = A_{\bullet s}$.
 - (b) Set $\theta_{\max} = \text{Max}_{\theta} [(x_s^k - \theta) \geq 0, (x_{j_1}^k + \theta y_1) \geq 0, \dots, (x_{j_m}^k + \theta y_m) \geq 0]$.
7. *Update Basis and Nonbasic Set.*
 - (a) If $\theta_{\max} = x_s^k$, set $B^{k+1} = B^k$, $x_{j_i}^{k+1} = x^k j_i + \theta y_i$ for $i = 1, \dots, m$, and $\mathcal{N}^{k+1} = \mathcal{N}^k - \{s\}$.

- (b) If $x_{j_r} + \theta_{\max} y_r = 0$, replace the basic variable j_r by s and set $x_{j_r}^k = x_s^k - \theta_{\max}$ and set $x_{j_i}^k = x_{j_i}^k + \theta_{\max} y_i$ for all $i \neq r$. (If r is not unique then choose the smallest such r .) Set $\mathcal{N}^{k+1} = \mathcal{N}^k - \{s\}$.

8. *Update an Loop Back.* Set $k \leftarrow k + 1$ and go to Step 4.

▷ **Exercise 4.13** Here we address the issues of rank and degeneracy.

1. If the coefficient matrix A is not of full rank, then we cannot pick a basis B in the algorithm from the columns of A . Show, in this case, how to modify the algorithm in one of two ways:

- (a) By augmentation with artificial variables.
 (b) By dropping the redundant row.

2. Suppose that the linear program has a degenerate primal optimal solution. Show how to modify the algorithm to take degeneracy into account.

▷ **Exercise 4.14** Suppose we are given an interior-point solution \bar{x} that is not basic and not necessarily optimal. Show how to modify Algorithm 4.2 to construct a basic feasible solution \hat{x} in n or fewer steps such that $c^T \hat{x} \leq c^T \bar{x}$. Consider both the degenerate and the nondegenerate cases.

4.7 COMPUTATIONAL COMMENTS

Computational comments 4, 5, and 6 made in the context of Karmarkar's algorithm in Section 3.3.5 also apply to interior-point methods in general.

Besides being able to obtain a solution with fewer iterations than Simplex iterations, the key to being able to develop a computationally efficient interior-point method lies in being able to solve, at each iteration, a problem of the form $AD^2A^T p = d$, where $D = \text{Diag}(x)$ is a diagonal matrix with positive diagonal terms. Although the systems are similar for the various interior-point methods, the diagonal matrix and the right-hand side tend to be different. Depending on the method, one of the following three systems is solved at each iteration:

$$AD^2A^T p = Ad \quad (4.57)$$

$$AD^2A^T p = d \quad (4.58)$$

$$ADp = d, \quad \text{and } p \text{ chosen as } \min \|p\|. \quad (4.59)$$

System (4.57) is solved at each iteration in projective methods and in the primal log barrier Newton methods. System (4.59) is solved at each iteration in the dual log barrier and primal-dual log barrier methods. System (4.58) is used to find an initial point; see Section 4.4.2. The techniques for solving these systems are similar but the numerical properties can be quite different. Numerical difficulties can arise due to AD becoming increasingly ill-conditioned because many of the diagonal elements

in D tend to zero as the optimal solution is approached. Numerically, the solution to (4.59) is “better defined” than the solution to (4.58), which, in turn, is “better defined” than the solution to (4.57). By “better defined,” we mean that small changes in the values of the parameters cause less fluctuation in the solution.

In order for an interior method to be considered efficient, systems such as $AD^2A^T p = d$ need to be solved very accurately and fast with low storage requirements. One approach that is fast and uses low storage is the *Cholesky factorization*, which is defined for a positive-definite and symmetric matrix. The matrix AD^2A^T is clearly symmetric; it is positive definite if A is full rank and D has a strictly positive diagonal. The Cholesky Factorization of the symmetric positive-definite matrix AD^2A^T is of the form

$$AD^2A^T = LL^T \quad (4.60)$$

where L is lower triangular. An alternative, equivalent way to write the Cholesky factorization is

$$AD^2A^T = \bar{L}\Gamma\bar{L}^T \quad (4.61)$$

where \bar{L} is a unit lower-triangular matrix (i.e., it is a lower-triangular matrix with all ones on the diagonal) and Γ is a diagonal matrix of all positive diagonal elements. This latter form is preferred because it avoids computing square roots.

▷ **Exercise 4.15** Suppose we write the LU factorization of AD^2A^T as $L\Theta U$ where L is unit lower-triangular, Θ is a diagonal matrix, and U is unit upper-triangular. Show in this case that the factors $L = \bar{L}$, $\Theta = \Gamma$, and $U = \bar{L}^T$ (that is, the LU factors are the same as the Cholesky factors), *provided* that interchanges of rows i and j are accompanied by a corresponding interchange of columns i and j for the same pairs in both factorizations.

Better numerical stability is achieved using the QR factorization, i.e.,

$$DA^T = QR \quad (4.62)$$

where Q is an orthornormal square matrix and R is an upper-triangular matrix. Then $AD^2A^T = R^TR$. The computational work to do a QR factorization is usually two to three times more than for a Cholesky factorization.

4.8 NOTES & SELECTED BIBLIOGRAPHY

For proofs on convergence of various iterative algorithms, see, for example, Ortega & Rheinboldt [1970]. For details and computational results using the Projected Newton Barrier Method, see Gill, Murray, Saunders, Tomlin, & Wright [1986]; this paper also shows that the projected Newton Barrier method and Karmarkar’s method are related. Renegar [1988] showed polynomial time convergence for a Newton-type algorithm applied to the linear program $\min c^Tx$, subject to $Ax \geq b$. Gonzaga [1989] and Shanno & Bagchi [1990] showed that Karmarkar’s method is just a special case of the logarithmic barrier function method; Karmarkar showed polynomial complexity for his method but Shanno and Bagchi

did not for theirs. Anstreicher [1990] has shown that the application to linear programs of Frisch's [1957] logarithmic barrier approach (as developed by Fiacco & McCormick [1968]) has polynomial complexity. See also Freund [1991a] for results on the theoretical efficiency of a shifted barrier function approach.

The primal-affine method, developed by Barnes [1986] and Vanderbei, Meketon, & Freedman [1986], defines the search direction by $p = -D_x P D_x c$ where $D_x = \text{Diag}(x)$ and P is defined by (4.37). It also turns out that this method is the same as the method proposed earlier by Dikin [1967, 1974]. Several interesting results on Dikin's method have appeared since its rediscovery and reclassification as the primal-affine method. Saigal [1993a] and Tsuchiya & Monteiro [1996] have shown that the primal-affine method converges superlinearly. Examples have been constructed by various researchers to show that the primal-affine method does not converge when the primal problem is degenerate. Mascarenhas [1993] has constructed an example in which the method does not converge to an optimal solution if a steplength of $\alpha = 0.999$, called the long-step, is used because when a steplength of $\alpha = 0.999$ is used the iterates stay too close to the boundary. Hall & Vanderbei [1993] have constructed an example where the dual iterates failed to converge if steplength $\alpha > 2/3$ is used; see Problem 4.2 on Page 146. Saigal [1993b] describes a variant of the method called the *power variant* of the primal-affine method.

Notice, in the limit as $\mu \rightarrow 0$, the Newton search direction (4.36) is $p = -D_x P D_x c$, the primal-affine method search direction. An explanation of the relationship between the primal logarithmic Newton barrier method and primal affine method is as follows. The idea of an interior-point method is to keep the iterates in the interior of the feasible region defined by inequalities. As described by Sonnevend [1986], we can find the "analytic center" of the feasible region by solving the problem: $\min -\sum_{j=1}^n \ln(x_j)$, subject to $Ax = b$. For this problem, the Newton direction is $p = D_x P e$, where $D_x = \text{Diag}(x)$ and P is defined by (4.37). This is reflected in the Newton direction (4.36), which is a combination of an affine term that points in the direction of optimality and a centering term that causes the iterates to stay away from the boundary. Hertog & Roos [1991] have shown, in their survey of search directions for interior-point methods, that most of these methods use search directions that are a combination of a component that points toward an optimal solution and a centering term.

The approach described in Section 4.4.2 to determine an initial feasible interior x^0 is described in Lustig, Marsten, & Shanno [1994].

Another class of interior-point methods operates on $\max b^T y$ subject to $A^T y \leq c$, the dual of a linear programming in standard form. Huard's [1970] method of centers was first applied to this problem by Renegar [1988], who obtained an algorithm that takes $O(\sqrt{n}\mathcal{L})$ iterations, where n is the dimension of x and \mathcal{L} is the total number of bits required to store the problem's data in the computer. Later Gonzaga [1992] showed that this was a special case of the logarithmic barrier function method. Notice that the dual of a linear program has no equality constraints, hence the use of a logarithmic barrier function results in the following unconstrained problem:

$$\max_y b^T y + \mu \sum_{j=1}^n \ln(c_j - A_{\bullet j}^T y), \quad (4.63)$$

where $A_{\bullet j}$ is the j th column of the coefficient matrix A . Letting $s = c - A^T y$, the first and second derivatives of the barrier function are given by

$$g(y) = b - \mu A D_s^{-1} e \quad \text{and} \quad G(y) = -\mu A D_s^{-2} A^T, \quad (4.64)$$

where

$$D_s = \text{Diag}(s) = \text{Diag}(c - A^T y), \quad (4.65)$$

and $e = (1, 1, \dots, 1)^T$. Note that $g(y)$ and $G(y)$ are defined only if $s_j = c_j - A_{\bullet j}^T y > 0$ for all j . The Newton search direction (see Exercise 4.7) is given by

$$\Delta y = -\frac{1}{\mu} (AD_s^{-2} A^T)^{-1} b + (AD_s^{-2} A^T)^{-1} AD_s^{-1} e. \quad (4.66)$$

The first term of this search direction Δy points toward optimality and the second term does the centering. A dual-affine variant search direction can be found by letting $\mu \rightarrow 0$, namely

$$\Delta y_A = -(AD_s^{-2} A^T)^{-1} b. \quad (4.67)$$

Shortly after the publication of Karmarkar's [1984] result, Adler, Resende, Veiga, & Karmarkar [1989] implemented an interior-point method, called the dual-affine variant, to solve the dual problem. The implementation demonstrated superiority over an earlier version of MINOS, a simplex-based code, on a class of problems distributed by Gay [1985]. A description of the data structures used in the approach can be found in Adler, Karmarkar, Resende, & Veiga [1990]. Later McShane, Monma, & Shanno [1989] demonstrated similar results using a primal-dual path-following algorithm. Marsten, Saltzman, Shanno, Ballinton, & Pierce [1989] did further interesting computational work using the dual-affine approach. Tsuchiya & Muramatsu [1995] prove convergence of the primal-affine and dual-affine scaling methods without the nondegeneracy assumptions. Their proof is based on Dikin's [1990, 1992] results. Since then, simpler proofs have been developed by Monteiro, Tsuchiya, & Wang [1993] and Saigal [1992].

Megiddo [1986, 1988] first devised the theory for primal-dual interior-point methods (see Section 4.5 for details of the method), which has performed very well in practice. Based on Megiddo's theory, Kojima, Mizuno, & Yoshise [1989a] developed an $O(n\mathcal{L})$ algorithm, where \mathcal{L} is the total number of bits required to store the problem's data in the computer. Shortly thereafter, Lustig, Marsten, & Shanno [1990, 1991a, 1991b, 1992a, 1992b, 1992c] implemented and reported promising results for various versions of a primal-dual algorithm. Primal-dual algorithms were first implemented with different steplengths in the primal and dual spaces by McShane, Monma, & Shanno [1989]. Lustig [1991] and Lustig, Marsten, & Shanno [1994] showed how to derive directions (without the need for artificial variables) when an initial feasible solution is not easily available for application of the primal-dual algorithm. Choi, Monma, & Shanno [1990] show how to conveniently handle upper and lower bounds on variables in primal-dual interior-point methods. Lustig, Marsten, & Shanno [1994] show that it is possible to handle free (unrestricted) variables in a primal-dual interior-point method by splitting it into the difference of its positive and negative parts (similar to the way discussed in *Linear Programming 1*, for the regular Simplex Method) and then at each iteration setting the smaller of the two variables to a constant thereby shifting the origin of the unrestricted variable. Shortly after Lustig, Marsten, & Shanno's [1991a] implementation of the primal-dual interior-point method, Mehrotra [1992a] devised a predictor-corrector method, that utilizes a combination of three search directions: the predictor, the corrector, and the centering direction. Assuming that the current solution (x, y, s) does not necessarily satisfy the first-order corrections of optimality we substitute $(x + \Delta x, y + \Delta y, s + \Delta s)$ in place of the current solution (x, y, s) in (4.44), (4.45), and (4.46) to obtain the Newton equations

$$A^T \Delta y + \Delta s = c - A^T y - s \quad (4.68)$$

$$A\Delta x = b - Ax \quad (4.69)$$

$$D_s\Delta x + D_x\Delta s = \mu e - D_x D_s e - D_{\Delta x} D_{\Delta s} e. \quad (4.70)$$

Mehrotra proposed first solving the above system with μ set to 0 and $D_{\Delta x} D_{\Delta s} e$ dropped. The resulting solution $(\Delta x, \Delta y, \Delta s) = (\Delta x, \bar{\Delta y}, \bar{\Delta s})$ is then used to substitute for the product $D_{\Delta x} D_{\Delta s}$ and the system re-solved. In this method μ is adjusted to be small when the affine direction produces a large decrease in complementarity (see (4.46)) from the previous iteration, and μ is adjusted to be large when the affine direction produces a small decrease in complementarity. The iterative process of substituting for $D_{\Delta x} D_{\Delta s} e$ described earlier can be repeated to obtain a better search direction $(\Delta x, \Delta y, \Delta s)$ but requires a lot more work. Carpenter, Lustig, Mulvey, & Shanno [1993] have examined this *multiple corrections procedure* and concluded that for a general primal-dual algorithm one such iteration works best in terms of overall execution time. Lustig, Marsten, & Shanno [1992a] prove that the algorithm will have guaranteed convergence if the predictor-corrector approach is used. They further show how to determine whether a regular primal step should be taken or a correction step performed. They report that the test has little computational effect on the problems even though it is necessary to prove guaranteed convergence. In another paper, Lustig, Marsten, & Shanno [1992c] describe an implementation of the primal-dual algorithm with a slightly modified Mehrotra [1992a] predictor-corrector method. Mitchell & Borchers [1992] have applied the primal-dual interior-point algorithm in a cutting-plane setting.

For Megiddo's [1986, 1988] primal-dual method, Monteiro & Adler [1989a] and Kojima, Mizuno, & Yoshise's [1989b] improved Kojima, Mizuno, & Yoshise's [1989a] result to $O(\sqrt{n}\mathcal{L})$ complexity, where \mathcal{L} is the total number of bits required to store the problem's data in the computer; unfortunately in their algorithm μ is being reduced so slowly that in practice the algorithm is very inefficient. Lustig, Marsten, & Shanno [1990] present empirical evidence that their implementation appears to result in $O((\log n)\mathcal{L})$ iterations. Todd [1994b] proves that a long-step primal-dual algorithm similar to an earlier version of the Lustig, Marsten, Shanno [1994] OB1 algorithm may require $O(n^{1/3})$ iterations before achieving a reasonable constant improvement in the duality gap.

All the complexity results mentioned so far for primal-dual algorithms are for algorithms using strictly feasible points. In 1991, Kojima, Megiddo, & Mizuno [1993] showed global convergence for an infeasible primal-dual method under special conditions. Mizuno [1992] and, earlier in 1992, Zhang [1994] provided a polynomial time complexity bound for the Kojima-Megiddo-Mizuno infeasible interior-point algorithm. Lustig, Marsten, & Shanno [1992a] show global convergence for the predictor-corrector algorithm. Complexity bounds for variants of the Mehrotra predictor-corrector method are established by Zhang & Zhang [1995]. Mizuno, Kojima, & Todd [1995] developed an infeasible interior-point primal-dual potential reduction algorithm that is polynomial time bounded.

Superlinear convergence of a primal-dual algorithm has been shown by S. Wright [1993]. Zhang, Tapia, & Dennis [1992] give sufficient conditions for superlinear and quadratic convergence of primal-dual algorithms in terms of the centering parameter and the step size chosen. Among others, Ye, Güler, Tapia, & Zhang [1993] and Mehrotra [1993] describe methods that attain superlinear and quadratic convergence together with polynomial bounds on complexity. El-Bakry, Tapia, & Zhang [1991] compare local convergence strategies and point out that theoretical asymptotic superlinear and quadratic convergence usually translate into fast linear convergence in practice. Typically infeasible interior-point methods detect infeasibility by not being able to converge to optimality (or, as in the case of artificial variables, having one or more not equal to zero even though a high objective

coefficient is attached to them); in fact, guaranteed detection of infeasibility is available theoretically only if the method is started from a very inefficient initial point. In 1992, Ye, Todd, & Mizuno [1994] developed a homogeneous and self-dual formulation that treats optimality and infeasibility more symmetrically for the purpose of being able to detect infeasibility.

Clearly, the implementation of an efficient interior-point algorithm requires an efficient Cholesky factorization that also minimizes the storage requirement for the lower triangular matrix L . Sparsity is maintained in L by first determining an ordering of the rows of AD^2A^T and then using a sparse column Cholesky factorization. Two heuristics for permuting the rows of AD^2A^T to minimize the fill-in in L are the multiple minimum degree ordering (see, for example Liu [1985]); and the minimum local fill-in ordering, (see, for example, Markowitz [1957] and Duff, Erisman, & Reid [1986]). In practice the minimum fill-in local ordering produces a sparser L than the minimum degree ordering but requires more computational time. The Cholesky factors are obtained through the use of a sparse column Cholesky factorization procedure; for details see George & Liu [1981]. If A has relatively few dense columns then the Cholesky factors will also have few dense columns. Techniques for handling such dense columns can be found for general cases in Grcar [1990], for general linear programs in Vanderbei [1991], and for stochastic linear programs in Lustig, Mulvey, and Carpenter [1991].

Eventually the matrices AD^2A^T get very ill-conditioned leading to problems in computing the Cholesky factors. One technique to reduce the effect of ill-conditioning is preprocessing. Another possibility is regularization, which is the perturbing of the linear program problems to make them “easier” to solve. Yet another possibility is refinement, which is iteratively solving modified problems to ensure that the error in the solution process is small. For a discussion of regularization and refinement, see Gill, Murray, Ponceleón, & Saunders [1995].

As noted in this chapter, the use of the QR factorization can improve numerical stability at the expense of computational speed. However, there is still the possibility of a large amount of fill-in (additional nonzeros being created) in R . One possibility is to solve instead a system of equations whose coefficient matrix is:

$$\begin{pmatrix} -D_x^{-2} & (AD_x)^T \\ AD_x & \end{pmatrix}. \quad (4.71)$$

For details on the solution of such systems see Fourer & Mehrotra [1993], Vanderbei [1995, 1992], and Vanderbei & Carpenter [1993]. In the context of finite element analysis, Vavasis [1993, 1994] to ensure stability when solving (4.71) suggests caution be exercised and proposes a stable method. M. Wright [1992], on the other hand, proposes working with an unsymmetric Jacobian matrix, found in primal-dual methods, of the form

$$\begin{pmatrix} D_s & -D_x A^T \\ A & \end{pmatrix}. \quad (4.72)$$

Other options for reducing fill-in are that of incomplete or partial Cholesky factorizations. For a discussion on incomplete factorizations, see for example, Axelsson & Munksgaard [1983], Jones & Plassmann [1995], Meijerink & van der Vorst [1981], Munksgaard [1980], and Thapa [1984b]. Mehrotra [1992b] has experimented with using an incomplete Cholesky factorization for use in a preconditioned conjugate gradient method; however, his results were not very promising. Iterative methods can also be used to solve the Newton equations (4.48), (4.49), and (4.51); see Kim & Nazareth [1994].

For a review of methods with a focus on computational results, see, for example, Gill, Murray, & Saunders [1988] and Lustig, Marsten, & Shanno [1994]. The latter paper reports that test results have demonstrated the following: (i) problems where the sum of the rows and columns is less than 2000 are more likely to be solved faster by the Simplex Method; (ii) for problems where the sum of the number of rows and columns is between 2000 and 10000, simplex codes and interior-point codes compete evenly; and (iii) problems larger than the latter are more likely to be solved faster by interior-point codes. Of course, the structure of the models plays a very important role and even very large problems may very easily be solved faster by very good implementations of the Simplex Method.

The first-order conditions were derived through the use of a Lagrangian, however, we should note that these are simply the Karush-Kuhn-Tucker (KKT) conditions. As noted in Section 4.5, the primal log barrier method uses $D_x = \mu D_s^{-1} e$, the dual log barrier method uses $D_s = \mu D_x^{-1} e$, and the primal-dual log barrier method uses $D_x D_s e = \mu e$, where $e = (1, 1, \dots, 1)^T$. Gill, Murray, Ponceleón, & Saunders [1995], among others, suggests also looking at the fourth possibility of $D_x^{-1} D_s^{-1} = \mu e$. They further suggests research on the development of solvers that would directly solve the resulting Jacobian equations if each of the four options is used in turn in place of (4.46) in the first order conditions. The associated Jacobians are:

$$J_p = \begin{pmatrix} I & \mu D_x^{-2} & \\ I & & A^T \\ & A & \end{pmatrix}, \quad J_d = \begin{pmatrix} \mu D_s^{-2} & I & \\ I & & A^T \\ & A & \end{pmatrix},$$

$$J_{pd} = \begin{pmatrix} D_x & D_s & \\ I & & A^T \\ & A & \end{pmatrix}, \quad J = \begin{pmatrix} D_x^{-1} D_s^{-1} & D_x^{-2} D_s^{-1} & \\ I & & A^T \\ & A & \end{pmatrix}.$$

See Gill, Murray, Ponceleón, & Saunders [1994] for a discussion of these systems and for global convergence proofs and guarantees of nonsingularity in each of the above Jacobians.

Several papers have appeared that suggest the superiority of solving reduced KKT systems because of better sparsity control, more direct handling of free variables, and the application of static reordering schemes for factorization. For example, see Forsgren & Murray [1990], Fourer & Mehrotra [1992], Gill, Murray, Ponceleón, & Saunders [1994], Vanderbei [1995, 1992, 1994], and Vanderbei & Carpenter [1993]. As an example of reduced KKT systems consider the first-order conditions for von Neumann's symmetric primal (2.1) and dual (2.2) systems:

$$\begin{aligned} Ax - w &= b \\ A^T y + s &= c \\ D_x D_s e &= \mu e \\ D_y D_w e &= \mu e \end{aligned}$$

where $e = (1, 1, \dots, 1)^T$,

$$D_x = \text{Diag}(x), \quad D_s = \text{Diag}(s), \quad D_y = \text{Diag}(y), \quad D_w = \text{Diag}(w).$$

Application of Newton's method results in:

$$\begin{pmatrix} D_x & & D_s & & \\ & D_y & & D_w & \\ I & & & & A^T \\ & -I & A & & \end{pmatrix} \begin{pmatrix} \Delta s \\ \Delta w \\ \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} \mu e - D_x D_s e \\ \mu e - D_y D_w e \\ c - A^T y - s \\ b - Ax + w \end{pmatrix}.$$

The first two sets of equations can be used to solve for Δs and Δw respectively, and substituted in the next two sets to give us the reduced KKT system:

$$\begin{pmatrix} -D_s D_x^{-1} & A^T \\ A & D_w D_y^{-1} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} c - A^T y - \mu D_x^{-1} e \\ b - Ax + \mu D_y^{-1} e \end{pmatrix}.$$

As mentioned earlier, it is likely that the best commercial software of the future will be some sort of a hybrid of the Simplex Method and an interior-point method. A problem with interior-point methods has been the inability to quickly obtain a basic feasible solution (or extreme point). Algorithm 4.2 on Page 138 addresses this issue; it, together with Exercise 4.14 on Page 139 is based on Problem 11 on Page 145 of Dantzig [1963]. For additional details on obtaining a basic feasible solution, see Andersen & Ye [1995], Bixby, Gregory, Lustig, Marsten, & Shanno [1992], Bixby & Saltzman [1994], Charnes & Kortanek [1963], Kortanek & Zhu [1988], and Megiddo [1991].

For an introduction to nonlinear programming see, for example, Avriel [1976], Fiacco & McCormick [1968], Gill, Murray, & Wright [1981], Kuhn & Tucker [1950], Wolfe [1967], and Zangwill [1969]. For a thorough analysis of barrier function methods, see M. Wright [1976].

In conclusion, over the years there have been significant improvements in Simplex Method based codes. As a consequence, test results as of 2003 with interior-point codes show superiority only on problems with several thousand rows and columns. Typically, the results are superior on problems where a block-diagonal type structure exists.

4.9 PROBLEMS

4.1 *Based on a problem in Dantzig [1963].* Suppose z_o ; $x_1^o > 0, x_2^o > 0, \dots, x_k^o > 0$ and $x_{k+1}^o = \dots = x_n^o = 0$ constitutes a feasible solution to a linear program in standard form.

- Show that if $k > m$ a new solution can be found with m or fewer variables at positive values.
- Show that this reduction process can take up to $k - m$ steps.
- Show that the reduction can be done in such a way that the objective function is not increasing at each step. Show also that a case can arise in which a class of solutions is generated where $z \rightarrow -\infty$.

4.2 *Hall & Vanderbei [1993].* This problem demonstrates that the dual iterates of the affine scaling algorithm do not converge when $\alpha > 2/3$. Consider the primal problem:

$$\begin{aligned} \text{Minimize} \quad & x_1 + x_2 + x_3 & = z \\ \text{subject to} \quad & x_1 + x_2 + x_3 - x_4 = 0 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0. \end{aligned} \tag{4.73}$$

- Write down the dual of the problem.
- Start with the primal interior point $x = (10, 10, 19, 1)^T$ and use a steplength size of $\alpha = 0.995$ to demonstrate that the primal affine algorithm generates dual iterates that do not converge but generates primal iterates that do converge to the unique primal minimum $x^* = (0, 0, 0, 0)^T$.

-
- (c) Which of the assumptions required in the proof of convergence of Dikin's algorithm is violated?

This page intentionally left blank

DEGENERACY

We have seen that if degeneracy occurs, then it is possible to have a sequence of iterations with no decrease in the value of z . Under such circumstances, it may happen that a basic set will be repeated, thereby initiating an endless cycle of such repetitions. It turns out, for reasons not fully understood, that in practice almost all problems are degenerate and some are highly degenerate, but that in spite of degeneracy, cycling almost never happens. This is why early commercial software packages did not include any degeneracy resolving schemes. When there is degeneracy or “near” degeneracy it tends to slow the solution process, and this has given rise to a number of *anti-cycling* or *degeneracy resolving* schemes that have been very successfully used in commercial software packages to reduce the number of iterations.

5.1 EXAMPLES OF CYCLING

Example 5.1 (Hoffman) In 1951, A. J. Hoffman constructed an ingenious example to show that cycling can occur under degeneracy; it involves three equations and eleven variables; see Table 5-1 and Table 5-2. He showed, in the case of degeneracy, that if one *resolved* the ambiguity of choice regarding which variable to drop from the basic set by the rule of *selecting the first among them*, then the tableau at iteration 10 would turn out to be the same as that at iteration 0. Notice in Tables 5-1 and 5-2 that column 1, associated with the relation $x_1 = 1$, remains in the basis for all iterations. Next notice that the tableau for iteration 2 is exactly the same as that of iteration 0 if we relabel the indices $(2, 3, 4, \dots, 11)$ of iteration 0 as $(4, 5, 6, \dots, 11; 2, 3)$. Hence, eight more iterations will repeat iteration 0. It follows, in this case, using the first choice rule, that the same basic set would be repeated every ten iterations and the Simplex Algorithm would cycle forever without converging to an optimal solution.

x_1	x_2	x_3	x_4	x_5	x_6
-------	-------	-------	-------	-------	-------

Iteration 0

			$-(1 - \cos \theta) / \cos \theta$	w	
1	1	1	$\boxed{\cos \theta}$ $\sin \theta \tan \theta / w$	$-w \cos \theta$ $\cos \theta$	$\cos 2\theta$ $\tan \theta \sin 2\theta / w$

Iteration 1

	$(1 - \cos \theta) / \cos^2 \theta$			$w(2 \cos \theta - 1) / \cos \theta$	$-2 \sin \theta \tan \theta$
1	$\sec \theta$ $-\tan^2 \theta / w$	1	1	$-w$ $\boxed{\sec \theta}$	$4 \cos^2 \theta - 3$ $\tan^2 \theta / w$

Iteration 2

	$4 \sin^2 \theta$	$w(1 - 2 \cos \theta)$			$-(1 - \cos \theta) / \cos \theta$
1	$\cos \theta$ $-\tan \theta \sin \theta / w$	$w \cos \theta$ $\cos \theta$	1	1	$\boxed{\cos \theta}$ $\sin \theta \tan \theta / w$

Where $\theta = 2\pi/5$, $w > (1 - \cos \theta) / (1 - 2 \cos \theta)$

Table 5-1: Hoffman’s Example of Cycling (Continued on the Right)

▷ **Exercise 5.1** The purpose of this exercise is to demonstrate how many of the relations in Hoffman’s examples are determined assuming that $\theta = 2\pi/5$.

1. Show that $\cos 2\theta = \cos 3\theta$.
2. Show that $\sin 2\theta = -\sin 3\theta$.
3. Use (1) to show that $\cos 2\theta + \cos \theta = \cos \theta \cos 2\theta$.
4. Use (1) and (3) to show that on iteration 1 the objective coefficient for x_6 is $-2 \sin \theta \tan \theta$.
5. Use (1) to show that the coefficient $\bar{a}_{26} = 4 \cos^2 \theta - 3$.

Example 5.2 (Beale’s Three Equation, Seven Variable Example) In 1955, E. M. L. Beale constructed a second example, a version of which is shown in Table 5-3, which is remarkable for its simplicity. It also has three equations but only seven variables. Using the same rule for resolving a tie, the tableau at iteration 6 is the same as that at iteration 0; it has the same basic variables in the same order. It is conjectured that this is the simplest not totally degenerate example of cycling; i.e., none can be constructed with fewer variables regardless of the number of equations.

Example 5.3 (Tucker’s Totally Degenerate Example) The following example, due to Tucker, is said to be the simplest of all examples constructed so far. However, this

x7	x8	x9	x10	x11	RHS
Iteration 0					
2w	4 sin ² θ	-2w cos 2θ	4 sin ² θ	w(1 - 2 cos θ)	z(min)
-2w cos ² θ	cos 2θ	2w cos ² θ	cos θ	w cos θ	1
cos 2θ	-2 sin ² θ/w	cos 2θ	-tan θ sin θ/w	cos θ	0
Iteration 1					
2w cos θ	(cos θ - 1)/cos θ	3w	2 sin θ tan θ	-w(4 cos ² θ - 3)	z
-2w cos θ	4 cos ² θ - 3	2w cos θ	1	w	1
1	2 sin θ tan θ/w	4 cos ² θ - 3	-2 sin θ tan θ/w	4 cos ² θ - 3	0
Iteration 2					
+w		2w	4 sin ² θ	-2w cos 2θ	z
-w cos θ	cos 2θ	-2w cos ² θ	cos 2θ	2w cos ² θ	1
cos θ	tan θ sin 2θ/w	cos 2θ	-2 sin ² θ/w	cos 2θ	0

Where $\theta = 2\pi/5$, $w > (1 - \cos \theta)/(1 - 2 \cos \theta)$

Table 5-2: Hoffman’s Example of Cycling (Continued from the Left)

example has a totally degenerate solution (see Exercise 5.4).

$$\begin{aligned}
 &\text{Minimize} && -2x_1 - 3x_2 + x_3 + 12x_4 = z \\
 &\text{subject to} && -2x_1 - 9x_2 + x_3 + 9x_4 \leq 0 \\
 &&& \frac{1}{3}x_1 + x_2 - \frac{1}{3}x_3 - 2x_4 \leq 0 \\
 &&& \text{and } x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0.
 \end{aligned} \tag{5.1}$$

On adding slack variables $x_5 \geq 0$ and $x_6 \geq 0$, a feasible solution is readily available as the slacks $(x_5, x_6) = (0, 0)$. Choose the initial basic variables as the slacks x_5, x_6 .

1. Assume the index s of the incoming variable is chosen as the nonbasic variable with the most negative reduced cost.
2. Assume the index j_r of the outgoing variable is determined by looking at all $\bar{a}_{is} > 0$ as specified in the Simplex Algorithm and choosing $r = i$ as the smallest index such that $\bar{a}_{is} > 0$.

Then the basic sets of indices generated at each iteration for the first six iterations are: $\{5, 2\}$, $\{1, 2\}$, $\{1, 4\}$, $\{3, 4\}$, $\{3, 6\}$, and $\{5, 6\}$, respectively. Observe that the basis repeats on the sixth iteration, leading to an endless cycle of iterations.

▷ **Exercise 5.2** Apply the tableau form of the Simplex Algorithm to Tucker’s example, the linear program (5.1), to show that it cycles endlessly.

Basics	$-z$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS
--------	------	-------	-------	-------	-------	-------	-------	-------	-----

Iteration 0

$-z$	1	$-3/4$	150	$-1/50$	6				0
x_5		$1/4$	-60	$-1/25$	9	1			0
x_6		$1/2$	-90	$-1/50$	3		1		0
x_7				1				1	1

Iteration 1

$-z$	1		-30	$-7/50$	33	3			0
x_1		1	-240	$-4/25$	36	4			0
x_6			30	$3/50$	-15	-2	1		0
x_7				1				1	1

Iteration 2

$-z$	1			$-2/25$	18	1	1		0
x_1		1		$8/25$	-84	-12	8		0
x_2			1	$1/500$	$-1/2$	$-1/15$	$1/30$		0
x_7				1				1	1

Iteration 3

$-z$	1	$1/4$			-3	-2	3		0
x_3		$25/8$		1	$-525/2$	$-75/2$	25		0
x_2		$-1/160$	1		$1/40$	$1/120$	$-1/60$		0
x_7		$-25/8$			$525/2$	$75/2$	-25	1	1

Iteration 4

$-z$	1	$-1/2$	120			-1	1		0
x_3		$-125/2$	10,500	1		50	-150		0
x_4		$-1/4$	40		1	$1/3$	$-2/3$		0
x_7		$125/2$	-10,500			-50	150	1	1

Iteration 5

$-z$	1	$-7/4$	330	$1/50$			-2		0
x_5		$-5/4$	210	$1/50$		1	-3		0
x_4		$1/6$	-30	$-1/150$	1		$1/3$		0
x_7				1				1	1

Table 5-3: Beale's Example of Cycling

- ▷ **Exercise 5.3** The software DTZG Simplex Primal (*Linear Programming 1*) does not use any anti-cycling procedures. Run it on Tucker's example, the linear program (5.1), to verify that it also cycles endlessly.
- ▷ **Exercise 5.4** Show by introducing slack variables x_5 and x_6 , and an equation $x_7 = 1$ that this becomes a not totally degenerate example similar to that of Beale in size, degree, and number of iterations before cycling. Show that by deleting the last row and last column in Beale's example it becomes a totally degenerate example, similar to Tucker's example, that cycles endlessly.

5.2 ON RESOLVING DEGENERACY

Since cycling in the Simplex Algorithm is only possible under degeneracy, it is pertinent to ask how degeneracy can occur, how frequently it is encountered in practice, and how often it implies cycling. Degenerate solutions are possible only when the constants, b_i of the original right-hand side bear a special relation to the coefficients of the basic variables. This is clear because the process of reduction to one of the finite set of canonical forms depends only on the coefficients and *not* on the right-hand side; the final values \bar{b}_i are weighted sums of the original b_i s where the weights depend only on the coefficients in the basis. If all the b_i s were selected at random, it would be something of a miracle if one or more of the constants \bar{b}_i of the canonical system should vanish.

Nevertheless, *it is common experience*, based on the solutions of thousands of practical linear programming problems by the Simplex Method, that nearly every problem at some stage of the process is degenerate. It might be thought that, since degeneracy happens all the time, there should be many observed cases of cycling. However, to date, there have been very few known cases of cycling other than the specially concocted examples of Hoffman, Beale, and others. Apparently, cycling is a very rare phenomenon in practice. For this reason, most software for computers until the 1980s did not include special code for resolving (avoiding) degeneracy in order to prevent the possibility of cycling. Since the 1980s anti-cycling schemes have been used in some commercial software not only to prevent cycling but, more importantly, to reduce the number of iterations in degenerate and "near"-degenerate cases.

From a mathematical point of view, the phenomenon of cycling is fascinating. Long before Hoffman discovered his example, it was conjectured that cycling could happen and simple devices were proposed to avoid degeneracy and thus avoid the possibility of cycling. The goal of the early efforts was to devise a way of avoiding degeneracy that involved *as little extra work as possible*. The first proposal along these lines was by Dantzig and his student Edmondson; it involved perturbing the right-hand sides in such a way that on each iteration:

1. the basic feasible solution of the perturbed problem is nondegenerate, and

2. the basic solution for the corresponding unperturbed problem is feasible.

Another early proposal involving perturbation is due to Charnes; see Problems 5.9, 5.11, and 5.12 for methods that perturb the right-hand side.

The perturbation may be viewed as a rule that guides the proper choice of variables to drop from the basic set in case of ties. Many other rules have been developed over the years. Five well-known ones will now be described.

5.3 DANTZIG'S INDUCTIVE METHOD

We first present the inductive method that was initially thought to be difficult to implement because it appears to require a complicated bookkeeping scheme to keep track of a hierarchy of subproblems. Philip Wolfe, however, discovered a very simple rule, which we will describe in Section 5.4, that requires the updating of only one extra column of indices d that keeps track of this hierarchy.

The induction will be on the number of rows m . To initiate the induction, we begin with a one equation ($m = 1$) linear program in canonical form:

$$\begin{aligned} -z \quad & + \sum_{j \in \mathcal{N}} \bar{c}_j x_j = -\bar{z}_t \\ x_{j_1} + \sum_{j \in \mathcal{N}} \bar{a}_{1j} x_j & = \bar{b}_1, \quad x_j \geq 0, \quad j = 1, \dots, n. \end{aligned} \tag{5.2}$$

If $\bar{b}_1 > 0$ (i.e., the problem is nondegenerate) then after a pivot the updated $\bar{b}_1 > 0$ because $\bar{b}_1 = \bar{b}_1 / \bar{a}_{i_s}$ and $\bar{a}_{i_s} > 0$. It follows for $m = 1$ that the Simplex Algorithm with a positive right-hand side will terminate in a finite number of iterations.

If, on the other hand, the one-equation ($m = 1$) problem is degenerate, i.e., $\bar{b}_1 = 0$, create an auxiliary problem that is the same as problem (5.2) in every respect except that $\bar{b}_1 = 1$. *Perform identical pivot steps on both problems.* Clearly, both remain basic feasible; the only difference is that the updated $\bar{b}_1 = 0$ for the original unperturbed problem. Furthermore, both terminate at the same time because the termination condition depends on the left-hand side of the problems only. Since the auxiliary problem is nondegenerate, the Simplex Algorithm will terminate after a finite number of steps, and thus the original problem will also be solved in a finite number of steps.

- ▷ **Exercise 5.5** Show that if an LP in standard form is totally degenerate, i.e., $b_i = 0$ for all i , it will remain so after pivoting. Also show that if the LP is not totally degenerate, i.e., $b \neq 0$, then after pivoting updated $b \neq 0$.

At this point we have shown that we can solve an $m = 1$ problem in a finite number of pivot steps. Next we shall demonstrate an inductive proof of finiteness of the Simplex Algorithm using the ideas that we have just discussed.

THEOREM 5.1 (Finiteness of the Inductive Method) *If for all $k < m$ there exist rules \mathcal{A}_k for choosing to pivot in the Simplex Algorithm that solves a k -equation linear program in a finite number of pivot steps, then for $k = m$ there exists a rule \mathcal{A}_m for choosing to pivot in the Simplex Algorithm which solves an m -equation linear program in a finite number of pivot steps, terminating with $\bar{c} \geq 0$ or, for some s , $\bar{c}_s < 0$ and $\bar{A}_{\bullet s} \leq 0$.*

Proof. We are given algorithms \mathcal{A}_k for $k = 1, \dots, m - 1$ that solve any linear program with $k < m$ rows in a finite number of steps. We wish to use these algorithms to generate an algorithm \mathcal{A}_m for solving any m -row linear program in a finite number of feasible pivot steps.

Consider first an m -equation linear program in canonical form:

$$\begin{aligned} -z + \sum_{j \in \mathcal{N}} \bar{c}_j x_j &= -\bar{z}_t, \\ x_{j_i} + \sum_{j \in \mathcal{N}} \bar{a}_{ij} x_j &= 0 \quad \text{for } i = 1, \dots, k, \\ x_{j_i} + \sum_{j \in \mathcal{N}} \bar{a}_{ij} x_j &= \bar{b}_i, \quad \bar{b}_i > 0 \quad \text{for } i = k + 1, \dots, m. \end{aligned} \tag{5.3}$$

Call the *restricted linear program* \mathcal{R}_k the one obtained by setting aside for the moment the $m - k$ rows of (5.3) in which $\bar{b}_i > 0$.

Case 1: Suppose $k < m$. Since \mathcal{R}_k has $k < m$ rows, we have an algorithm \mathcal{A}_k that makes a finite sequence \mathcal{P} of row-column choices of pivots that terminates with either $\bar{c} \geq 0$ or, for some s , $\bar{c}_s < 0$ and $\bar{a}_{is} \leq 0$ for i restricted to the rows of \mathcal{R}_k .

Simultaneously apply these same pivot choices \mathcal{P} (restricted to the rows of \mathcal{R}_k) to (5.3), except when we pivot we do the elimination on all the rows of (5.3). Note this elimination does not affect feasibility because \bar{b}_r of the pivot row is 0 and hence there is no change to the value of \bar{b}_i . Although the updated \bar{c} changes from iteration to iteration, the change in \bar{c} of the restricted problem \mathcal{R}_k is the same as that for (5.3).

If \mathcal{A}_k results in $\bar{c} \geq 0$ on some iteration then we have a finite sequence of feasible steps that solves (5.3). On the other hand, if \mathcal{A}_k results, on some iteration, with updated $\bar{c}_s < 0$ and $\bar{a}_{is} \leq 0$ for i restricted to the rows of \mathcal{R}_k , then choose for pivot on the full system $\bar{a}_{rs} > 0$ for a row r among the rows not in \mathcal{R}_k in the usual manner. If no such r exists then we can construct a class of solutions such that $z \rightarrow -\infty$. Otherwise we pivot on \bar{a}_{rs} on some row where $\bar{b}_r > 0$, and this results in a strict decrease of the objective function. After the pivot update we repeat this process generating a new restricted problem. We know that there can only be a finite number of strict decreases of the objective function z (because there is only a finite number of distinct canonical forms). We call the set of pivots used to solve (5.3), Algorithm \mathcal{A}_m .

Case 2: Suppose that the $k = m$ problem is totally degenerate. Create an auxiliary problem by setting the right-hand sides $\bar{b}_i > 0$, for example $\bar{b} = e = (1, 1, \dots, 1)^T$. This auxiliary problem can be solved by Algorithm \mathcal{A}_m of Case 1. The latter will also terminate in a finite number of pivot steps with either $\bar{c} \geq 0$ (in which case we terminate the corresponding pivoting on the original system with an optimal solution) or a situation where $\bar{c}_s < 0$ and $\bar{A}_{\bullet s} \leq 0$ and we have an unbounded solution for the auxiliary problem. If this is the situation then we solve the corresponding totally degenerate problem by applying the same pivot steps that solved the auxiliary problem and generate the same unbounded class of solutions for the original system.

This completes the proof. ■

▷ **Exercise 5.6** Show, except for the updating of the right-hand side of \mathcal{R} , that no extra work is involved to maintain the restricted problems.

5.4 WOLFE'S RULE

Except for the work of inductively generating the where to pivot rules \mathcal{A}_k , the inductive method requires no extra work. It was originally thought that the method was impractical because of the work involved to generate the rules \mathcal{A}_k . Wolfe observed that if an auxiliary problem \mathcal{R}_1 is degenerate its auxiliary problem \mathcal{R}_2 is a subset of \mathcal{R}_1 and therefore the set of auxiliary problems at any point in the algorithm form a hierarchy that can be kept track of and updated by a single m vector of integers. We first solve a simple example using the inductive method and Wolfe's Rule.

Example 5.4 (Inductive Method and Wolfe's Rule) We illustrate the Inductive Method and Wolfe's Rule by applying it to Beale's example of cycling. In Table 5-4, the column \hat{b} is the modified right-hand side for the restricted problem and column d keeps track of the hierarchy of restricted problems. Iteration 0 has the first two right-hand sides equal to 0; as a result $d_1 = d_2 = 1$, $d_3 = 0$, and $\hat{b}_1 = \hat{b}_2 = 1$, implying that the first restricted problem \mathcal{R}_1 consists of the rows 1 and 2. The pivot is next chosen from among the restricted problem rows 1 and 2 using the right hand side \hat{b} . Using the usual rules, the pivot is on column 1 and row 2, shown in boldface. Next the pivot is performed on the entire problem using the original right-hand side. In addition, the right-hand side \hat{b} is updated for the restricted problem \mathcal{R}_1 only. No new \hat{b}_i become 0 and the pivoting process continues using \mathcal{R}_1 . This time, the incoming column is x_3 which has all negative coefficients \bar{a}_{i3} in \mathcal{R}_1 . Hence, we look at the remaining rows and attempt to find a pivot. A pivot is found in row 3; on pivoting, on the entire problem it turns out that an optimal solution is found. This process is formalized in Algorithm 5.1.

Basics	$-z$	x_1	x_2	x_3	x_4	x_5	x_6	x_7	RHS	\hat{b}	d
Iteration 0: $d_i = 1$ for \mathcal{R}_1											
$-z$	1	$-3/4$	150	$-1/50$	6				0		
x_5		$1/4$	-60	$-1/25$	9	1			0	1	1
x_6		$1/2$	-90	$-1/50$	3		1		0	1	1
x_7				1				1	1		0
Iteration 1											
$-z$	1		15	$-1/20$	$21/2$		$3/2$		0		
x_5			-15	$-3/100$	$15/2$	1	$1/2$		0	$-1/2$	1
x_1		1	-180	$-2/50$	6		2		0	2	1
x_7				1				1	1		0
Iteration 2: Optimal											
$-z$	1		15		$21/2$		$3/2$	$1/20$	$1/20$		
x_5			-15		$15/2$	1	$1/2$	$3/100$	$3/100$	$-1/2$	1
x_1		1	-180		6		2	$2/50$	$2/50$	2	1
x_3				1					1		0

Table 5-4: Inductive Method and Wolfe's Rule Applied to Beale's Example

Algorithm 5.1 (Wolfe's Rule for Selecting a Pivot) Consider a linear program in feasible canonical form:

$$\begin{aligned}
 -z &+ \bar{c}^T x_N = -z_t \\
 Ix_B + \bar{A}x_N &= \bar{b},
 \end{aligned}$$

where x_B are the basic variables and x_N are the nonbasic variables. Let d be a m -vector that keeps track of the hierarchy of restricted problems. For example, suppose the α -component of d is $d_\alpha = 3$. This states that equation α belongs to a restricted problem $\mathcal{R}_3 = \{i \mid d_i \geq 3\}$, which in turn is a subset of a restricted problem $\mathcal{R}_2 = \{i \mid d_i \geq 2\}$, and so on. Perform the following steps:

1. Initialize $d = 0$.
2. For all i , if $\bar{b}_i = 0$ then set $d_i \leftarrow d_i + 1$ and $\bar{b}_i = 1$.
3. Let \mathcal{D} be the restricted subset $\{i \mid d_i = \max_k d_k\}$.
4. Find

$$s = \underset{j}{\operatorname{argmin}} \bar{c}_j, \tag{5.4}$$

where s is the index j (argument) where \bar{c}_j attains a minimum, that is,

$$\bar{c}_s = \min_j \bar{c}_j. \tag{5.5}$$

5. *Test for Optimality.* If $\bar{c}_s \geq 0$, set $\bar{b}_i = 0$ for $d_i > 0$ and report the basic feasible solution as optimal and *stop*.

6. If $\bar{c}_s < 0$, then s is the index of the incoming basic variable.
7. *Test for Unbounded z .* If $\bar{c}_s < 0$ and $\bar{A}_{\mathcal{D}s} \leq 0$ then let

$$d^* = \max_i \begin{cases} d_i & \text{if } \bar{a}_{is} > 0; \\ -1 & \text{otherwise.} \end{cases}$$

- (a) If $d^* = -1$ then $\bar{A}_{\bullet s} \leq 0$. Set $\bar{b}_i = 0$ for all i such that $d_i > 0$ and terminate with solutions such that $z \rightarrow -\infty$.
 - (b) If $d^* \geq 0$ then set $\bar{b}_i = 0$ and $d_i = d^*$ for all i such that $d_i > d^*$. Next redefine $\mathcal{D} = \{i \mid d_i = \max_k d_k\}$.
8. Use the usual rules for selecting the pivot row r restricted to rows in \mathcal{D} . Do a full pivot on all rows i , for $i = 1, \dots, m$ and the objective function, except do not modify \bar{b}_i for $i \notin \mathcal{D}$ or the value of z_t on the objective row.
 9. Go to Step 2.

▷ **Exercise 5.7** How is Wolfe's Rule related to the inductive method?

5.5 BLAND'S RULE

Except possibly for the Random Choice Rule, Robert Bland's Rule is the simplest to implement.

Rule 5.2 (Bland) Whenever the regular choice for selecting the pivot in the Simplex Method would result in a 0 change of objective value of the basic solution then, instead of the regular choice, do the following:

1. *Incoming Column.* Choose the pivot column $j = s$ with relative cost $\bar{c}_j < 0$ having the smallest index j .
2. *Outgoing Column.* Choose the outgoing basic column j_r among those eligible for dropping with the smallest index j_i .

THEOREM 5.2 (Finite Termination Using Bland's Rule) *Cycling is impossible using Bland's Rule.*

Proof. We will prove the theorem for a linear program in standard form $\min c^T x$ subject to $Ax = b$, $x \geq 0$, where A is an $\bar{m} \times \bar{n}$ matrix.

Assume on the contrary that applying Bland's Rule for some LP results after a number of iterations in a repeat of an earlier canonical form and hence would cycle thereafter.

Some columns may have remained nonbasic throughout the entire cycle; we drop them from the LP as dropping them will not result in a different choice of incoming column when applying Bland's Rule. It may also happen that there may be some basic column with index j_i , which remained in as the i th basic column throughout the cycle; we drop all such basic columns with indices j_i and their corresponding

i th equations as this will not affect the choice of outgoing column. Maintaining the original ordering of the columns and rows, relabel the indices of the columns from $j = 1, \dots, n$ and the indices of the equations from $i = 1, \dots, m$. After these deletions, each column of the adjusted LP has the property that on at least one iteration of the cycle it is basic and at another iteration it is nonbasic.

Therefore, on some iteration t_o , column $A_{\bullet n}$ is basic as the r th basic column, that is, $j_r = n$, and is replaced on the next iteration by some other column s . We denote by \mathcal{C}_o the canonical form of iteration t_o and its basic indices by $j_1, \dots, j_r, \dots, j_m$ where $j_r = n$ is the outgoing column. Clearly

$$\bar{c}_s < 0, \quad s < n, \quad (5.6)$$

and, because $n > j_i$ for all $i \neq r$, we have by Bland's outgoing basic column rule,

$$\begin{aligned} \bar{a}_{rs} &> 0, & \text{where } j_r = n, \\ \bar{a}_{is} &\leq 0, & \text{for all } i \neq r. \end{aligned} \quad (5.7)$$

According to our contrary assumption, on some future iteration t_1 , the column $A_{\bullet n}$ is nonbasic and is the candidate for reentering the basic set, which by Bland's incoming column rule, can only happen if its reduced cost is negative while all other reduced costs are nonnegative, i.e.,

$$\begin{aligned} \hat{c}_n &< 0 \quad \text{and} \\ \hat{c}_j &\geq 0 \quad \text{for all } j < n. \end{aligned} \quad (5.8)$$

Now the canonical system \mathcal{C}_1 at t_1 was obtained from the canonical system \mathcal{C}_o at t_o by a sequence of pivoting. Therefore the objective coefficients $(\hat{c}_1, \hat{c}_2, \dots, \hat{c}_n)$ of \mathcal{C}_1 can be obtained by multiplying the rows of \mathcal{C}_o by the simplex multipliers $(\hat{\pi}_1, \hat{\pi}_2, \dots, \hat{\pi}_m)$, summing, and subtracting from the objective row $(\bar{c}_1, \bar{c}_2, \dots, \bar{c}_n)$ of \mathcal{C}_o . These simplex multipliers are unique and are obviously

$$(\hat{\pi}_1, \hat{\pi}_2, \dots, \hat{\pi}_m) = (-\hat{c}_{j_1}, -\hat{c}_{j_2}, \dots, -\hat{c}_{j_m})$$

because the corresponding columns of \mathcal{C}_o are basic and its columns are the identity in the canonical form. Hence, in particular, we can apply the values of $\hat{\pi}$ to column s of \mathcal{C}_o to obtain the reduced cost \hat{c}_s of column s in \mathcal{C}_1 as:

$$\hat{c}_s = \bar{c}_s - \sum_{i=1}^m \hat{\pi}_i \bar{a}_{is} = \bar{c}_s + \sum_{i=1}^m \hat{c}_{j_i} \bar{a}_{is}.$$

Rearranging we obtain

$$\sum_{i=1}^m \hat{c}_{j_i} \bar{a}_{is} = -\bar{c}_s + \hat{c}_s. \quad (5.9)$$

Because $\bar{c}_s < 0$ by (5.6) and $\hat{c}_s \geq 0$ by (5.8) since $s < n$, the right-hand side of (5.9) is positive:

$$-\bar{c}_s + \hat{c}_s > 0.$$

On the other hand, by (5.7)

$$\bar{a}_{rs} > 0, \quad \bar{a}_{is} \leq 0,$$

and by (5.8)

$$\hat{c}_{j_r} = \hat{c}_n < 0, \quad \hat{c}_{j_i} \geq 0,$$

implying that the left-hand side of (5.9) is negative:

$$\sum_{i=1}^m \hat{c}_{j_i} \bar{a}_{is} < 0,$$

which is a contradiction. Hence we conclude that our assumption that cycling can occur using Bland's Rule is false. ■

Comment: Bland's Rule is clearly very easy to implement and, as we have just shown, guaranteed to prevent cycling. Its chief drawback is that its choice of incoming column may not be a very good one. Another reason is that it does not appear to be amenable to partial pivoting schemes that will be discussed *Linear Programming 4: Implementation*.

▷ **Exercise 5.8** State Bland's Rule in the context of a linear program with upper and lower bounds on the variables.

▷ **Exercise 5.9** State Bland's Rule in the context of an inequality constrained linear program.

$$\begin{aligned} \text{Minimize} \quad & c^T x = z \\ \text{subject to} \quad & Ax \geq b, \quad A: m \times n. \end{aligned} \tag{5.10}$$

5.6 KRISHNA'S EXTRA COLUMN RULE

The canonical system on iteration t has an extra degeneracy resolving column $\begin{pmatrix} -\gamma_t \\ \beta_t \end{pmatrix}$ "tacked" onto its right-hand side:

$$\begin{aligned} (-z) \quad & + \bar{c}_{N^t}^T x_{N^t} = -\bar{z}_t, \quad -\gamma_t \\ Ix_{B^t} + \bar{A}^t x_{N^t} & = \bar{b}^t, \quad \beta^t \end{aligned} \tag{5.11}$$

The extra column has the property that if it replaced the right-hand side $\begin{pmatrix} -\bar{z}_t \\ \bar{b}^t \end{pmatrix}$ it would never have a β^t with a zero component. Therefore, γ_t is strictly decreasing on each iteration, implying there can be no repeat of a canonical form.

Let s , the incoming column, be chosen by some rule such as $s = \operatorname{argmin}_j \bar{c}_j$, with $\bar{c}_j < 0$. Let \mathcal{R}^t be the set of rows k on iteration t that are tied for pivot choices:

$$\mathcal{R}^t = \left\{ k \mid \frac{\bar{b}_k^t}{\bar{a}_{ks}^t} \leq \frac{\bar{b}_i^t}{\bar{a}_{is}^t}, \bar{b}_i^t \geq 0, \bar{a}_{is}^t > 0, \bar{a}_{ks}^t > 0, i = 1, \dots, m \right\}. \quad (5.12)$$

Then the outgoing column j_r is chosen by:

$$r = \operatorname{argmin}_{i \in \mathcal{R}^t} \left(\frac{\beta_i^t}{\bar{a}_{is}^t} \right). \quad (5.13)$$

Pivoting is then done (as before) using $\bar{a}_{rs}^t x_s$ as the pivot term. The extra column $\begin{pmatrix} -\gamma^t \\ \beta^t \end{pmatrix}$ is updated the same way that any other column j or the right hand side is updated. To initiate, the canonical form for $t = 0$ is partitioned into basic and nonbasic parts with an extra column $\begin{pmatrix} -\gamma_o \\ \beta^o \end{pmatrix}$ tacked on:

$$\begin{aligned} (-z) \quad & + \bar{c}_{N^o}^o x_{N^o} = -\bar{z}_o, \quad -\gamma_o \\ I_{x_{B^o}} + \bar{A}^o x_{N^o} & = \bar{b}^o, \quad \beta^o, \quad \bar{b}^o \geq 0, \quad \beta^o > 0; \end{aligned} \quad (5.14)$$

where γ_o and $\beta^o = (\beta_1^o, \dots, \beta_m^o)^T$ are defined by Krishna's Rule

$$\begin{aligned} -\gamma_o &= 0, \\ \beta_1^o &= \operatorname{FRAC}(\pi) = \operatorname{FRAC}(3.1415926\dots) = .1415926\dots, \\ \beta_i^o &= \operatorname{FRAC}(\beta_{i-1}^o \pi), \quad \text{for } i = 2, \dots, m, \end{aligned} \quad (5.15)$$

where $\operatorname{FRAC}(\alpha)$ is defined to be the fractional part of α . Clearly Krishna's rule can never be implemented because it requires an infinite number of digits of π . In practice one could make do with the accuracy of the machine (without the theoretical guarantee of no cycling).

▷ **Exercise 5.10** Prove that $\beta_i^o > 0$ for $i = 1, \dots, m$.

Krishna's Rule is based on the fundamental property of transcendental numbers, such as π , that it can never be the root of a polynomial equation with all rational coefficients; more precisely,

$$\alpha_0 + \alpha_1 \pi + \alpha_2 \pi^2 + \dots + \alpha_m \pi^m \neq 0 \quad (5.16)$$

whatever be the choice of rational α_i not all zero. It follows (see Exercise 5.11), that β_i^o defined by (5.15) has the property

$$\bar{\alpha}_1 \beta_1^o + \bar{\alpha}_2 \beta_2^o + \dots + \bar{\alpha}_m \beta_m^o \neq 0 \quad (5.17)$$

whatever be the choice of rational $\bar{\alpha}_i$ not all zero.

▷ **Exercise 5.11** Prove that (5.17) follows from (5.16) and the definition of β_i^o given in (5.15). Prove that $\beta_i^o > 0$ for $i = 1, \dots, m$.

LEMMA 5.3 (Properties of the Extra Column Rule) *Given that all coefficients \bar{a}_{ij}^o are rational, r is the row chosen for pivot by (5.13) and \mathcal{R}^t is the set of rows i on iteration t that are tied for pivot by the ratio $\bar{b}_i^t/\bar{a}_{is}^t$ criterion given in (5.12). The extra column β rule has the following properties for every iteration t :*

Property (1): $\beta_i^t \neq 0$, for $i = 1, \dots, m$;

Property (2): $\beta_r/\bar{a}_{r,s}^t < \beta_i/\bar{a}_{is}^t$ for all $i \in \mathcal{R}^t, i \neq r$;

Property (3): If $\bar{b}_i^t = 0$, then $\beta_i^t > 0$;

Property (4): If $\bar{z}_{t+1} = \bar{z}_t$, then $\gamma_{t+1} < \gamma_t$.

Proof. To prove Property (1) note that the canonical form $(Ix_{B^t} + \bar{A}^t x_{N^t} = \bar{b}^t, \beta^t)$ for iteration t can be generated from $(A^o x = Ix_{B^o} + \bar{A}^o x_{N^o} = b^o, \beta^o)$ by multiplying the latter by a nonsingular matrix $M = [A_{B^t}^o]^{-1}$ and reordering the columns:

$$MA^o = A^t, Mb^o = \bar{b}^t, M\beta^o = \beta^t, \tag{5.18}$$

because $MA_{B^t}^o = A_{B^t}^t = I$ (the identity). Moreover since $A_{B^t}^o$ is a nonsingular matrix with all elements rational, M is nonsingular with all elements rational. It follows from (5.17) and (5.18) that

$$\beta_i^t = M_{i1}\beta_1^o + M_{i2}\beta_2^o + \dots + M_{im}\beta_m^o \neq 0 \tag{5.19}$$

because $\bar{a}_j = M_{ij}$ are rational and at least one $M_{ij} \neq 0$ in row i of M (since, if row i were all zero, M would be a singular matrix). This proves Property (1).

To prove Property (2), note by definition of \mathcal{R}^t that $\bar{a}_{js}^t > 0$ and $\bar{a}_{ks}^t > 0$ for $j, k \in \mathcal{R}^t$. We have from Property (1) that

$$\begin{aligned} \beta_j^t &= M_{j1}\beta_1^o + M_{j2}\beta_2^o + \dots + M_{jm}\beta_m^o \neq 0 \\ \beta_k^t &= M_{k1}\beta_1^o + M_{k2}\beta_2^o + \dots + M_{km}\beta_m^o \neq 0 \end{aligned}$$

and that

$$\frac{\beta_j^t}{\bar{a}_{js}^t} - \frac{\beta_k^t}{\bar{a}_{ks}^t} \neq 0,$$

because

$$\frac{M_{ji}}{\bar{a}_{js}^t} - \frac{M_{ki}}{\bar{a}_{ks}^t} \neq 0,$$

are rational and not zero for at least one i because no two columns of M are proportional. Therefore one of the ratios must be the smallest and $j = r$ by (5.13) is chosen by definition as the smallest.

The proof that Property (3) holds is inductive. To initiate the inductive step, note that for $t = 0$, Property (3) obviously holds since $\beta^0 > 0$ by construction, see Exercise 5.11. We assume that the relations in Property (3) are true up to some t , and we will prove that they hold for $t + 1$.

On iteration t consider the following six terms of the canonical form:

$$\begin{array}{ccc} \bar{a}_{rs}^t & \bar{b}_r^t & \beta_r^t \\ \bar{a}_{is}^t & \bar{b}_i^t & \beta_i^t \end{array} \quad (5.20)$$

where r is the pivot row for generating iteration $t + 1$ from t and $i \neq r$. After the pivot on \bar{a}_{rs} , the corresponding right-hand-side terms for $t + 1$ are:

$$\begin{array}{ccc} \bar{b}_r^{t+1} = \frac{\bar{b}_r^t}{\bar{a}_{rs}^t} & \beta_r^{t+1} = \frac{\beta_r^t}{\bar{a}_{rs}^t} \\ \bar{b}_i^{t+1} = \bar{b}_i^t - \frac{\bar{b}_r^t \bar{a}_{is}^t}{\bar{a}_{rs}^t} & \beta_i^{t+1} = \beta_i^t - \frac{\beta_r^t \bar{a}_{is}^t}{\bar{a}_{rs}^t}. \end{array} \quad (5.21)$$

We assume inductively, if $\bar{b}_r^t = 0$ then $\beta_r^t > 0$. It is easy to see that $\beta_r^{t+1} > 0$ when $\bar{b}_r^{t+1} = 0$ because $\bar{b}_r^{t+1} = 0$ implies $\bar{b}_r^t = 0$ since $\bar{a}_{rs}^t > 0$, which in turn implies, by induction, $\beta_r^t > 0$ and $\beta_r^{t+1} > 0$ easily follows. Therefore Property (3) holds for $i = r$. We wish now to show Property (3) for $i \neq r$.

If $\bar{a}_{is}^t < 0$ and $\bar{b}_i^{t+1} = 0$, the latter can only happen if $\bar{b}_i^t = 0$ and $\bar{b}_r^t = 0$ since \bar{b}_i^{t+1} in (5.21) is the sum of two nonnegative terms. In this case inductively $\beta_i^t > 0$ and $\beta_r^t > 0$. Thus $\beta_i^{t+1} > 0$ since it is the sum of two positive terms in (5.21).

If $\bar{a}_{is}^t > 0$ and $\bar{b}_i^{t+1} = 0$, this can only happen if

$$\bar{b}_i^{t+1} = \bar{a}_{is}^t \left(\frac{\bar{b}_i^t}{\bar{a}_{is}^t} - \frac{\bar{b}_r^t}{\bar{a}_{rs}^t} \right) = 0; \quad (5.22)$$

in other words, i and r are tied and so both i and r are in \mathcal{R}^t , see (5.12). In this case by (5.13), r is chosen so that, rewriting the expression for β_i^{t+1} ,

$$\beta_i^{t+1} = \bar{a}_{is}^t \left(\frac{\beta_i^t}{\bar{a}_{is}^t} - \frac{\beta_r^t}{\bar{a}_{rs}^t} \right). \quad (5.23)$$

the bracket term is positive (zero is not possible because of properties (1) and (2)).

Finally, if $\bar{a}_{is}^t = 0$, then $\bar{b}_i^{t+1} = \bar{b}_i^t$ and $\beta_i^{t+1} = \beta_i^t$ so that property (3) holds for $t + 1$ if it holds for t . Therefore $\bar{b}_i^{t+1} = 0$ implies $\beta_i^{t+1} > 0$.

To prove Property (4) note that

$$\bar{z}_{t+1} = \bar{z}_t + \bar{c}_s^t \frac{\bar{b}_r^t}{\bar{a}_{rs}^t}, \quad \gamma_{t+1} = \gamma_t + \bar{c}_s^t \frac{\beta_r^t}{\bar{a}_{rs}^t},$$

where $\bar{c}_s^t < 0$, so that either $\bar{z}_{t+1} < \bar{z}_t$ if $\bar{b}_r^t > 0$ or $\bar{z}_{t+1} = \bar{z}_t$ if $\bar{b}_r^t = 0$ in which case $\gamma_{t+1} < \gamma_t$ since $\beta_r^t > 0$ in this case. ■

THEOREM 5.4 (Finite Termination Using Krishna's Rule) *The Simplex Algorithm will terminate in a finite number of iterations.*

Proof. It is clear that as long as \bar{z} strictly decreases from iteration to iteration, each canonical form generated must be different from all others previously generated. When \bar{z} does not decrease it is said to *stall*. It can only remain stalled a finite number of iterations because for the sequence of t stalled, $\bar{b}_r^t = 0$ and γ_t is strictly decreasing, implying that their corresponding canonical forms are all different. ■

- ▷ **Exercise 5.12** Why does the nonrepetition of the canonical forms imply a finite number of iterations?

- ▷ **Exercise 5.13** The *order of degeneracy* is defined as the number of basic variable values equal to zero on iteration t . Prove that if the order of degeneracy is at most 1 on every iteration t , the Simplex Method, implemented without any rule for resolving ties, will always converge in a finite number of iterations.

- ▷ **Exercise 5.14** Show that if the auxilliary problem right-hand side for Wolfe's Rule is modified using Krishna's Rule, then $d_i > 1$ is not possible.

5.7 ON AVOIDING DEGENERATE PIVOTS

So far we have discussed how to avoid cycling under degeneracy. A related issue is how to avoid degenerate pivots so that we can get a nonzero decrease in the objective function. Although it is not possible to totally avoid such pivots in the Simplex Method, we can attempt to bypass some of them.

For simplicity of exposition, consider a linear program in standard form rather than one with bounded variables. The choice of variable entering a basis is based on computing the reduced costs σ_N as the solution to:

$$\sigma_N = c_N - N^T \pi, \quad \text{where } B^T \pi = c_B.$$

In the Revised Simplex Method, we pick a $\sigma_s < 0$ and compute a search direction p_B^s by solving $Bp_B^s = -A_{\bullet s}$. Then we find the variable to leave the basis by finding the largest $\alpha \geq 0$ such that feasibility is maintained, i.e., $x_B + \alpha p_B^s \geq 0$. If, for some i ,

$$(x_B)_i = 0 \quad \text{and} \quad [p_B^s]_i < 0$$

then $\alpha = 0$. This results in a degenerate pivot with no change to the objective. If possible we would like to avoid such a step. Thus, if we could foresee in advance that a particular search direction would result in a step $\alpha = 0$, we could, by selecting a different direction, succeed in bypassing a degenerate pivot. One obvious way to do this is to scan j such that $\sigma_j < 0$, one by one, stopping when a direction is found

that results in a step $\alpha > 0$ or, if no such direction is found, then one of these directions is chosen whose $\alpha = 0$. However, this method is computationally very expensive if many $\sigma_j < 0$. Fortunately it is possible to efficiently determine that some of the directions will definitely cause degenerate pivots. Then we can choose among the remaining directions and hope for the best. This approach requires computing an additional vector γ_N , analogous to computing the reduced costs σ , such that if $\gamma_j < 0$ we know for sure that the associated direction p^j will result in a degenerate pivot step and thus this j should be avoided if possible. The new rule for column selection would then be

$$\sigma_s = \min_j \{ \sigma_j < 0 \text{ such that } \gamma_j \geq 0 \}. \quad (5.24)$$

if such an s exists.

To create such a γ , we define an additional “objective” function $f = d^T x$ where

$$d_i = \begin{cases} 1 & \text{if } i \in \mathcal{B} \text{ and } x_i = 0 \\ 0 & \text{otherwise.} \end{cases} \quad (5.25)$$

We use the “objective” f to compute the simplex multiplier ρ with respect to d by $B^T \rho = d_B$, and the relative costs γ_N as follows:

$$\gamma_N = d_N - N^T \rho = -N^T \rho \quad (5.26)$$

since $d_N = 0$ by definition. We now show that γ_N provides us with information that allows us to not consider some of the directions p^j that would result in degenerate pivot steps. To see this, substitute $\rho = (B^T)^{-1} d_B$ in (5.26):

$$\gamma_N = -N^T \rho = -(B^{-1} N)^T d_B.$$

Thus, for each $j \in \mathcal{N}$,

$$\gamma_j = -(B^{-1} A_{\bullet j})^T d_B = -(p_B^j)^T d_B = \sum_{\{i \in \mathcal{B} | x_i = 0\}} p_i^j \quad (5.27)$$

where p_B^j is the solution to $B p_B^j = -A_{\bullet j}$, or in tableau form $p_B^j = -\bar{A}_{\bullet j}$.

From (5.27), we see that γ_j is the *sum* over all the components of p^j that correspond to basic variables at 0. Thus, if $\gamma_j < 0$ it implies that at least one element of p_B^j is negative such that the corresponding component of x_B is zero. That is, from (5.27), $\gamma_j < 0$ implies that there exists an $i_0 \in \mathcal{B}$ such that $x_{i_0} = 0$ and $p_{i_0}^j < 0$. This implies a step of $\alpha = 0$. Thus, in order to attempt to avoid the possibility of degenerate steps we choose the incoming variable x_s by (5.24). If no such σ_s exists we choose σ_s as the minimum over the σ_j only. Note that if $\gamma_j \geq 0$ it is still possible that one or more $p_i^j < 0$ and $x_i = 0$ for $i \in \mathcal{B}$ implying a degenerate pivot. However, in practical applications it has been observed that the use of (5.24) reduces the number of pivot steps significantly.

5.8 NOTES & SELECTED BIBLIOGRAPHY

In 1951, A. J. Hoffman [1953] constructed an example to demonstrate cycling in a linear program involving three equations and eleven variables. E. M. L. Beale [1955a] constructed a second example, that is remarkable for its simplicity. Hoffman's and Beale's examples displayed in this chapter are adapted from Dantzig [1963]. A. W. Tucker constructed an even simpler, but totally degenerate, example to demonstrate cycling. See also Kotiah & Steinberg [1977, 1978] and Marshall & Suurballe [1969].

Dantzig [1963] proved that the Random Choice Rule prevented cycling with probability 1, see Problem 5.1. Earlier, in 1951, Dantzig proposed a method of perturbation of the right-hand side (row i was perturbed by the addition of ϵ^i for any $0 < \epsilon^i < \epsilon^*$) as a way of avoiding degeneracy when using the Simplex Method, the proofs of which he outlined and gave as homework exercises to classes he was teaching at the time. Edmondson turned in a proof in March 1951, see Edmondson [1951] and Dantzig [1951b]; also see Problem 5.9. In the summer of 1951, Philip Wolfe, then a student at Berkeley, spent the summer with Dantzig at the Pentagon and proposed a lexicographic interpretation of the perturbation idea (which later Dantzig, Orden, & Wolfe [1955] published as a joint paper), see Problem 5.12. The basic idea was to consider attaching an identity matrix to the right-hand side; then ties were broken by applying the min ratio test to the first column of the basis inverse, followed by the second column, etc. Independently, at about the same time, A. Charnes (see Charnes [1952]) developed a different perturbation scheme. Years later, Wolfe [1963], see Section 5.4, proposed a third way based on Dantzig's [1960a] inductive proof of the Simplex Method that is very elegant because it resolves degeneracy using only one extra column of index pointers. The approach of Section 5.6 for attaching an extra column β for resolving degeneracy is due to A. Krishna [1989].

For details on Bland's method see Bland [1977]. Harris's procedure can be found in Harris [1975]. Gill, Murray, Saunders, & Wright [1989] developed a practical "anti-cycling" procedure. The paper also shows that their method can be viewed as a modification of Wolfe's procedure. A version of their "anti-cycling" procedure is implemented by Murtagh and Saunders in the popular optimization software MINOS.

The approach presented in Section 5.7 for avoiding degenerate pivots was first shown by Kalan [1976]. For additional work on this and related approaches, see Greenberg [1978c], Klotz [1988], and Nazareth [1987]. Klotz [1988] has run many test problems that show that the computational time can be reduced significantly by using degeneracy resolving schemes.

Dantzig [1988b] developed a simple anti-cycling device that avoids dual degeneracy of the parameterized objective with probability "one." This approach shows that it is not necessary to use row selection rules for resolving degeneracy; instead it shows that independent of what tying row is selected to choose the outgoing variable, the Gass & Saaty [1955b] parametric method applied to the objective function can be used to choose the incoming column. Tests run on a set of nine problems showed 56% improvement in the number of iterations and a CPU time reduction of approximately 48%. On a computer, most implementations use partial pricing; an adaptation of this approach to partial pricing that guarantees finite convergence is an open problem.

In addition to the already mentioned researchers, many others have suggested anti-cycling techniques; see, for example, Balanski & Gomory [1963], Benichou, Gauthier, Hentges, & Ribi re [1977], and Rockafellar [1984]. Fletcher [1985, 1987] developed a method for resolving degeneracy that is designed to display favorable properties in the presence of rounding error.

Whether any degeneracy avoiding scheme is needed in practice has never been settled. It has been observed, however, that even when there is no degeneracy, there is a high probability of “near”-degeneracy. This suggests that pivot-selection criteria should be designed to seek feasible solutions in directions away from degenerate and “near”-degenerate basic feasible solutions, or better yet, driven by dual feasibility considerations. Doing so has been observed to reduce the total number of iterations in highly degenerate cases. The practical benefits of anti-cycling are many. For a discussion on this see, for example, Faulkner [1988], Gill, Murray, Saunders, & Wright [1989], and Ryan & Osborne [1988].

5.9 PROBLEMS

- 5.1 *Random Choice Rule.* This rule states that the outgoing basic variable at any iteration of the Simplex Algorithm is chosen from among those r satisfying

$$\frac{\bar{b}_r}{\bar{a}_{rs}} = \text{Min}_{i: \bar{a}_{is} > 0} \frac{\bar{b}_i}{\bar{a}_{is}}$$

with equal probability.

- (a) Show that the Simplex Algorithm, using the Random Choice Rule, will terminate in k iterations with probability

$$P \geq \left(\frac{1}{m}\right)^k$$

where m is the number of equations and k is the longest of the shortest chain leading to an optimal canonical form.

- (b) Show that, using the Random Choice Rule, the probability of failing to reach an optimum in N iterations tends to zero as $N \rightarrow \infty$.

- 5.2 Solve Tucker’s cycling example (see Example 5.3) using Wolfe’s Rule.
 5.3 Solve Tucker’s cycling example (see Example 5.3) using Bland’s Rule.
 5.4 Solve Tucker’s cycling example (see Example 5.3) using Krishna’s Rule, where instead of π , the approximation 3.1415926 is used.
 5.5 Solve Tucker’s cycling example (see Example 5.3) using a Random Choice Rule.
 5.6 Solve Beale’s cycling example (see Example 5.2) using Bland’s Rule.
 5.7 Solve Beale’s cycling example (see Example 5.2) using Krishna’s Rule, where instead of π , the approximation 3.1415926 is used.
 5.8 Solve Beale’s cycling example (see Example 5.2) using a Random Choice Rule.
 5.9 *Dantzig [1951b] & Edmondson [1951] Perturbation Method.* Consider a linear program in canonical form:

$$\begin{aligned} & \text{Minimize} && c^T x \\ & \text{subject to} && Ax = b, && A: m \times n, \\ & && x \geq 0, \end{aligned} \tag{5.28}$$

where A is a rectangular matrix of dimension $m \times n$, $b \geq 0$ is a column vector of dimension m , c is a column vector of dimension n , and x is a column vector of dimension n . For $i = 1, \dots, m$, perturb each right-hand side b_i by adding ϵ^i where $0 < \epsilon < \epsilon^*$ and ϵ^* is a small number.

- (a) Show that on subsequent iterations, the right-hand sides will become polynomial expressions in ϵ of the form

$$b = B^{-1} \begin{pmatrix} \epsilon \\ \epsilon^2 \\ \vdots \\ \epsilon^m \end{pmatrix}. \quad (5.29)$$

- (b) Letting β_{ij} denote element (i, j) , show that for each i there exists a k such that $\beta_{ik} \neq 0$.
 (c) Show that no two rows of the inverse are proportional to each other.
 (d) Consider the m -order polynomial

$$f(\epsilon) = a_0 + a_1\epsilon + \cdots + a_m\epsilon^m. \quad (5.30)$$

Show that there exists a $\epsilon^* > 0$ such that $f(\epsilon) > 0$ for all $0 < \epsilon < \epsilon^*$ if and only if not all $a_i = 0$ and the nonzero term with the lowest index i has a positive coefficient.

- (e) Given two polynomials

$$f(\epsilon) = \sum_{i=0}^m a_i \epsilon^i, \quad g(\epsilon) = \sum_{i=0}^m b_i \epsilon^i, \quad (5.31)$$

then for some range $0 \leq \epsilon \leq \epsilon^*$, $f(\epsilon) < g(\epsilon)$ if for some k

$$a_i = b_i \quad \text{for } i = 1, \dots, k-1$$

$$a_k < b_k$$

$$\text{and } a_i, \quad b_i \quad \text{arbitrary for } i = k+1, \dots, m$$

- (f) From part (a) we know that for any iteration t , each polynomial expression in ϵ in (5.29) has at least one nonzero term. Show that if the first term is positive for every i , then there is a range of values $0 < \epsilon < \epsilon_t^*$ such that for any fixed ϵ in the range, the values of all basic variables are positive.
 (g) *Lexicographic Rule.* The maximum value \bar{x}_s of the entering variable x_s and the choice of which variable j_r to drop from the basic set is determined by

$$\bar{x}_s = \frac{\bar{b}_r(\epsilon)}{\bar{a}_{rs}} = \text{Min}_{\bar{a}_{is} > 0} \left\{ \frac{\bar{b}_i + \sum_{k=1}^m \beta_{ik} \epsilon^k}{\bar{a}_{is}} \right\} \quad (5.32)$$

Show that the minimum of several polynomial expressions is found by first comparing the constant terms; if there are ties, then the vector of coefficients corresponding to ϵ are used in the comparison; if there are still ties, then the vector of coefficients corresponding to ϵ^2 are used in the comparison; and so on. Show that it is not possible to have a tie at the end of the process.

- (h) Show that there exists a common range of values $0 < \epsilon < \epsilon^*$ such that for any finite number of iterations of the Simplex Method as applied to any perturbed problem within the range, the values of all basic variables remain positive and the choice of the variable entering and leaving the basic set is unique and independent of the particular value of ϵ in the range.

- (i) Show that the Simplex Algorithm as applied to the perturbed problem terminates in a finite number of iterations.
- (j) Show that the optimal basic feasible solution of the perturbed problem will yield the corresponding solution for the unperturbed problem by setting $\epsilon = 0$ in (5.29).

5.10 Dantzig [1963].

- (a) Is it possible to construct a class of perturbed problems that are infeasible, but the corresponding class of unperturbed problems are feasible?
- (b) Can the class of perturbed problems be feasible, but the unperturbed problem infeasible?
- (c) Can the class of perturbed problems have a finite lower bound for z , but not the unperturbed?
- (d) Can the class of perturbed problems have a lower bound of $-\infty$ for z , but not the unperturbed?

5.11 Charnes [1952] *Perturbation Method*. Show how to develop an alternative perturbation scheme where the right-hand sides are replaced by polynomial expressions $b_i(\epsilon)$, i.e.,

$$b_i(\epsilon) = b_i + \sum_{j=1}^n a_{ij}\epsilon^j. \quad (5.33)$$

5.12 Dantzig, Orden, & Wolfe's [1955] *Lexicographic Method*.

- (a) An m -component vector A is said to be *lexico-positive*, denoted by $A \succ 0$, if at least one component is nonzero and the first such is positive. The term *lexico* is a word suggested by A. W. Tucker because of its analogy to sorting of names in alphabetical order. A vector A is said to be *lexico-greater* than B , written $A \succ B$, if $A - B \succ 0$. The smallest of several vectors will be denoted by Lexico-Min. Prove that this lexicographic ordering of vectors is transitive, in other words

$$A \succ B \quad \text{and} \quad B \succ C \quad \implies \quad A \succ C.$$

- (b) Instead of perturbing constraints, suppose that the constants b_i are replaced by vectors:

$$\begin{aligned} \hat{b}_1^T &= (b_1, 1, 0, \dots, 0) \\ \hat{b}_2^T &= (b_2, 0, 1, \dots, 0) \\ &\vdots \\ \hat{b}_m^T &= (b_m, 0, 0, \dots, 1). \end{aligned}$$

- (a) Show, analogous to (5.29), that the vector of basic variables and objective value on some subsequent iteration are replaced by matrix \bar{b} and vector \bar{z} where

$$\begin{aligned} \bar{b} &= B^{-1}b, \quad B^{-1} \\ \bar{z} &= z_t, \quad \pi^T \end{aligned}$$

- (b) Show that the basic variable j_r chosen to be dropped is selected so that

$$\frac{\bar{b}_r}{\bar{a}_{rs}} = \text{Lexico-Min} \left\{ \frac{\bar{b}_i}{\bar{a}_{is}} \right\}$$

and the choice of r is unique.

- (c) Prove that $\bar{b}_i > 0$ for all iterations implying $z_0 \succ z_1 \succ z_2 \succ \dots$ (i.e., a strictly lexico decreasing sequence of vectors), implying that no canonical form can repeat.

- 5.13 Suppose you have a linear program with three constraints and you wish to use lexicography (see Problem 5.12) to resolve degeneracy. Instead of adding a 3×3 identity matrix to the right-hand side, you decide to add the matrix:

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 2 & 1 \\ 1 & 4 & 1 \end{bmatrix}.$$

Can you still guarantee that all ties will be broken on the current and all subsequent iterations. Why or why not?

- 5.14 *Ph.D. Comprehensive Exam, September 21, 1991, at Stanford.*

- (a) Most commercial software packages for solving linear programs by the Simplex Method have no protection against cycling. Why is this a safe procedure in practice?
- (b) On the other hand, some commercial software packages for solving linear programs by the Simplex Method do have anti-cycling procedures. Does it have any purpose other than avoiding cycling?
- (c) State the names of a number of ways that cycling can be avoided.
- (d) Outline a proof of why one of the anti-cycling rules works.
- 5.15 *Ph.D. Comprehensive Exam, March 30, 1970, at Stanford.* Let (P_1, P_2, \dots, P_m) be m linearly independent vectors in m -space, and let P_0 be any other m -vector. Prove that if we let

$$x_1 P_1 + x_2 P_2 + \dots + x_m P_m = P_0 + \begin{pmatrix} \varepsilon \\ \varepsilon^2 \\ \vdots \\ \varepsilon^m \end{pmatrix}$$

then there exists an $\varepsilon_0 > 0$ such that for ε satisfying $0 < \varepsilon < \varepsilon_0$,

$$x_i \neq 0 \quad \text{for all } i = 1, \dots, m.$$

- 5.16 *Ph.D. Comprehensive Exam, September 23, 1972, at Stanford.* The system

$$Ax = b, \quad x \geq 0, \quad A \in R^{m \times n} \tag{5.34}$$

is said to be *nondegenerate* if it is feasible and if each solution x has at least m positive components. A k -dimensional bounded polyhedral convex set P is said to be *regular* if each extreme point is adjacent to exactly k other extreme points.

- (a) If $P = \{x \geq 0 \mid Ax = b\}$ is nonempty and bounded, does regularity of P imply nondegeneracy of (5.34), or vice versa, or are they equivalent?
- (b) Assuming (5.34) is feasible, when can one perturb the right-hand side b to obtain nondegeneracy?
- (c) View the set P defined in part (a) as the convex hull of its extreme points. Can P be made regular by perturbing these extreme points?

5.17 *Ph.D. Comprehensive Exam, September 26, 1980, at Stanford.* Consider a linear program of the form

$$\begin{array}{ll} \text{Minimize} & c^T x \\ \text{subject to} & Ax = b, \\ & x \geq 0. \end{array}$$

where the matrix A is $m \times n$. Let $\beta = \{j_1, j_2, \dots, j_m\}$ denote a basic set of indices for which the corresponding basic solution x° is feasible but degenerate with $x_{j_1}^\circ = 0$. Suppose that for every choice of nonnegative values of the nonbasic variables (*except all zero*) the value of x_{j_1} is negative.

- (a) Prove that the basic feasible solution x° is optimal.
- (b) Find a way to obtain dual prices that prove the optimality of x° .

This page intentionally left blank

VARIANTS OF THE SIMPLEX METHOD

By a *variant* of the Simplex Method (in this chapter) we mean an algorithm consisting of a sequence of pivot steps in the primal system using alternative rules for the selection of the pivot. Historically these variants were developed to take advantage of a situation where an infeasible basic solution of the primal is available. In other applications there often occurs a set of problems differing from one another only in their constant terms and cost factors. In such cases, it is convenient to omit Phase I and to use the optimal basis of one problem as the initial basis for the next.

6.1 INTRODUCTION

Several methods have been proposed for varying the Simplex Algorithm to reduce the number of iterations. This is especially needed for problems involving many equations in order to reduce the computation time. It is also needed for problems involving a large number of variables n , for the number of iterations in practice appears to grow roughly proportional to n .

For example, instead of using the selection rule $\bar{c}_s = \min \bar{c}_j$, one could select $j = s$ such that introducing x_s into the basic set gives the largest decrease in the value of z in the next basic solution. This requires computing, for $\bar{c}_j < 0$, the largest in absolute value of $\bar{c}_j \theta_j$, where θ_j is determined so that if x_j replaced x_{j_r} then the solution will remain feasible. This rule is obviously not practical when using the revised Simplex Method with multipliers. Even using the standard canonical form, considerably more computations would be required per iteration. It is possible, however, in the nondegenerate case, to develop a modification of the canonical form in which the coefficient of the i th basic variable is allowed to be different from unity

in the i th equation but $\bar{b}_i = 1$. In this form the selection of s by the steepest descent criterion would require a little more effort; moreover (by means of a special device), no more effort than that for the standard Simplex Algorithm would be required to maintain the tableau in proper modified form from iteration to iteration (see Section 6.2 for details).

The simplest variant occurs when the new problem differs from the original in the cost coefficients alone. In this case, the cost coefficients are replaced by the new ones, and Phase II of the Revised Simplex Method is applied. Another possibility is to use the parametric objective method to be described in Section 6.4.

An important variant occurs when the new problem differs from the original in the constant terms only. In this case the optimal basis of the first problem will still price out dual feasible, i.e., $\bar{c}_j \geq 0$, for the second, but the associated primal solution may not be feasible. For this situation, we could use the *Dual-Simplex* Algorithm, which is the variant of the standard Primal-Simplex Algorithm, to be discussed in Section 6.3, or the parametric right-hand-side method to be described in Section 6.4, or the *Primal-Dual* method of Section 6.6.

However, when the problems differ by more than either the constant terms or the cost coefficient terms, the old basis may be neither primal feasible nor dual feasible. When neither the basic solution nor the dual solution generated by its simplex multipliers remains feasible, the corresponding algorithm is called *composite*. The *Self-Dual* parametric algorithm discussed in Section 6.5 is an example of such a composite algorithm.

Correspondence of Primal and Dual Bases.

In 1954, Lemke discovered a certain correspondence between the bases of the primal and dual systems that made it possible to interpret the Simplex Algorithm *as applied to the dual* as a sequence of basis changes in the primal; this interpretation is called the Dual-Simplex Algorithm. From a computational point of view, the Dual-Simplex Algorithm is advantageous because the size of the basis being manipulated in the computer is $m \times m$ instead of $n \times n$. In this case, however, the associated basic solutions of the primal are not feasible, but the simplex multipliers continue to price out optimal (hence, yield a basic feasible solution to the dual). It is good to understand the details of this correspondence, for it provides a means of easily dualizing a problem without transposing the constraint matrix.

Consider the standard form

$$\begin{array}{ll} \text{Minimize} & c^T x = z \\ \text{subject to} & Ax = b, \quad A: m \times n, \\ & x \geq 0, \end{array} \quad (6.1)$$

and the dual of the standard form:

$$\begin{array}{ll} \text{Maximize} & b^T \pi = v \\ \text{subject to} & A^T \pi \leq c, \quad A: m \times n, \end{array} \quad (6.2)$$

PRIMAL-DUAL CORRESPONDENCES		
	Primal	Dual
Basis	B	$\bar{B} = \begin{pmatrix} B^T & 0 \\ N^T & I \end{pmatrix}$
Basic variables	x_B	$\pi, y_N = \bar{c}_N$
Nonbasic variables	x_N	$y_B = \bar{c}_B$
Feasibility condition	$Ax = b, x \geq 0$	$\bar{c} \geq 0$

Table 6-1: Primal-Dual Correspondences

where π_i is unrestricted in sign. We assume A is full rank. Adding slack variables $y \geq 0$ to the dual problem (6.2) we get:

$$\begin{aligned} &\text{Maximize} && b^T \pi &= v \\ &\text{subject to} && A^T \pi + Iy = c, & A: m \times n, \\ &&& y \geq 0. & \end{aligned} \quad (6.3)$$

Clearly the dual has n basic variables and m nonbasic variables. The variables π are unrestricted in sign and, when A is full rank, always constitute m out of the n basic variables of the dual. The basis for the primal is denoted by B and the nonbasic columns are denoted by N . Note that $y = c - A^T \pi = \bar{c} \geq 0$ when the dual is feasible and that $\bar{c}_B = 0$ and $\bar{c}_N \geq 0$ when the primal is optimal. The basic and nonbasic columns for the dual are denoted by \bar{B} , an $n \times n$ matrix, and \bar{N} , an $n \times m$ matrix

$$\bar{B} = \begin{pmatrix} B^T & 0 \\ N^T & I_{n-m} \end{pmatrix}, \quad \bar{N} = \begin{pmatrix} I_m \\ 0 \end{pmatrix}, \quad (6.4)$$

where I_{n-m} is an $(n-m)$ -dimensional identity matrix and I_m is an m -dimensional identity matrix. Thus, (π, y_N) as basic variables and y_B as nonbasic variables constitute a basic solution to the dual if and only if $y_B = 0$.

- ▷ **Exercise 6.1** Show that the determinant of B has the same value as that of \bar{B} . Also show that if B^{-1} exists, then \bar{B}^{-1} exists.

It is now clear that there is a correspondence between primal and dual bases. These correspondences are shown in Table 6-1. With these correspondences in mind, we shall discuss variations of the Simplex Method.

- ▷ **Exercise 6.2** How is the concept of complementary primal and dual variables related to the correspondence of primal and dual bases?

Definition (Dual Degeneracy): We have already defined degeneracy and non-degeneracy with respect to the primal. A basis is said to be *dual degenerate*

if one or more of the \bar{c}_j corresponding to nonbasic variables x_j are zero and to be *dual nondegenerate* otherwise.

6.2 MAX IMPROVEMENT PER ITERATION

In this section we present an alternative canonical form for efficiently determining the incoming column that yields the maximum improvement per iteration. It has been observed on many test problems that this often leads to fewer iterations. We will assume, to simplify the discussion, that all basic feasible solutions that are generated are nondegenerate.

Assume $x_j \geq 0$ for $j = 1, \dots, n$ and x_j for $j = 1, \dots, m$ are basic variables, in the *standard canonical form*:

$$\begin{array}{rcccc}
 -z & & + \bar{c}_{m+1}x_{m+1} + \cdots + \bar{c}_jx_j + \cdots + \bar{c}_nx_n & = & -\bar{z}_0 \\
 x_1 & & + \bar{a}_{1,m+1}x_{m+1} + \cdots + \bar{a}_{1j}x_j + \cdots + \bar{a}_{1n}x_n & = & \bar{b}_1 \\
 & \ddots & & & \vdots \\
 & & & & \vdots \\
 x_p & & + \bar{a}_{p,m+1}x_{m+1} + \cdots + \bar{a}_{pj}x_j + \cdots + \bar{a}_{pn}x_n & = & \bar{b}_p \\
 & \ddots & & & \vdots \\
 & & & & \vdots \\
 & & x_m + \bar{a}_{m,m+1}x_{m+1} + \cdots + \bar{a}_{mj}x_j + \cdots + \bar{a}_{mn}x_n & = & \bar{b}_m
 \end{array} \tag{6.5}$$

where \bar{a}_{ij} , \bar{c}_j , \bar{b}_i , and \bar{z}_0 are constants and we assume that $\bar{b}_i > 0$ for $i = 1, \dots, m$. If there are two or more $\bar{c}_j < 0$, our problem is to find which $j = s$ among them has the property that putting column s into the basis and driving some column $j = r$ out of the basis gives the greatest decrease in z while preserving feasibility. Using the standard canonical form (6.5), the *main work* is that of performing ratio tests for the several columns j such that $\bar{c}_j < 0$, plus the update work of pivoting, which takes mn operations where each operation consists of one addition (or subtraction) plus one multiplication. To do this more efficiently, consider the *alternative canonical form*:

$$\begin{array}{rcccc}
 -z & & + \bar{c}_{m+1}x_{m+1} + \cdots + \bar{c}_nx_n & = & -\bar{z}_0 \\
 \alpha_{11}x_1 & & + \alpha_{1,m+1}x_{m+1} + \cdots + \alpha_{1n}x_n & = & \mathbf{1} \\
 & \ddots & & & \vdots \\
 & & & & \vdots \\
 \alpha_{pp}x_p & & + \alpha_{p,m+1}x_{m+1} + \cdots + \alpha_{pn}x_n & = & \mathbf{1} \\
 & \ddots & & & \vdots \\
 & & & & \vdots \\
 & & \alpha_{mm}x_m + \alpha_{m,m+1}x_{m+1} + \cdots + \alpha_{mn}x_n & = & \mathbf{1}
 \end{array} \tag{6.6}$$

formed by rescaling the rows by dividing them by $\bar{b}_i > 0$ for all $i = 1, \dots, m$. In this format we scan column j such that $\bar{c}_j < 0$ and find for each such j the row

$$r_j = \operatorname{argmax} \alpha_{ij}.$$

If $\alpha_{r_j j} \leq 0$ for any j such that $\bar{c}_j < 0$ then a class of solutions can be constructed so that $z \rightarrow -\infty$ in the usual way. If $\alpha_{r_j j} > 0$ for all j such that $\bar{c}_j < 0$, then the best choice for maximum decrease of the objective value in the next iteration is $j = s$ where

$$s = \operatorname{argmin}_{\{j|\bar{c}_j < 0\}} (\bar{c}_j / \alpha_{r_j j}). \quad (6.7)$$

The corresponding pivot row is

$$r = r_s. \quad (6.8)$$

The pivot operations require mn operations but the rescaling of the rows such that the right hand sides are all ones requires an additional mn multiplications so that this format is inferior to (6.5) in that it requires considerably more work per iteration.

We can, however, perform the operations efficiently by considering the following *revised alternative canonical form* that is generated from (6.6) by subtracting some row p , called a *key row*, from other rows $i \neq p$:

$$\begin{array}{rcll} -z & & + \bar{c}_{m+1}x_{m+1} + \cdots + \bar{c}_n x_n = & -\bar{z}_0 \\ \beta_{11}x_1 & + \beta_{1p}x_p & + \beta_{1,m+1}x_{m+1} + \cdots + \beta_{1n}x_n = & 0 \\ & \ddots & \vdots & \vdots \\ & \beta_{pp}x_p & + \beta_{p,m+1}x_{m+1} + \cdots + \beta_{pn}x_n = & 1 \\ & \ddots & \vdots & \vdots \\ \beta_{mp}x_p & \ddots & + \beta_{m,m+1}x_{m+1} + \cdots + \beta_{mn}x_n = & 0 \end{array} \quad (6.9)$$

where $\beta_{ij} = \alpha_{ij} - \alpha_{pj}$ for $i \neq p$ and $\beta_{pj} = \alpha_{pj}$. Let

$$\bar{r}_j = \operatorname{argmax}_{i \neq p} \alpha_{ij} = \operatorname{argmax}_{i \neq p} (\alpha_{ij} - \alpha_{pj}) = \operatorname{argmax}_{i \neq p} \beta_{ij}.$$

Note that

$$r_j = \begin{cases} p & \text{if } \alpha_{pj} > \alpha_{\bar{r}_j j}, \\ \bar{r}_j & \text{otherwise.} \end{cases}$$

Therefore:

1. The work to find the pivot row s and the corresponding pivot row $r = r_s$ from the revised alternative canonical form (6.9) is about the same as it was to find it from the alternative canonical form (6.6).
2. The work to update (6.9) for doing the pivot turns out, as we will see next, to be about the same as that for the standard canonical form (6.5), so that (6.9) is the preferred canonical form.

The updating of (6.9) is done in the following two steps:

1. If $r \neq p$, modify the β matrix before pivoting by making row p , where 1 appears on the right-hand side, the same as the pivot row r as follows. Make $p = r$ by

- (a) first subtracting row r of (6.9) from every other $i \neq p$. The adjusted β_{ij} for $i \neq r$ and $i \neq p$ are

$$\beta_{ij} \leftarrow \beta_{ij} - \beta_{rj} = (\alpha_{ij} - \alpha_{pj}) - (\alpha_{rj} - \alpha_{pj}) = (\alpha_{ij} - \alpha_{rj}).$$

- (b) Next add row p to row r . The adjusted β_{rj} are

$$\beta_{rj} \leftarrow \beta_{rj} + \beta_{pj} = (\alpha_{rj} - \alpha_{pj}) + \alpha_{pj} = \alpha_{rj}.$$

- (c) Then subtract the adjusted row r from row p . The adjusted β_{pj} is given by

$$\beta_{pj} \leftarrow \beta_{pj} - \beta_{rj} = \alpha_{pj} - \alpha_{rj}.$$

This step requires mn subtractions/additions.

2. The pivot update consists of

- (a) multiplying row r by 1, and
 (b) multiplying rows $i \neq r$ by $-\beta_{rs}/\beta_{is}$ provided $\beta_{is} \neq 0$; if $\beta_{is} = 0$ we multiply row i by 1. (The proof is given by the proof to Lemma 6.1.)

This step requires mn multiplications.

LEMMA 6.1 (Relation for Updated β_{ij}) *If $p = r$, the updated $\beta_{ij} = \bar{\beta}_{ij}$, for $i \neq r$, and $\beta_{is} \neq 0$, is obtained by multiplying β_{ij} for all j by the same ratio (β_{rs}/β_{is}) , namely, $\bar{\beta}_{ij} = -\beta_{ij}(\beta_{rs}/\beta_{is})$.*

Proof. At iteration t , the α format of Equation (6.9) for equation r and equation i is

$$\begin{aligned} \alpha_{r1}x_1 + \cdots + \alpha_{rj}x_j + \cdots + \alpha_{rs}x_s + \cdots + \alpha_{rn} &= 1 \\ \alpha_{i1}x_1 + \cdots + \alpha_{ij}x_j + \cdots + \alpha_{is}x_s + \cdots + \alpha_{in} &= 1 \end{aligned}$$

After pivoting

$$\begin{aligned} \alpha_{r1}x_1 + \cdots + \alpha_{rj}x_j + \cdots + \alpha_{rs}x_s + \cdots + \alpha_{rn} &= 1 \\ \hat{\alpha}_{i1}x_1 + \cdots + \hat{\alpha}_{ij}x_j + \cdots + 0x_s + \cdots + \hat{\alpha}_{in} &= 1 - \frac{\alpha_{is}}{\alpha_{rs}} \end{aligned}$$

where

$$\hat{\alpha}_{ij} = \alpha_{ij} - \alpha_{rj} \frac{\alpha_{is}}{\alpha_{rs}} \quad \text{for } j \neq s.$$

Dividing by $1 - \alpha_{is}/\alpha_{rs}$ so that the right-hand side is 1, the updated coefficient α_{ij} of the j th term is

$$\bar{\alpha}_{ij} = \frac{\alpha_{ij} - \alpha_{rj}(\alpha_{is}/\alpha_{rs})}{1 - (\alpha_{is}/\alpha_{rs})}.$$

Subtracting the equation r so that the right hand side is zero and clearing fractions, we get the updated β_{ij} as

$$\begin{aligned}\bar{\beta}_{ij} &= \frac{[\alpha_{ij} - \alpha_{rj}(\alpha_{is}/\alpha_{rs})] - \alpha_{rj}[1 - (\alpha_{is}/\alpha_{rs})]}{1 - (\alpha_{is}/\alpha_{rs})} \\ &= -\frac{\alpha_{ij} - \alpha_{rj}}{\alpha_{is} - \alpha_{rs}}\alpha_{rs} \\ &= -\beta_{ij}\left(\frac{\beta_{rs}}{\beta_{is}}\right).\end{aligned}\tag{6.10}$$

■

- ▷ **Exercise 6.3** Show that $\bar{\beta}_{rj} = \beta_{rj}$ for all j and that for all i for which $\beta_{is} = 0$, we have $\bar{\beta}_{ij} = \beta_{ij}$ for all j .
- ▷ **Exercise 6.4** In the β format (6.9) show how to get α_{rjj} after updating β_{ij} .
- ▷ **Exercise 6.5** Show how to update the relative cost factors in the β format (6.9).
- ▷ **Exercise 6.6** Modify the approach of maximum improvement per iteration for the degenerate case.

6.3 DUAL-SIMPLEX METHOD

In practice, after a solution to a linear program is obtained, it often happens that an adjustment is required for a slight modification of the original problem. If it turns out that this modified problem is primal infeasible but still prices out optimal, the Dual-Simplex Method can be used to efficiently find a feasible optimal solution to the original problem.

Recall that the *Primal-Simplex Algorithm* maintains primal feasibility at each iteration while trying to decrease the primal objective function. It does this at each iteration by reducing the infeasibility of the dual variable $\bar{c}_s < 0$ by pivoting column s into the basis in place of some basic column j_r while maintaining primal feasibility. The *Dual-Simplex Algorithm*, as we shall see, maintains dual feasibility at each iteration while trying to increase the dual objective function. It does this at each iteration by reducing the primal infeasibility of $x_{j_r} = \bar{b}_r < 0$ to $x_{j_r} = 0$ by pivoting column j_r out of the basis and bringing into the primal basis some column s .

We know by the Weak Duality Theorem 2.3 that when the dual problem is known to be feasible, the primal problem cannot be unbounded. Thus, the Dual-Simplex Method, that is usually applied to a problem which is dual feasible, will terminate

	Primal-Simplex Method	Dual-Simplex Method
Applied to	$\begin{aligned} c_N^T x_N &= z - z_o \\ I x_B + A_N x_N &= b \\ x &\geq 0 \end{aligned}$	$\begin{aligned} c_N^T x_N &= z - z_o \\ I x_B + A_N x_N &= b \\ x &\geq 0 \end{aligned}$
Optimality criterion	$\bar{c} \geq 0$	$\bar{b} \geq 0$
Selection of incoming new basic variable	If $\bar{c}_s = \min_{j \in N} \bar{c}_j < 0$ then choose x_s , i.e., pivot in <i>column</i> s .	If $\bar{b}_r = \min_i \bar{b}_i < 0$ then choose x_r , i.e., pivot in <i>row</i> r .
Selection of outgoing basic variable	If $\frac{\bar{b}_r}{\bar{a}_{rs}} = \min_{\substack{j \\ \bar{a}_{js} > 0}} \frac{\bar{b}_i}{\bar{a}_{is}} \geq 0$, drop x_r ; pivot in <i>row</i> r	If $\frac{\bar{c}_s}{-\bar{a}_{rs}} = \min_{\substack{j \\ -\bar{a}_{rj} > 0}} \frac{\bar{c}_j}{-\bar{a}_{rj}} \geq 0$, drop x_s ; pivot in <i>column</i> s
Pivot element	\bar{a}_{rs}	\bar{a}_{rs}
Objective function	$z = c_N^T x_N + z_o$ decreases	$z = c_N^T x_N + z_o$ increases

Table 6-2: Primal-Simplex and Dual-Simplex Methods

with an optimal solution or go unbounded, implying that the primal problem is infeasible.

In the Dual-Simplex Method, when viewed in terms of the primal variables, one first decides which basic variable to *drop* and then which nonbasic variable to *introduce* in its place. The relations between the Primal-Simplex Method applied to a linear program in standard form and the Dual-Simplex Method applied to its dual are discussed next (see Table 6-2 for a summary).

1. Optimality Criterion.

The Dual-Simplex Method operates with the same tableau as the Primal-Simplex Method. However, the relative cost factors are nonnegative from iteration to iteration ($\bar{c}_j \geq 0$ instead of $\bar{b}_i \geq 0$). The primal problem is feasible and optimal if in the canonical form $\bar{c} \geq 0$ and $\bar{b} \geq 0$. These conditions also imply feasibility and optimality for the dual. The dual is feasible if $\bar{c} \geq 0$ and is optimal if

$$\bar{b} \geq 0. \quad (6.11)$$

2. Selection of the Outgoing Basic Variable.

In the primal method the *incoming* column is selected first. Since the dual essentially works on the transpose, we select the pivot row or outgoing variable first. If the optimality criterion is not satisfied for the dual, then a pivot row r can be selected such that $\bar{b}_r < 0$. A commonly used criterion is to pick row r such that \bar{b}_r is the most negative of the \bar{b}_i , i.e.,

$$\bar{b}_r = \min_{\{i=1, \dots, m\}} \bar{b}_i < 0. \quad (6.12)$$

3. Selection of the Incoming Basic Variable.

In the Dual-Simplex Method the incoming variable s is selected *after* the outgoing variable r has been selected. After pivoting on an element, say s , in row r we need \bar{b}_r/\bar{a}_{rs} , the value of the modified right-hand side to become positive. Thus, we must have $\bar{a}_{rs} < 0$ since $b_r < 0$. At the same time, we wish to maintain dual feasibility by ensuring that $\bar{c} \geq 0$ after the pivot step. That is, we need:

$$\bar{c}_j - \bar{c}_s \frac{\bar{a}_{rj}}{\bar{a}_{rs}} \geq 0 \quad \text{for } j = 1, \dots, n.$$

If it turns out for r that $\bar{a}_{rj} \geq 0$ for all $j \in \mathcal{N}$, then the problem is primal-infeasible (see Exercise 6.7 that follows). Otherwise we pick the incoming column s such that

$$\frac{\bar{c}_s}{-\bar{a}_{rs}} = \min_{\{j|j \in \mathcal{N}, \bar{a}_{rj} < 0\}} \frac{\bar{c}_j}{-\bar{a}_{rj}}, \quad (6.13)$$

and pivot on \bar{a}_{rs} . Note that this pivot step is *dual-degenerate* if it turns out that $\bar{c}_s = 0$.

- ▷ **Exercise 6.7** Why is it obvious that if in selecting an incoming variable, it turns out for r that $\bar{a}_{rj} \geq 0$ for all $j \in \mathcal{N}$, then the problem is primal-infeasible.

Just as with the Primal-Simplex Method, the Dual-Simplex Method can take degenerate steps and cycle indefinitely. One of the many primal degeneracy resolving techniques can be modified and used to prevent cycling in the Dual-Simplex Method.

- ▷ **Exercise 6.8** Construct a degeneracy resolving scheme to prevent cycling in the Dual-Simplex Method.

- ▷ **Exercise 6.9** For the primal method, the incoming column s can be represented in terms of the basis by $\bar{A}_{\bullet s}$. Referring to (6.4) show that the outgoing column r of B in the primal corresponds, in the dual, to a unit column of \bar{N} with component r equal to unity and the representation of this column in terms of \bar{B} is

$$B_{r\bullet}^{-1}; -\bar{a}_{r,m+1}, -\bar{a}_{r,m+2}, \dots, -\bar{a}_{r,n}. \quad (6.14)$$

We will illustrate the Dual-Simplex Method first when there are no artificial variables and then when artificials are present.

Example 6.1 (Illustration of the Dual-Simplex Method) Suppose a system has been transformed to yield

$$\begin{array}{rcl} (-z) & + & x_4 + 2x_5 + 4x_6 = -5 \\ x_1 & + & 2x_4 - 5x_5 + 5x_6 = 9 \\ x_2 & - & \mathbf{3x_4} + 6x_5 - 3x_6 = -3 \\ x_3 & + & x_4 - 4x_5 + 2x_6 = 2. \end{array} \quad (6.15)$$

Since all \bar{c}_j but not all constant terms, are nonnegative, we apply the Dual-Simplex Method. Thus, first we drop the basic variable, x_2 , because

$$\bar{b}_r = \bar{b}_2 = \min_i \bar{b}_i = -3.$$

Next we introduce x_4 into the next basic set, because $j = 4$ is determined by the criterion,

$$s = \operatorname{argmin}_{\{j|\bar{a}_{2j}<0\}} \frac{\bar{c}_j}{-\bar{a}_{2j}} = \operatorname{argmin} \left\{ \frac{\bar{c}_4}{-\bar{a}_{24}} = \frac{1}{3}, \frac{\bar{c}_6}{-\bar{a}_{26}} = \frac{4}{3} \right\} = 4.$$

After pivoting, the system becomes (6.16). Since all \bar{b}_i and \bar{c}_j are nonnegative, the basic solution is now feasible and optimal.

$$\begin{array}{rcll} (-z) & + \frac{1}{3}x_2 & & + 4x_5 + 3x_6 = -6 \\ x_1 & + \frac{2}{3}x_2 & & - 1x_5 + 3x_6 = 7 \\ & - \frac{1}{3}x_2 & + x_4 - 2x_5 + x_6 & = 1 \\ & + \frac{1}{3}x_2 + x_3 & & - 2x_5 + x_6 = 1. \end{array} \quad (6.16)$$

We now consider the case when artificials are present. *Conceptually, any artificial basic variable, x_j , whose value is positive in the basic solution, may be replaced by $-x'_j = x_j$, so that the basic solution becomes "infeasible," which allows the application of the Dual-Simplex Method.* In practice, it is probably better not to make the formal substitution, $x_j = -x'_j$, for artificial variables of positive value, but to modify the rules of the procedure to produce the same effect. The next example illustrates the Dual-Simplex Method when artificial variables are present.

Example 6.2 (Dual-Simplex Method with Artificial Variables) Suppose, for Example 6.1, that x_2 and x_3 are artificial, meaning we seek a solution in which $x_2 = 0$ and $x_3 = 0$. We shall proceed as before; however, we shall disregard all artificial variables once they drop out of the basic set. Thus x_2 , which is artificial, will be dropped from the system in (6.16). The basic solution is still not feasible because x_3 is artificial; thus replacing x_3 by $-x'_3$, we have

$$\begin{array}{rcll} (-z) & & & 4x_5 + 3x_6 = -6 \\ x_1 & & & - 1x_5 + 3x_6 = 7 \\ & + x_4 - 2x_5 + x_6 & & = 1 \\ & + x'_3 & + 2x_5 - 1x_6 & = -1. \end{array} \quad (6.17)$$

After pivoting on $-1x_6$, the next iteration results in an optimal solution, as shown in the tableau below.

$$\begin{array}{rcll} (-z) & + 3x'_3 & + 10x_5 & = -9 \\ x_1 & + 3x'_3 & + 5x_5 & = 4 \\ & + x'_3 + x_4 & & = 0 \\ & - x'_3 & - x_5 + x_6 & = 1. \end{array} \quad (6.18)$$

According to our rules, since artificial x'_3 is nonbasic, terms in x'_3 are dropped from the problem.

As we have pointed out, many problems have a feasible solution to the dual readily available. For example, if the equations are weighted by the multipliers of a

previously optimized system having the same matrix of coefficients, a_{ij} and \bar{c}_j , and if the equations are weighted by the multipliers and their sum is subtracted from the z -equation, the coefficients, c'_j , of the transformed z -equation are nonnegative. After augmentation of the new system with artificial variables, the system is

- in canonical form with respect to the artificial basis, and
- its relative cost factors, c'_j , are nonnegative.

Hence, optimizing via the Dual-Simplex Algorithm provides an optimum to the primal system without the usual Phase I.

Even in cases where the minimizing form has a *few* negative coefficients, it is expedient to replace each negative c'_j by $c'_j = 0$ and then to optimize by the Dual-Simplex Algorithm. This will provide a basic feasible solution to the original system (not necessarily optimal), which may then be used with the true values of c_j to initiate the usual Phase II of the Primal-Simplex process.

- ▷ **Exercise 6.10** Discuss how to recover the true values of \bar{c}_j in the case when c'_j is set to 0.
- ▷ **Exercise 6.11** Prove that no more than k iterations are required to eliminate k artificial variables from a basic set while maintaining feasibility of the dual. Note that the primal problem may be infeasible at the end. What is the dual of this exercise?

6.4 PARAMETRIC LINEAR PROGRAMS

The term *parametric linear programming* is applied to the situation where the coefficients of the objective function and/or the right-hand-side constants are allowed to vary with a parameter, say θ . In this section, we shall first examine the effect of varying the coefficients of the objective function and then examine the effect of varying the right-hand-side constants.

6.4.1 PARAMETERIZING THE OBJECTIVE FUNCTION

In this case, the objective function coefficients c_j are assumed to change simultaneously at given rates γ_j . Thus, the class of linear programs of interest is:

$$\begin{array}{ll} \text{Minimize} & (c + \theta\gamma)^T x = z(\theta) \\ \text{subject to} & Ax = b, \quad A : m \times n, \\ & x \geq 0, \end{array} \quad (6.19)$$

where $\gamma = (\gamma_1, \gamma_2, \dots, \gamma_n)^T$ are the given fixed rates of change of the objective function coefficients $c = (c_1, c_2, \dots, c_n)^T$ per unit of the scalar parameter $\theta \geq 0$.

We shall examine the behavior of (6.19) as θ varies. Without loss of generality we have assumed $\theta \geq 0$ because the case of $\theta \leq 0$ is equivalent to replacing θ by $-\bar{\theta}$.

The feasibility of problem (6.19) is clearly independent of the objective function; thus, we shall only examine the case when the problem is feasible. Let us first assume that the objective function has a finite optimum when $\theta = 0$ and the optimal basis is B . Let π be the optimal prices for a basis B when $\theta = 0$ and $\hat{\pi}(\theta) = \pi + \theta\rho$ for some $\theta > 0$. Then, for a given value of θ , we can determine π and ρ from

$$B^T \hat{\pi}(\theta) = B^T(\pi + \theta\rho) = c_B + \theta\gamma_B, \quad (6.20)$$

i.e., π and ρ are solutions to

$$B^T \pi = c_B \quad \text{and} \quad B^T \rho = \gamma_B. \quad (6.21)$$

Next we determine the reduced costs $\hat{\sigma}(\theta)$ with respect to B from

$$\begin{aligned} \hat{\sigma}(\theta) &= c_N + \theta\gamma_N - N^T \hat{\pi}(\theta) \\ &= c_N - N^T \pi + \theta(\gamma_N - N^T \rho). \end{aligned} \quad (6.22)$$

We are interested in the range of $\theta \geq 0$ for which B is an optimal basis. In particular by the assumed optimality of B for $\theta = 0$, we have

$$\bar{c}_N = \hat{\sigma}(0) = c_N - N^T \pi \geq 0. \quad (6.23)$$

The range of $\theta \geq 0$ for which B is an optimal basis is the range for which $\hat{\sigma}(\theta) \geq 0$. From Equation (6.22), we require

$$\hat{\sigma}(\theta) = \bar{c}_N + \theta\bar{\gamma}_N \geq 0, \quad (6.24)$$

where

$$\bar{\gamma}_N = \gamma_N - N^T \rho. \quad (6.25)$$

Then from (6.24), the basis B remains optimal for θ satisfying the vector relation

$$-\theta\bar{\gamma}_N \leq \bar{c}_N, \quad \text{where} \quad \bar{c}_N \geq 0. \quad (6.26)$$

Two cases arise in determining the range of θ that maintains optimality:

1. If $\bar{\gamma}_N \geq 0$ then the basis B is optimal for all values of $\theta \geq 0$.
2. If one or more components $\bar{\gamma}_j < 0$ for $j \in \mathcal{N}$, then the basis B is optimal for all θ in the range $0 \leq \theta \leq \theta_1$ where

$$\theta_1 = \min_{\{j \in \mathcal{N} | \bar{\gamma}_j < 0\}} \frac{\bar{c}_j}{-\bar{\gamma}_j}. \quad (6.27)$$

For $\theta = \theta_1 + \epsilon$, where $\epsilon > 0$, the basis B will no longer be optimal, and one or more nonbasic variables will become candidates for entering the basis.

For the rest of this discussion, assume that we are solving the problem in the canonical form of the Simplex Method. If $\theta > \theta_1$, then one or more nonbasic variables will be candidates for entering the basis B . It is clear that if any candidate variable x_s has coefficients $\bar{a}_{is} \leq 0$ for $i = 1, \dots, m$, the problem is unbounded for all $\theta > \theta_1$ because we have found a ray along which the objective function can be made arbitrarily small. When this happens, we terminate with the class of solutions $x_B = b - \alpha \bar{A}_{\bullet s}$, $x_s = \alpha \geq 0$, $x_j = 0$ for $j \in \mathcal{N}$ and $j \neq s$ where the corresponding $z \rightarrow -\infty$ as $\alpha \rightarrow \infty$.

LEMMA 6.2 (Nonnegativity of Relative Cost Factors) *When there is only one eligible candidate x_s to enter the basis at $\theta = \theta_1 + \epsilon > 0$, $\epsilon > 0$, and there are one or more coefficients $\bar{a}_{is} > 0$ then the relative cost factors with respect to the new basis at $\theta = \theta_1$ defined by (6.27) are nonnegative and remain nonnegative for some range of $\theta > \theta_1$.*

Proof. It is evident that $\bar{c}_s + \theta_1 \bar{\gamma}_s = 0$ and $\bar{c}_j + \theta_1 \bar{\gamma}_j > 0$ for $j \in \mathcal{N}$ and $j \neq s$ because we are assuming only one candidate x_s . If we pivot on \bar{a}_{rs} assuming $\theta \geq \theta_1$, we get the new reduced costs for column s equal to zero and, for the remaining nonbasic columns we get

$$\hat{c}_j = \bar{c}_j + \theta \bar{\gamma}_j - \frac{\bar{a}_{rj}}{\bar{a}_{rs}} (\bar{c}_s + \theta \bar{\gamma}_s), \quad j \in \mathcal{N}, \quad j \neq s. \quad (6.28)$$

Noting that $\bar{c}_s + \theta_1 \bar{\gamma}_s = 0$, we rewrite Equation (6.28) as

$$(\bar{c}_j + \theta_1 \bar{\gamma}_j) + (\theta - \theta_1) \left(\bar{\gamma}_j - \frac{\bar{a}_{rj}}{\bar{a}_{rs}} \bar{\gamma}_s \right). \quad (6.29)$$

Because, by assumption, $\bar{c}_j + \theta_1 \bar{\gamma}_j > 0$ for $j \in \mathcal{N}$, $j \neq s$, the first term dominates the second term for some range $\theta > \theta_1$. ■

▷ **Exercise 6.12** Extend Lemma 6.2 to the case when there is more than one candidate for entering variable at $\theta = \theta_1$. Resolve the tie by solving a restricted linear program involving only the basic columns and the tied columns with new objective coefficients $(\theta - \theta_1) \bar{\gamma}_j$. Let \bar{B} be the optimal basis of the restricted problem. Show that the new relative cost factors with respect to \bar{B} are nonnegative for some range of $\theta > \theta_1$ for all $j = 1, \dots, n$.

THEOREM 6.3 (When Minimizing, the Optimal Value is a Continuous Piecewise Linear Concave Function) *The optimal value of the parametric objective function for the linear program (6.19) is a continuous piecewise linear concave function of the parameter θ .*

Proof. Let $0 \leq \theta \leq \theta^*$ be the range of values for θ for which a finite minimum exists for the objective function. As θ increases from 0 to θ_1 , as defined by (6.27), the basis does not change and thus the basic feasible solution ($x_B, x_N = 0$) does not change. Hence the objective function value changes linearly with θ in this

range. Similarly, for $\theta_1 < \theta \leq \theta_2$ there is a new basis and the objective function also changes linearly with θ until the next point θ_2 , where the next optimal basis change is reached. However, under nondegeneracy, the slope $\gamma^T x$ with respect to θ is different beyond θ_1 because the optimal solution x^2 in the new interval is not the same as the optimal solution x^1 in the previous interval. (Under degeneracy, it is possible that $x^2 = x^1$, implying the slopes are the same.) Thus, in general, the function is clearly piecewise linear and continuous.

Let θ' and θ'' be any two points in the interval $0 \leq \theta \leq \theta^*$ and let x' and x'' be the corresponding feasible optimal solutions to (6.19) with optimal objective function values $z(\theta')$ and $z(\theta'')$, respectively. Pick any λ in the range $0 \leq \lambda \leq 1$ and define $\theta^\lambda = \lambda\theta' + (1 - \lambda)\theta''$. Let the optimal solution at θ^λ be denoted x^λ and the optimal objective value by $z(\theta^\lambda)$. Then

$$\begin{aligned} z(\theta^\lambda) &= (c + \theta^\lambda \gamma)^T x^\lambda = \lambda(c + \theta' \gamma)^T x^\lambda + (1 - \lambda)(c + \theta'' \gamma)^T x^\lambda \\ &\geq \lambda z(\theta') + (1 - \lambda)z(\theta'') \end{aligned}$$

where the last line follows from the optimality of $z(\theta')$ and $z(\theta'')$. This proves that the function is concave and we have already shown that it is piecewise linear continuous. ■

COROLLARY 6.4 (When Maximizing, the Optimal Value Is a Continuous Piecewise Linear Convex Function) *If the objective function of the parametric linear program defined by (6.19) is maximized instead of minimized, then the optimal value is a continuous piecewise linear convex function of the parameter θ .*

COROLLARY 6.5 (When Minimizing, the Optimal Value Is a Continuous Piecewise Linear Concave Function) *If the objective function of the parametric linear program defined by (6.19) is of the form $z(\theta) = (1 - \theta)c^T x + \theta\gamma^T x$, the optimal value is a continuous piecewise linear concave function of the parameter θ .*

▷ **Exercise 6.13** Prove Corollaries 6.4 and 6.5.

▷ **Exercise 6.14** Construct a linear program with a parametric objective function $(c + \theta\gamma)^T x$ such that the objective has a finite minimum for all $-1 \leq \theta \leq 1$, but at $\theta = 1 + \epsilon$ and at $\theta = -1 - \epsilon$ the objective function $z \rightarrow -\infty$ for a class of feasible solutions where $\epsilon > 0$ is arbitrarily small. If so, also prove that $z \rightarrow -\infty$ for all $\epsilon > 0$ arbitrarily large.

▷ **Exercise 6.15** Apply parametric programming to revise the optimal solution to a linear program when the cost coefficients x are replaced by new cost coefficients.

In practice, parametric linear programming problems are solved using two objective functions, $z = c^T x$ and $\gamma^T x$. Pricing operations are carried out on both functions. The relative cost factors $\bar{c} + \theta\bar{\gamma}$ are never explicitly computed but x_s is determined by (6.27) where \bar{c}_j and $\bar{\gamma}_j$ are determined from the simplex multipliers π and ρ , which are the solutions to $B^T \pi = c_B$ and $B^T \rho = \gamma_B$ for the current updated basis B .

6.4.2 PARAMETERIZING THE RIGHT-HAND SIDE

In this case the right-hand side constants b_i are assumed to change at given rates β_j . Thus, the class of linear programs of interest is:

$$\begin{aligned} & \text{Minimize} && c^T x = z(\phi) \\ & \text{subject to} && Ax = b + \phi\beta, \quad A: m \times n, \\ & && x \geq 0, \end{aligned} \quad (6.30)$$

where $\beta = (\beta_1, \beta_2, \dots, \beta_m)^T$ are the given fixed rates of changes to the right-hand side per unit of the scalar parameter ϕ . Once again, without loss of generality, we restrict $\phi \geq 0$ because looking at $\phi \leq 0$ is equivalent to replacing ϕ by $-\bar{\phi}$. It can easily be verified that if β does not lie in the range space of the coefficient matrix A , the linear program is feasible only for $\phi = 0$.

▷ **Exercise 6.16** If $Ax = b, x \geq 0$ is feasible and $Ax = \beta, x \geq 0$ is also feasible, show that $Ax = b + \phi\beta$ is feasible for all choices of $\phi \geq 0$. Also show that if a constraint is redundant for some $\phi > 0$ then it is redundant for all values of $\phi > 0$.

Assume that the linear program is feasible for both b and β . Then the optimal basis B at $\phi = 0$ stays feasible for the range of ϕ for some range $\phi \geq 0$, namely, ϕ satisfying:

$$B^{-1}b + \phi B^{-1}\beta \geq 0. \quad (6.31)$$

Therefore, letting $\bar{b} = B^{-1}b$ and $\bar{\beta} = B^{-1}\beta$, the basis B remains primal feasible for all ϕ satisfying the vector relation:

$$-\phi\bar{\beta} \leq \bar{b}. \quad (6.32)$$

▷ **Exercise 6.17** Show that if the optimal basic feasible is nondegenerate then B stays feasible and optimal for some range $[0, \phi_1]$ where $\phi_1 > 0$.

Two cases arise in determining the range of ϕ that maintains feasibility:

1. If $\bar{\beta} \geq 0$ then the optimal basis B results in a feasible solution for all values of $\phi \geq 0$. The values of the basic variables and the objective are the only ones that change; the basic set of columns remain unchanged.
2. If, on the other hand, one or more components $\bar{\beta}_i < 0$, then the range of ϕ that maintains feasibility is $0 \leq \phi \leq \phi_1$, where

$$\phi_1 = \min_{\{i | i \in \mathcal{B}, \bar{\beta}_i < 0\}} \frac{\bar{b}_i}{-\bar{\beta}_i}. \quad (6.33)$$

At $\phi = \phi_1 + \epsilon$, where $\epsilon > 0$, the problem is primal-infeasible but is still dual-feasible since $\bar{c}_N \geq 0$ does not depend on the right-hand side b .

The analysis of the parametric right-hand-side case is similar to the parametric cost function case discussed in the previous section, except that now we apply the Dual-Simplex Method of Section 6.3 to bring back primal feasibility.

- ▷ **Exercise 6.18** Show that a linear program with a parametric right-hand side whose objective is being minimized is a dual of a linear program with a parametric cost function whose objective is being maximized and vice versa.

THEOREM 6.6 (When Minimizing, the Optimal Value Is a Continuous Piecewise Linear Convex Function) *The optimal value of the objective function $z(\phi)$ for the linear program (6.30) is a continuous piecewise linear convex function of the parameter ϕ .*

- ▷ **Exercise 6.19** Prove Theorem 6.6 using either Corollary 6.5 and duality or a proof similar to Theorem 6.3.
- ▷ **Exercise 6.20** Develop the theory for the parametric right-hand side analogous to that of the parametric cost case.

6.5 SELF-DUAL PARAMETRIC ALGORITHM

The *self-dual parametric algorithm*, sometimes referred to as the *criss-cross method*, is applied to a linear program where both the objective function and the right-hand side are functions of a parameter θ . One application of this method is in the case when neither the basic solution nor its complementary dual solution is feasible. It is then a simple matter to increase all the negative \bar{b}_i and \bar{c}_j to nonnegative values by adding some constant θ to all of them. (A variant is to add a vector $\phi\beta$ to the right-hand side and $\theta\gamma$ to the objective coefficients.)

The modified problem is now optimal. Next we will consider ways to maintain the feasibility of the primal and dual systems as the constants and cost coefficients are gradually changed linearly as a function of θ toward their original values. Depending on whether the basic solution of the dual or the primal first becomes infeasible as θ moves toward zero, the Primal-Simplex or the Dual-Simplex pivot choice criterion is employed. The method is illustrated in the following example.

Example 6.3 (Illustration of a Self-Dual Parametric Algorithm) In the following canonical system, the original problem is obtained by setting $\theta = 0$; the associated basic solutions are infeasible for both the primal and dual.

$$\begin{array}{rcl}
 (-z) & & 8x_4 + (\theta - 4)x_5 = 0 \\
 x_1 & + & x_4 + x_5 = 8 \\
 x_2 & - & 2x_4 + \mathbf{1}x_5 = -1 + \theta \\
 x_3 + 3x_4 & - & 2x_5 = -1 + \theta.
 \end{array} \tag{6.34}$$

On the other hand, if $\theta \geq 4$, the associated solutions are both feasible. If we start with $\theta = 5$, say, and then let θ approach zero, the associated solutions will remain feasible down to the critical value $\theta = 4$. Just below $\theta = 4$, the primal solution still remains feasible, but the dual solution becomes infeasible since $\bar{c}_3 = \theta - 4$ is negative. Hence, for θ less than 4 but “very close” to it, we use the *Primal Simplex Algorithm*, introducing x_5 while maintaining the feasibility of both systems. The variable to be dropped is determined from the minimum of the ratios \bar{b}_i/\bar{a}_{is} for \bar{a}_{is} positive:

$$\frac{\bar{b}_1}{\bar{a}_{15}} = 8, \quad \frac{\bar{b}_2}{\bar{a}_{25}} = -1 + \theta$$

Since in the neighborhood of $\theta = 4$ the second ratio is minimal, x_2 is to be dropped from the basic set at the next iteration. The new canonical system, after pivoting on $\bar{a}_{25} = 1$, is

$$\begin{array}{rcccccl} (-z) & & + (4 - \theta)x_2 & + 2\theta x_4 & = (4 - \theta)(-1 + \theta) & \\ & x_1 & -1x_2 & + 3x_4 & = 9 - \theta & \\ & & x_2 & - 2x_4 + x_5 & = -1 + \theta & \\ & & 2x_2 + x_3 - \mathbf{1}x_4 & & = -3 + 3\theta & \end{array} \quad (6.35)$$

which remains feasible for all θ in the range $1 \leq \theta \leq 4$. Below the critical value $\theta = 1$, the primal basic solution becomes infeasible. For θ less than 1 but very close to it, basic variable $x_3 < 0$; therefore we use the *Dual-Simplex Algorithm* to drop x_3 as a basic variable. The variable to be introduced is given by the minimum of the ratios $\bar{c}_j/(-\bar{a}_{3j})$ for j such that $\bar{a}_{3j} < 0$; in this case, the only variable in row 3 with a negative coefficient is x_4 . Pivoting on $\bar{a}_{34} = -1$, we obtain

$$\begin{array}{rcccccl} (-z) & & + (4 + 3\theta)x_2 + 2\theta x_3 & & = (4 - \theta)(-1 + \theta) + 2\theta(-3 + 3\theta) & \\ & x_1 & + 5x_2 + 3x_3 & & = 0 + 8\theta & \\ & & -3x_2 - 2x_3 & + x_5 & = 5 - 5\theta & \\ & & -2x_2 - x_3 + x_4 & & = 3 - 3\theta & \end{array} \quad (6.36)$$

which is feasible for both the primal and dual systems for $0 \leq \theta \leq 1$. Hence, the *optimal* solution to the original problem is obtained by setting $\theta = 0$.

In general, it is not necessary to add the same parameter, θ , to all of the negative constants, \bar{b}_i and \bar{c}_j , as was done in (6.34). Several different parameters could be added and each allowed to tend toward zero according to some prespecified rules as to how they move relative to each other. Either way, the net result is the successive application of either the *Primal-Simplex* or *Dual-Simplex* rules to change the basis.

In order to show that such a process will terminate in a finite number of steps, we now prove two theorems for the case of a single parameter, θ .

THEOREM 6.7 (Feasibility of Complementary Bases) *It is not possible to have the same complementary bases feasible in the primal and dual for two values $\theta_1 < \theta_2$, unless the pair is also feasible for all values in the range $\theta_1 \leq \theta \leq \theta_2$.*

Proof. Note that for any fixed primal basis the values of the primal-basic or dual-basic variables are linear functions of θ so that, clearly, when a variable is nonnegative for both $\theta = \theta_1$ and $\theta = \theta_2$, then it is nonnegative throughout the interval $\theta_1 \leq \theta \leq \theta_2$. ■

THEOREM 6.8 (Finite Termination) *If each change in basis is accompanied by a positive decrease in θ , there can only be a finite number of iterations.*

Proof. Suppose that as θ decreases there is a change of feasible basis from B^0 to B^1 at some critical value $\theta = \theta_1$. After the change in basis, suppose B^1 remains feasible for all $\theta_2 < \theta \leq \theta_1$. The basis B^1 cannot be a repeat of some earlier basis that was feasible at $\hat{\theta} > \theta$ because $B^1 = \hat{B}$ by Theorem 6.7 would have been feasible for all $\theta_2 < \theta \leq \hat{\theta}$ and therefore θ_1 would not have been a critical value; this is a contradiction. ■

THEOREM 6.9 (Positive Decrease With Only One Degeneracy) *If there is only one degeneracy in the primal and dual solutions before and after pivoting at a critical value of $\theta = \theta_0$, then after the pivot there will be a positive decrease in θ .*

Proof. Since $\theta = \theta_0$ is critical, there is at least one degeneracy in either the primal or dual systems. If a degeneracy occurs in the primal we assume there are no others in the primal and dual systems and, similarly, if a degeneracy occurs in the dual we assume there are no others in the primal and dual systems. We now assume dual degeneracy occurs at $\theta = \theta_0$. If we prove the theorem for the case of dual degeneracy, the proof for a primal degeneracy will follow by duality. Suppose that dual degeneracy occurs at \bar{c}_s at a critical value of $\theta = \theta_0$, and for $\theta_o \leq \theta \leq \theta_o + \delta$ we have $\bar{c}_s = \epsilon = d(\theta - \theta_0) > 0$, where $d > 0$. All other nonbasic \bar{c}_j can now be expressed linearly in ϵ , by substituting $\theta = \theta_0 + \epsilon/d$; thus $\bar{c}_j = \alpha_j + \epsilon\beta_j$ where, by hypothesis, α_j is strictly positive for nonbasic $j \neq s$. By hypothesis, at $\epsilon = 0$, the primal solution is nondegenerate before and after x_s displaces some variable, x_{j_r} , in the basic set. Under these conditions the new values of the relative cost factors for nonbasic \hat{c}_j will be

$$\begin{aligned}\hat{c}_j &= \bar{c}_j - (\bar{a}_{rj}/\bar{a}_{rs})\epsilon = \alpha_j + [\beta_j - \bar{a}_{rj}/\bar{a}_{rs}]\epsilon, \quad j \neq s, \\ \hat{c}_{j_r} &= -(1/\bar{a}_{rs})\epsilon.\end{aligned}\tag{6.37}$$

Since α_j is positive for all updated nonbasic j , except j_r , there clearly is a range of values, $\epsilon_0 < \epsilon < 0$, for some fixed $\epsilon_0 < 0$, for which \hat{c}_j remains positive for all $j \neq j_r$. For $j \neq j_r$ it also follows that $\hat{c}_{j_r} > 0$ because $\epsilon < 0$ in this range and $\bar{a}_{rs} > 0$. ■

THEOREM 6.10 (Range of θ Over Which Feasible Solutions Exist) *If a feasible solution to the primal and dual systems exists for $\theta = 0$ and $\theta = \theta_0$, then feasible solutions exist for all θ in the interval $0 \leq \theta \leq \theta_0$.*

▷ **Exercise 6.21** Prove Theorem 6.10.

▷ **Exercise 6.22** Prove that Theorem 6.10 also implies that the solution set (x, π) generated by all vectors of constant terms, b_i , and cost terms, c_j , for which both the primal and dual problems remain feasible simultaneously, is a convex polyhedron.

6.6 THE PRIMAL-DUAL ALGORITHM

Like the Dual-Simplex method (see Section 6.3), the Primal-Dual Algorithm starts with a feasible solution to the dual already at hand. It iteratively *decreases the infeasibility form* of the primal and improves the feasible dual solution in such a manner that, if a feasible basic solution is finally obtained, it will already be optimal.

Associated with each iteration is a subproblem with fewer columns called the *column-restricted primal*, which is optimized by a primal method (for example, the Phase I Simplex Algorithm). When the optimal solution of the column-restricted primal has been obtained, the optimal dual solution to this restricted primal is used to generate an improved dual solution to the original problem. This gives rise to a *new* column-restricted primal to be optimized. After a finite number of improvements an optimal solution is obtained for the original problem, or a class of feasible dual solutions with unbounded objective is obtained, establishing that the original primal problem is infeasible.

Once again consider a linear program in standard form (6.1) and its dual (6.2). Then, as we have seen before, $y = \bar{c} = c - A^T\pi$ is a vector of dual slack variables and $v = b^T\pi$ is the value of the dual objective.

After adding a full set of artificials, $x_a = x_{n+1}, \dots, x_{n+m}$, to (6.1), we have the following usual Phase I problem:

$$\begin{aligned} 0^T x + e^T x_a &= w \quad (\min) \\ Ax + Ix_a &= b. \end{aligned} \tag{6.38}$$

Suppose an initial dual feasible solution $\pi = \pi^1$ is known in advance; let

$$\bar{c}^1 = c - A^T\pi^1 \geq 0, \quad z_1 = b^T\pi^1. \tag{6.39}$$

For convenience, assume that we have reordered the original variables so that $\bar{c}_j^1 = 0$ for $j = 1, \dots, l$, and $\bar{c}_j^1 > 0$ for $j = l + 1, \dots, n$. The problem (6.38) is then solved by the usual Phase I procedure, except we *restrict the choice of columns entering the basis to those whose $\bar{c}_j^1 = 0$* . This Phase I problem is called a *column-restricted Phase I problem*. As artificial variables drop out of the basis we drop them from the problem. At the end of Phase I, we get the simplex multipliers $\gamma = \gamma^1$ with respect to the Phase I objective and a solution $x_a = x_a^1$, $x = x^1$, and $z = z_1$.

THEOREM 6.11 (Optimality Condition) *If, at the end of Phase I, $w_1 = 0$, then the solution $x^1 \geq 0$ is an optimal feasible solution for the primal (6.38), and the solution π^1 is an optimal solution for the dual of the original system.*

Proof. By assumption $w_1 = \min w = 0$, therefore, the artificial variables all have zero values in the basic solution. After dropping them, the primal feasible solution $x = x^1 \geq 0$ has $\bar{c}_j^1 = 0$ for $x_j^1 > 0$ and $\bar{c}_j^1 \geq 0$ for $x_j^1 = 0$, which fulfills the condition of the optimality of $x = x^1$ for the original system. ■

Therefore, if $w_1 = 0$ at the end of Phase I, we terminate with an optimal solution to the original problem. On the other hand, if $w_1 > 0$, then we can get an improved

dual-feasible solution π^2 to the original system (6.1) for some $\theta > 0$ by:

$$\pi^2 = \pi^1 + \theta\gamma^1. \quad (6.40)$$

For π^2 to result in an improved dual-feasible solution, we need for some $\theta > 0$

$$\bar{c}^2 = c - A^T\pi^2 = c - A^T(\pi^1 + \theta\gamma^1) = \bar{c}^1 + \theta\bar{d}^1 \geq 0, \quad (6.41)$$

where $\bar{d}^1 = -A^T\gamma^1$.

We already have $\bar{d}_j^1 \geq 0$, for $j = 1, \dots, l$ because this is a condition of optimality of the column-restricted primal; for these j , we clearly have $\bar{c}_j^2 \geq 0$ for all $\theta \geq 0$. If all the factors, \bar{d}_j^1 , are nonnegative for $j = l + 1, \dots, n$ then, since $w_1 > 0$, and $\bar{d}_j \geq 0$ for all j , (6.40) and (6.41) constitute a class of feasible solutions to the dual objective of the original problem, whose value,

$$v = z_1 + \theta w_1, \quad (6.42)$$

tends to $+\infty$ with increasing θ , implying no feasible solution to the primal exists.

If one or more of the \bar{d}_j^1 , for $j = l + 1, \dots, n$ are negative then for dual feasibility we pick θ to satisfy:

$$\theta = \theta_1 = \min_{\{j|\bar{d}_j^1 < 0\}} \frac{\bar{c}_j^1}{-\bar{d}_j^1} > 0, \quad (6.43)$$

since $\bar{c}_j^1 > 0$ for $j = l + 1, \dots, n$. The resulting $\pi^2 = \pi^1 + \theta_1\bar{d}^1$ and corresponding $\bar{c}^2 = c - A^T\pi^2$ are then dual feasible; furthermore, $z_2 = z_1 + \theta_1 w_1 > z_1$ since $w_1 > 0$.

▷ **Exercise 6.23** Refer to Theorem 5.1 on Page 155 concerning a degenerate basic feasible solution obtained using the Simplex Algorithm. Then a restricted linear program \mathcal{R} is optimized, corresponding to rows where $\bar{b}_i = 0$. Develop an analogy between this restricted-row problem and the restricted-column problem of this section.

▷ **Exercise 6.24** Let θ_1 be given by (6.43) and $\bar{c}^2 = \bar{c}^1 + \theta\bar{d}^1$. Letting j_o , where $1 \leq j_o \leq l$, correspond to a basic column, prove $\bar{c}_{j_o}^2 = 0$. Prove, for $j = l + 1, \dots, n$, there is at least one $\bar{c}_j^2 = 0$.

All columns $j = 1, \dots, n$ whose $\bar{c}_j^2 = 0$ and all artificial columns that have not been dropped constitute the columns associated with the new restricted primal problem. Since there is at least one $\bar{c}_j^2 = 0$ and $\bar{d}_j^1 < 0$ for $j = l + 1, \dots, n$ at the end of the previous restricted problem (see Exercise 6.24), we know there are new candidates for entering the basis in the Phase I process applied to the new column-restricted primal and permitting the iterative process to continue.

▷ **Exercise 6.25** Show that the optimal basis of the previous restricted problem can be used as the starting basis for the new restricted problem.

- ▷ **Exercise 6.26** In the Primal-Simplex Algorithm, the pivot steps can sometimes result in no improvement when the basic solutions are degenerate. In the Primal-Dual Algorithm there is improvement of the dual and *strict* decrease of primal infeasibility on *every* iteration. Does degeneracy play a role? If so, where?

At first glance it may appear that we might not have been able to initiate the algorithm if *all* \bar{c}_j^1 were positive for $j = 1, \dots, n$. However, in this case, we view the basic set of artificials as the full set of variables of the restricted primal. Clearly, the artificial columns are the optimal basic columns and their optimal multipliers are $\gamma^1 = (1, 1, \dots, 1)$. The algorithm can then be initiated after finding an improved dual solution by computing θ by (6.43).

- ▷ **Exercise 6.27** Suppose that no π^1 was given but the original problem satisfied $c_j \geq 0$ for all j . How would you initiate the Primal-Dual Algorithm?

Example 6.4 (Illustration of Primal-Dual Algorithm) We illustrate the Primal-Dual Algorithm on the problem of finding $x_1 \geq 0, x_2 \geq 0, \dots, x_5 \geq 0, \min z$, and artificial variables, $x_6 = x_7 = x_8 = w = 0$, satisfying

$$\begin{array}{rcl}
 (-w) & & + x_6 + x_7 + x_8 = 0 \\
 (-z) + 2x_1 + 5x_2 + 10x_3 + 4x_4 + 28x_5 & & = 0 \\
 x_1 + 2x_2 + 2x_3 + 0x_4 - 4x_5 + x_6 & & = 8 \\
 x_1 - 4x_2 + 4x_3 - 4x_4 + 4x_5 & + x_7 & = 12 \\
 + x_2 + 0x_3 + 2x_4 + 2x_5 & & + x_8 = 2.
 \end{array} \tag{6.44}$$

Since all the c_j s are positive, a feasible dual solution is $\pi = (0, 0, 0)^T$ and therefore $\bar{c}^o = c$. The initial simplex multipliers with respect to the Phase I objective are $\gamma^o = (1, 1, 1)^T$. This γ^o can then be used to generate the w -equation in terms of the original variables as the difference between the first equation and the sum of the last three equations:

$$-2x_1 + x_2 - 6x_3 + 2x_4 - 2x_5 = w - 22. \tag{6.45}$$

Letting \bar{d}_j^o be the coefficients of x_j in this new expression for w , we determine the largest number, $\theta = \theta_o$, such that $\bar{c}^1 = \bar{c}^o + \theta_o \bar{d}^o \geq 0$ has all nonnegative components. In this case, $\theta = \theta_o = 1$. Thus, the new z -equation (6.46) is obtained by adding the z -equation in (6.44) to the w -equation in (6.45); where we have dropped the w variable because all we are really doing is adding to the z -equation a linear combination of the original equations, which have no artificial variables:

$$0x_1 + 6x_2 + 4x_3 + 6x_4 + 30x_5 = z - 22. \tag{6.46}$$

Note that in (6.46), x_6, x_7 , and x_8 are already basic, and, besides these, only x_1 has a relative cost factor of zero. Thus, the first *column-restricted primal* is obtained by choosing variables x_1 and artificials x_6, x_7 , and x_8 that have not been dropped; see columns with \uparrow

in (6.47).

Iteration 1: *Initiate First Column-Restricted Primal:* $(x_6, x_7, x_8; x_1)$

$$\begin{array}{rccccccr}
 (-w) - & 2x_1 & + & x_2 & - & 6x_3 & + & 2x_4 & - & 2x_5 & & = & -22 \\
 & \mathbf{1x_1} & + & 2x_2 & + & 2x_3 & + & 0x_4 & - & 4x_5 & + & x_6 & = & 8 \\
 & & x_1 & - & 4x_2 & + & 4x_3 & - & 4x_4 & + & 4x_5 & & + & x_7 & = & 12 \\
 & & & + & x_2 & + & 0x_3 & + & 2x_4 & + & 2x_5 & & & + & x_8 & = & 2
 \end{array} \tag{6.47}$$

.....

$$\begin{array}{ccccccc}
 & \uparrow & & \uparrow & \uparrow & \uparrow & \\
 & 0x_1 & + & 6x_2 & + & 4x_3 & + & 6x_4 & + & 30x_5 & * & * & * & = & z - 22
 \end{array}$$

The only variable x_j in the column-restricted primal (i.e., those with \uparrow) with negative \bar{d}_j^0 in the w -equation is $\bar{d}_1^0 = -2$. Pivoting on $1x_1$ in the second equation, x_6 drops from the basic set and is dropped from the problem since it is artificial; the updated w -equation then becomes:

$$(-w) + 0x_1 + 5x_2 - 2x_3 + 2x_4 - 10x_5 = -6. \tag{6.48}$$

Thus an optimal solution to the column-restricted problem $(x_6, x_7, x_8; x_1)$ is obtained in one iteration because $\bar{d}_j^1 \geq 0$ for the restricted j .

We are now ready to set up the second column-restricted primal. To do so we adjust the z -equation by determining the largest value of $\theta = \theta_1$ such that $\bar{c}^2 = \bar{c}^1 + \theta\bar{d}^1 \geq 0$, where \bar{c}^1 are the coefficients of x_j in the z -equation in (6.47) and \bar{d}_j^1 are the coefficients of x_j in the w -equation in (6.48) respectively. Now $\theta_1 = 2$, so that the new z -equation, which again does not contain w because all we are really doing is adding to the z -equation a linear combination of the original equations that have no artificial, is:

$$0x_1 + 16x_2 + 0x_3 + 10x_4 + 10x_5 = z - 34. \tag{6.49}$$

Since $\bar{c}_3 = 0$ in this equation, the variables of the new restricted primal are the variables x_1, x_3 , and the artificials that have not been dropped, x_7 and x_8 . The basic variables are x_1, x_7 , and x_8 . The variable x_3 is now admissible as an incoming nonbasic variable in the next column-restricted primal.

Iteration 2: *Initiate Second Column-Restricted Primal:* $(x_1, x_7, x_6; x_3)$

$$\begin{array}{rccccccr}
 (-w) & & 5x_2 & - & 2x_3 & + & 2x_4 & - & 10x_5 & & = & -6 \\
 x_1 & + & 2x_2 & + & 2x_3 & + & 0x_4 & - & 4x_5 & & = & 8 \\
 & & - & 6x_2 & + & \mathbf{2x_3} & - & 4x_4 & + & 8x_5 & + & x_7 & = & 4 \\
 & & + & x_2 & + & 0x_3 & + & 2x_4 & + & 2x_5 & & + & x_8 & = & 2
 \end{array} \tag{6.50}$$

.....

$$\begin{array}{ccccccc}
 & \uparrow & & \uparrow & & \uparrow & \uparrow \\
 & 0x_1 & + & 16x_2 & + & 0x_3 & + & 10x_4 & + & 10x_5 & * & * & = & z - 34
 \end{array}$$

The only negative \bar{d}_j^1 in the w -equation for the column-restricted primal is $\bar{d}_3^1 = -2$. Pivoting on $2x_3$ in the third equation, x_7 drops from the basic set and is dropped from the problem since it is artificial; the updated w -equation becomes:

$$(-w) + 0x_1 - 1x_2 + 0x_3 - 2x_4 - 2x_5 = -2. \tag{6.51}$$

Thus an optimal solution to the column-restricted problem $(x_1, x_7, x_8; x_3)$ is again obtained in one iteration because $\bar{d}_j^2 \geq 0$ for the restricted j .

We are now ready to set up the third column-restricted primal. To do so we adjust the z -equation by determining the largest value of $\theta = \theta_2$ such that $\bar{c}^3 = \bar{c}^2 + \theta_2 \bar{d}^2 \geq 0$, where \bar{c}^2 are the coefficients of x_j in the z -equation in (6.50) and \bar{d}_j^2 are the coefficients of x_j in the w -equation in (6.51) respectively. Now $\theta_2 = 5$, so that the new z -equation, which again does not contain w because all we are really doing is adding to the z -equation a linear combination of the original equations that have no artificial, is:

$$0x_1 + 11x_2 + 0x_3 + 0x_4 + 0x_5 = z - 44. \tag{6.52}$$

Note that this time we have generated an extra zero in the z -equation because there was a tie in columns 4 and 5 in generating $\theta_2 = 5$. Therefore the corresponding column-restricted primal is $(x_1, x_3, x_8; x_4, x_5)$, since both x_4 and x_5 have zero cost factors. Notice that, in this example, except for x_2 , all the original variables now belong to the column-restricted primal.

Iteration 3: *Initiate Third Column-Restricted Primal: $(x_1, x_3, x_8; x_4, x_5)$*

$$\begin{array}{rccccrcr}
 (-w) & & - & 1x_2 & & - & 2x_4 & - & 2x_5 & & = & -2 \\
 & x_1 & + & 8x_2 & & + & 4x_4 & - & 12x_5 & & = & 4 \\
 & & - & 3x_2 & + & x_3 & - & 2x_4 & + & 4x_5 & & = & 2 \\
 & & + & x_2 & & + & 2x_4 & + & 2x_5 & + & x_8 & = & 2
 \end{array} \tag{6.53}$$

.....

$$\begin{array}{cccccc}
 \uparrow & & \uparrow & \uparrow & \uparrow & \uparrow \\
 0x_1 + 11x_2 + 0x_3 + 0x_4 + 0x_5 & * & = & z - 44
 \end{array}$$

To minimize w for the new column-restricted primal, we now introduce x_4 into the basic set by pivoting on $2x_4$ in the fourth equation, dropping x_8 from the basic set and from the problem since it is artificial, and obtaining the system:

Iteration 4: (*Optimal*)

$$\begin{array}{rccccrcr}
 (-w) + 0x_1 + & 0x_2 + & 0x_3 + & 0x_4 + & 0x_5 = & 0 \\
 & x_1 + & 6x_2 & & - & 16x_5 = & 0 \\
 & & - & 2x_2 + & x_3 & & 6x_5 = & 4 \\
 & & + & 0.5x_2 & & + & x_4 + & x_5 = & 1
 \end{array} \tag{6.54}$$

.....

$$\begin{array}{rcc}
 + & 11x_2 & = & z - 44
 \end{array}$$

whose associated solution is $(0, 0, 4, 1, 0)$ and $w = 0$. Since $w = 0$ it means that the values of $x_j \geq 0$ constitute a feasible solution to the original *unrestricted* problem. Since the x_j basic in the *column-restricted* problem corresponds to $\bar{c}_j = 0$ and because $\bar{c}_j \geq 0$ for all j has been maintained throughout, it follows that this basic solution is feasible and optimal for the original problem. The minimum value of $z = 44$ is obtained from the z -equation in (6.54).

Algorithm 6.1 (Primal-Dual Algorithm at Iteration t) The steps are repeated for $t = 1, 2, \dots$

1. *Minimizing Infeasibility of the Column-Restricted Primal.*

At the start of cycle t , it is assumed that we are given a dual feasible solution, $\bar{c}^t = c^t - A^T \pi^t \geq 0$, to the original problem generated by some $\pi = \pi^t$. We assume that the column-restricted primal problem consists of $q \geq 1$ artificials that, together

with possibly some columns of the original problem, form a basis. All columns j whose $\bar{c}_j^t = 0$ belong to the column-restricted problem, and all columns j whose $\bar{c}_j^t > 0$ are not in this restricted problem. It is further assumed that there is at least one nonbasic x_j that belongs to the column-restricted problem; if no such x_j exist, go to Step 3.

Using only the columns of the *restricted primal* problem for pivot choice, the Simplex Algorithm with a degeneracy-resolving scheme is applied to minimize w . Artificial variables are dropped from the system when they become nonbasic. During this step, which may consist of several Phase I simplex iterations, the values of the multipliers, π^t , are not modified. The simplex multipliers, γ^t , associated with the w -objective, change, of course, at each Phase I Simplex Algorithm iteration until w is “minimized;” that is, until \bar{d}_j is nonnegative for each x_j of the *column-restricted primal*.

2. Termination Condition.

- (a) If $\min w = 0$, terminate with a basic solution that is feasible and minimal for the original problem.
- (b) If $\min w > 0$, and $\bar{d}_j \geq 0$ for $j = 1, \dots, n+q$, terminate and report an infeasible original problem because no primal feasible solution exists.
- (c) Otherwise, go to Step 3 with the optimal γ^t .

3. Improving the Dual Solution (Finding a New Column-Restricted Primal).

An improved solution of the dual and a new column-restricted primal is found by using new multipliers,

$$\pi_i^{t+1} = \pi_i^t + \theta \gamma_i^t \quad \text{for } i = 1, \dots, m, \quad (6.55)$$

and $\max \theta = \theta^t$ which generate nonnegative cost factors,

$$\bar{c}_j^{t+1} = \bar{c}_j^t + \theta \bar{d}_j^t \quad \text{for } j = q+1, \dots, n+q. \quad (6.56)$$

The scalar $\max \theta = \theta^t$ is a strictly *positive* number defined by

$$\theta^t = \frac{\bar{c}_s^t}{-\bar{d}_s^t} = \min_{\bar{d}_j^t < 0} \frac{\bar{c}_j^t}{-\bar{d}_j^t} > 0. \quad (6.57)$$

The new column-restricted primal is obtained by using all the basic variables and those nonbasic variables whose cost factors, \bar{c}_j^{t+1} , are zero. In this step, at least one new variable appears in the new column-restricted primal, namely, x_s , as determined by (6.57). Note also that $\bar{d}_s < 0$, so that at least one iteration must take place before w is minimized within the new column-restricted primal.

Comment. Given a feasible solution to the dual, the Primal-Dual algorithm assumes we have at hand a routine that solves the column-restricted problem by minimizing the sum of the artificial variables. It terminates with either proving that no feasible solution exists or finding an improved primal feasible solution to the original problem. Thus the algorithm terminates after a finite number of iterations.

6.7 THE PHASE I LEAST-SQUARES ALGORITHM

In this section we describe a Phase I algorithm to obtain a strict improvement at each iteration even if degeneracy is present. It is different from other variants that try to recognize and avoid degenerate steps in the Simplex Method. As we shall see, this algorithm solves a nonnegative least-squares problem at each iteration in order to find a feasible solution.

In each iteration it obtains a better fit to the right-hand side by solving a simple two-variable least-squares subproblem to select an incoming column to augment a set of independent columns (called *basis*) in order to get a better least-squares fit to the right-hand side. Because a strict improvement is obtained at each iteration, cycling cannot occur and convergence is guaranteed. It is closely related to a number of other algorithms proposed for nonnegative least-squares and quadratic programs.

The general Phase I problem is:

$$\text{Find } x_j \geq 0, \quad \sum_{j=1}^n P_j x_j = b, \quad P_j \in \mathfrak{R}^m, \quad b \in \mathfrak{R}^m. \quad (6.58)$$

Both the Simplex Algorithm and the Least Squares (LSQ) Algorithm augment this system by a set of artificial slack (errors) variables $\epsilon^T = (\epsilon_1, \epsilon_2, \dots, \epsilon_m)$ unrestricted in sign:

$$\text{Find } x_j \geq 0, \quad \sum_{j=1}^n P_j x_j + I\epsilon = b, \quad P_j \in \mathfrak{R}^m, \quad b \in \mathfrak{R}^m. \quad (6.59)$$

Both seek to minimize the absolute values of the ϵ_i s. The Simplex Algorithm objective z is to determine $x_j \geq 0$ and ϵ_i to minimize

$$z = |\epsilon_1| + |\epsilon_2| + \dots + |\epsilon_m|. \quad (6.60)$$

The LSQ objective z is to determine $x_j \geq 0$ and ϵ_i to minimize

$$z = \frac{1}{2} (\epsilon_1^2 + \epsilon_2^2 + \dots + \epsilon_m^2). \quad (6.61)$$

Each iteration of LSQ starts with an improved solution to the right-hand side:

$$g^t = \sum_{i=1}^p P_{j_i} x_{j_i}, \quad x_{j_i} > 0 \quad \epsilon^t = b - g^t \quad (6.62)$$

which is first tested to see if it is the best least-squares fit to b . If not the best, a *strictly* improved feasible approximation is found next.

Definition (Basis): A set of independent columns $B = [P_{k_1}, P_{k_2}, \dots, P_{k_q}]$ is called a *basis*. The matrix $R = B^T B$ is called the *basis norm* and $R^{-1} = (B^T B)^{-1}$ is the *inverse of the basis norm*.

Definition (Least-Squares Approximation): $q = B\xi$ is a *basic (or least-squares) approximation* to b if B is a basis and ξ minimizes $\|b - B\xi\|^2$.

Definition (Basic Feasible Approximation): $q = B\xi$ is called a *basic feasible approximation* to b if ξ is a basic approximation that satisfies $\xi > 0$. In Phase I we seek a basic feasible approximation q such that $q = b$.

Algorithm 6.2 (The Phase I Least-Squares Algorithm) The algorithm maintains an update of the basis norm that is used to generate strictly improved basic feasible approximate solutions η . It begins by finding an initial basis, an approximate least-squares solution, and Lagrange multipliers (prices) as follows. Let $s = \operatorname{argmax}_j P_j^T b$ be the index of the incoming column P_s . If $P_s^T \epsilon \leq 0$, then *terminate* with the optimal approximate solution $x = 0$:

1. *Initialize.* The initial basis is $B = [P_s]$ and set $R = B^T B = \|P_s\|^2$. Then $R^{-1} = 1/\|P_s\|^2$; $\eta = R^{-1}(B^T b)$. The initial approximation satisfies $\eta > 0$.

2. *Begin Cycle.* Set

$$g = B\eta, \quad \epsilon = b - g. \quad (6.63)$$

3. *Determine Index s of Incoming Column.*

$$s = \operatorname{argmax}_j P_j^T \epsilon. \quad (6.64)$$

4. *Optimality Test.* If $P_s^T \epsilon \leq 0$ *terminate*. If $\epsilon = 0$, a basic feasible solution η has been found; if $\epsilon \neq 0$, report the problem infeasible, and $g = B\eta$ as the best approximate solution, and the error = $\|\epsilon\|$.

5. *Add Column to Basis.* The updated basis \bar{B} is

$$\bar{B} = [B, P_s]. \quad (6.65)$$

The updated basis norm is

$$\bar{R} = \bar{B}^T \bar{B} = \begin{pmatrix} B^T B & B^T P_s \\ P_s^T B & P_s^T P_s \end{pmatrix} = \begin{pmatrix} R & C \\ C^T & d \end{pmatrix}, \quad (6.66)$$

where $C = B^T P_s$ and $d = P_s^T P_s$. The update of the inverse of the basis norm

$$\bar{R}^{-1} = \begin{pmatrix} R^{-1} & 0 \\ 0 & 0 \end{pmatrix} + (1/d) \begin{pmatrix} \bar{C} \\ -1 \end{pmatrix} (\bar{C}^T \quad -1) \quad (6.67)$$

where $\bar{C} = R^{-1}C$ and $\bar{d} = d - \bar{C}^T C$.

6. *Obtain Next Approximation* If $\bar{\eta} > 0$. Set

$$\bar{\eta} = \bar{R}^{-1}(\bar{B}^T b) \quad (6.68)$$

If $\bar{\eta} > 0$ then $\eta \leftarrow \bar{\eta}$, $B \leftarrow \bar{B}$, $R \leftarrow \bar{R}$ and go to Step 2.

7. *Drop P_{j_r} from Basis.* If $\bar{\eta}_r = 0$, drop corresponding columns P_{j_r} from \bar{B} , adjust \bar{R}^{-1} , then go to Step 6. See the Exercises for an efficient way to adjust \bar{R}^{-1} if one or more components $\bar{\eta}_i = 0$.

8. *Delete Column from Basis.* If $\bar{\eta} \geq 0$ is not true, determine the maximum λ such that

$$\bar{\eta} = (1 - \lambda) \begin{pmatrix} \eta \\ 0 \end{pmatrix} + \lambda \bar{\eta} \geq 0. \quad (6.69)$$

Go to Step 6.

- ▷ **Exercise 6.28** Prove $s \neq j_r$, where s is the index of the incoming column (Step 3) and j_r is the index of the outgoing column (Step 7).

THEOREM 6.12 (Optimality Test) *A feasible approximation*

$$g^t = \sum_j P_j x_j^t, \quad x^t \geq 0, \quad \epsilon^t = b - g^t \quad (6.70)$$

cannot be improved if

$$P_j^T \epsilon^t \leq 0 \text{ for all } j \text{ and } P_j^T \epsilon^t = 0 \text{ for all } x_j^t > 0. \quad (6.71)$$

THEOREM 6.13 (Improvement Possible) *If the optimality test fails for some j , an improved approximation can be found.*

- ▷ **Exercise 6.29** Prove Theorems 6.12 and 6.13.

- ▷ **Exercise 6.30** If the p columns of the approximation (6.62) are not independent, show how to generate an approximation

$$g^t = \sum_{i=1}^q P_{k_i} \bar{x}_{k_i}^t, \quad \bar{x}_{k_i}^t > 0, \quad q < p, \quad (6.72)$$

where $B = [P_{k_1}, P_{k_2}, \dots, P_{k_q}]$ are a subset of independent columns of P_{j_i} . Note that g^t and $\epsilon^t = b - g^t$ are the same g^t and ϵ^t as in (6.62).

THEOREM 6.14 (Least-Squares Approximation) *The ξ that minimizes $\|b - B\xi\|^2$ can be computed by solving*

$$(B^T B)\xi = B^T b. \quad (6.73)$$

- ▷ **Exercise 6.31** Prove Theorem 6.14.

- ▷ **Exercise 6.32** If B is a q -column basis, $B^T B$ is a $q \times q$ square, symmetric nonsingular matrix.

THEOREM 6.15 (ϵ Orthogonal to Columns of B) *If $B = [P_{k_1}, P_{k_2}, \dots, P_{k_q}]$ is a basis and ξ minimizes $\|b - B\xi\|^2$, then $\epsilon = b - B\xi$ satisfies $P_{k_i}^T \epsilon = 0$ for $i = 1, \dots, q$.*

- ▷ **Exercise 6.33** Prove Theorem 6.15. Consider the problem: Find $\xi \geq 0$, ϵ , $\min z$ satisfying $B\xi + I\epsilon = b$, $z = \sum_{i=1}^q \epsilon_i^2$. Prove that the basic approximation is optimal if $\xi > 0$, $\epsilon = b - B\xi$. Prove this implies, by the optimality conditions (6.71), that $P_{k_i}^T \epsilon = 0$ for $i = 1, \dots, q$.

THEOREM 6.16 (Incoming Column Is Independent of Basic Columns) *If $\epsilon^T P_s \neq 0$, $[P_{k_1}, P_{k_2}, \dots, P_{k_q}, P_s] = \bar{B}$ is an independent set of columns.*

- ▷ **Exercise 6.34** Prove Theorem 6.16.
- ▷ **Exercise 6.35** Show how to apply \bar{R} , the inverse of the updated \bar{B} , to determine the basic approximation $\bar{g} = \bar{B}\bar{\xi}$ where $\xi = \bar{\xi}$ minimizes $\|b - \bar{B}\xi\|^2$.
- ▷ **Exercise 6.36** Given $\bar{\epsilon} = b - \bar{B}\bar{\xi}$ (see Exercise 6.35) and $\epsilon = b - B\xi$ (see Theorem 6.16), prove that $\epsilon^T P_s < 0$ implies $\|\bar{\epsilon}\|^2 < \|\epsilon\|^2$.
- ▷ **Exercise 6.37** Given $\bar{\eta}$ from Step 8 of the LSQ algorithm, prove $\hat{g} = \bar{B}\hat{\xi}$, $\hat{\epsilon} = b - \bar{B}\hat{\xi}$ is an improved feasible approximation; i.e., $\|\hat{\epsilon}\|^2 < \|\epsilon\|^2$.
- ▷ **Exercise 6.38** Show that the improvement step can only be repeated at most q times before an improved feasible approximation is obtained.
- ▷ **Exercise 6.39** Prove that the LSQ algorithm terminates after a finite number of iterations

$$t < \binom{n}{0} + \binom{n}{1} + \dots + \binom{n}{m}.$$

6.8 NOTES & SELECTED BIBLIOGRAPHY

Over the years, a number of investigations have been systematically gathering empirical data on the comparative efficiency of various proposals for choosing incoming columns in the Simplex Method, such as the steepest descent criterion. Harold Kuhn of Princeton and Philip Wolfe of RAND (independently) were particularly active doing this in the early 1960s. Based on their preliminary findings, criteria independent of the units of the activities or of the items appear to be well worth the additional effort. Computational results on some new steepest edge Simplex Algorithms by Forrest & Goldfarb [1992] show that the computational time savings can be significant; see also Bixby [2002] for additional computational results.

Lemke [1954] developed the *Dual-Simplex* Algorithm as a variant of the standard Primal-Simplex Algorithm; see also, Dantzig [1954a]. The dual problem is a linear program and therefore can be solved by the steps of the Simplex Algorithm. These steps can be

simplified by eliminating the identity submatrix. This results in an algorithm analogous to the Primal-Simplex Method with criteria for choosing the outgoing basic columns and then the incoming basic variables. For a variant, see Forrest & Goldfarb [1992] who propose a steepest-edge criterion for choosing the outgoing variable during a Dual-Simplex iteration; this has worked very well in practice, see Bixby [2002].

Gass & Saaty [1955a, 1955b, 1955c], in their papers on the *parametric objective*, studied the case of fixed constant terms and varying cost coefficients. Other variants, computationally similar, are the “Method of Leading Variables”, by E. M. L. Beale [1954a] and the “PLP (Parametric Linear Programming)”, by W. Orchard-Hays [1956], and Orchard-Hays, Cutler, & Judd [1956].

The proof of Theorem 6.9 under the assumption of a single degeneracy at a break point in the *dual* basic solution is due to Gass & Saaty [1955a, 1955b, 1955c]. The proof of Theorem 6.9 under the assumption of a single degeneracy at a break point in the *primal* basic solution is due to Orchard-Hays [1956]. Some of the early pioneering work on parametric programming was done by Manne [1956].

The parametric programming procedure described in Section 6.4 sometimes takes a lot of work to find all ranges of θ for which optimal bases are available. Consider the following example by Murty [1980], which is closely related to the one by Klee & Minty [1972] for the Simplex Method:

$$\begin{aligned} \text{Minimize} \quad & \sum_{j=1}^n (\theta - 2^{n-j}) x_j = z \\ \text{subject to} \quad & x_i + 2 \sum_{j=i+1}^n x_j \leq 4^{n-i}, \quad i = 1, \dots, m, \\ & x \geq 0. \end{aligned}$$

It can be shown that this problem has 2^n basic feasible solutions, each of which is a unique optimal solution for a suitably chosen value of the parameter θ . Therefore, any parametric programming procedure would, in the course of varying θ , enumerate 2^n solutions; clearly this is impractical for this particular contrived problem when n is large.

The *Self-Dual* algorithm of Section 6.5 is an example of a composite algorithm. For example, see Orchard-Hays [1954, 1956]. This algorithm is related to Newton’s method and has been used by Smale [1983] for investigating the worst-case complexity of the Simplex Method. For a discussion of the worst-case behavior of the Self-Dual Simplex Algorithm, see Murty [1980].

The *Primal-Dual* method was first developed by Ford & Fulkerson [1956] for transportation problems. Experiments indicate that the technique is very efficient for solving such problems. It is closely related to the work of H. Kuhn [1955], who developed a special routine for solving assignment problems, called the “Hungarian Method,” based on investigations by the Hungarian mathematician Egerváry [1931]. The Hungarian method was generalized by J. Edmonds to a method called the *blossom algorithm* for solving weighted matching problems in undirected networks (see, for example, Papadimitriou & Steiglitz [1982]).

The Primal-Dual Method of Ford & Fulkerson was later extended to the general linear program by Dantzig, Ford & Fulkerson [1956]; this is discussed in Section 6.6. These alterations of the algorithm apply when the old basis still prices out optimally in the new system and thus constitutes a feasible starting solution for the new dual. What markedly distinguishes the Ford-Fulkerson algorithm for distribution problems from the

more general case discussed here is that the former method uses a method of optimization of the restricted primal, which appears to be more efficient for distribution problems than the Simplex Method. The generalization to general linear programs, as originally published in 1956, used the Simplex Method to solve the column-restricted subproblems, because it was the the most efficient one available at that time. As pointed out in the text, any method for solving the column-restricted problems will do. It turns out that the Primal-Dual Algorithm may be viewed as a condensed sequence of simplex pivot steps. According to R. Gomory (private communication), the Primal-Dual Algorithm is a simplex variant whose number of iterations in practice is quite often fewer than that required by the Dual-Simplex Method.

The Phase I Least-Squares Algorithm described in this chapter is based on a strictly improving linear programming Phase I algorithm due to Leichner, Dantzig, & Davis [1993]. It is closely related to algorithms described, for example, by Björck, Å[1987], Lawson & Hanson [1974], Dantzig [1963], and van de Panne & Whinston [1969].

6.9 PROBLEMS

6.1 Review the following results:

- (a) Show that if a linear programming problem has a finite lower bound for some given right-hand side, then it has a finite lower bound for *any* right-hand side for which a feasible solution exists.
- (b) Suppose that a feasible linear programming problem is augmented with artificial nonnegative variables whose sum is bounded below by a constant (not necessarily zero). If z is minimized. prove that the minimum is not necessarily finite even though $\min z$ of the *original* problem is finite. However, if the artificials are bounded from above, then $\min z$ is finite or infinite depending on whether $\min z$ of the original problem is finite or infinite.

6.2 Develop the rules for the Dual-Simplex Method for a linear program in standard form with upper and lower bounds on the variables.

6.3 *Adapted from Hadley [1972].* The following procedure was suggested by Lemke for getting started with the Dual-Simplex Method. Given a linear program in standard form: $\min z = c^T x$, subject to $Ax = b$, $x \geq 0$, find m linearly independent columns from A . Let b^* be any positive linear combination of these m vectors. Next solve instead the problem with b replaced by b^* . If the minimum is $-\infty$, prove that no feasible solution to the dual exists. If a finite optimal solution is found, replace b^* with b . This gives a basic not necessarily feasible solution to the original problem and a dual feasible solution. The Dual-Simplex Method can now be used.

- (a) Why does this procedure work?
- (b) Do you think that this is a practical procedure for getting started with the Dual-Simplex Method?
- (c) Compare this procedure with the one discussed in this chapter.

- 6.4 Develop a Phase I procedure to generate a feasible starting solution to a linear program in standard form by adding a new row in such a way that it generates an obvious starting dual-feasible solution. Show that this is the dual of the Phase I procedure of adding one artificial variable.
- 6.5 Show that no basis can reoccur in the parametric linear programming procedure. What assumption is made about degeneracy?
- 6.6 Solve the following LP by the Dual-Simplex Method

$$\begin{aligned} \text{Minimize} \quad & x_1 + 4x_2 + 2x_3 \\ \text{subject to} \quad & 2x_1 + 3x_2 - 2x_3 \leq 14 \\ & x_1 - 2x_2 - 2x_3 \leq 6 \\ & -x_1 - 2x_2 \leq -10 \\ & \text{and } x_1 \geq 0, x_2 \geq 0, x_3 \geq 0. \end{aligned}$$

- 6.7 Compute

$$f(\theta) = \min\{c^T x \mid Ax = b + \theta d, x \geq 0\}$$

for all $\theta \in (-\infty, \infty)$ where $c = (1, 2, 2, 0, 0, 0)^T$ and

$$A = \begin{pmatrix} -2 & 2 & -1 & 1 & 0 & 0 \\ 1 & -1 & 3 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 2 \\ 1 \\ 10 \end{pmatrix}, \quad d = \begin{pmatrix} -1 \\ -1 \\ -1 \end{pmatrix}.$$

- 6.8 Compute

$$g(\theta) = \min\{(c + \theta d)^T x \mid Ax = b, x \geq 0\}$$

for all $\theta \in (-\infty, \infty)$ where $c = (1, 2, 0, 0, 0, 0)^T$, $d = (5, -5, -5, 0, 0, 0)^T$, and

$$A = \begin{pmatrix} 2 & -1 & 2 & 1 & 0 & 0 \\ -1 & 2 & -1 & 0 & 1 & 0 \\ -1 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

- 6.9 Solve the following LP by the self-dual parametric algorithm.

$$\begin{aligned} \text{Minimize} \quad & 2x_1 - 3x_2 = z \\ \text{subject to} \quad & x_1 + x_2 \leq 6 \\ & -2x_1 + 4x_2 \leq -1 \\ & 2x_1 - 6x_2 \leq -2 \\ & \text{and } x_1 \geq 0, x_2 \geq 0. \end{aligned}$$

- 6.10 Solve the following LP by the Primal-Dual Algorithm.

$$\begin{aligned} \text{Minimize} \quad & 3x_1 + x_2 + 3x_3 + x_4 = z \\ \text{subject to} \quad & x_1 + 2x_2 - x_3 + x_4 = 0 \\ & 2x_1 - 2x_2 + 3x_3 + 3x_4 = 9 \\ & x_1 - x_2 + 2x_3 - x_4 = 6 \\ & \text{and } x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0. \end{aligned}$$

- 6.11 Add slacks to von Neumann's primal-dual pair of LPs, $\min c^T x, Ax \geq b, x \geq 0$, and $\max b^T y, A^T y \leq c, y \geq 0$, to get:

$$\begin{array}{ll} \text{Minimize} & c^T x \\ \text{subject to} & Ax - Ix_s = b \\ & x, x_s \geq 0 \end{array} \qquad \begin{array}{ll} \text{Maximize} & b^T y \\ \text{subject to} & Iy_s + A^T y = c \\ & y_s, y \geq 0 \end{array}$$

Let α and β be complementary subsets of $\mathcal{S} = \{1, \dots, m+n\}$. In other words, $\alpha \subset \mathcal{S}$, $\beta \subset \mathcal{S}$, $\alpha \cup \beta = \mathcal{S}$, and $\alpha \cap \beta = \emptyset$. Suppose also that α has m elements (and hence β has n elements). Show that the square submatrix $(A, -I)_\alpha$ is nonsingular if and only if the square submatrix $(I, A^T)_\beta$ is nonsingular. Thus every basic solution to the primal has a corresponding complementary basic dual solution. (These solutions need not be feasible).

(Hint: Let α_1 and β_1 be complementary subsets of $1, \dots, m$ and α_2 and β_2 be complementary subsets of $m+1, \dots, m+n$ such that $\alpha = \alpha_1 \cup \alpha_2$ and $\beta = \beta_1 \cup \beta_2$).

- 6.12 *Ph.D. Comprehensive Exam, March 31, 1969, at Stanford.* We know that a linear programming problem whose variables have upper and lower bounds permit a special variant of the Simplex Method. It follows that the dual must also permit a special variant.

- State the variant.
- Characterize the class of duals.
- State the special variant for the dual problem within the above framework.

- 6.13 *Adapted from Ph.D. Comprehensive Exam, September 25, 1976, at Stanford.*

- Given two vectors $a = (a_1, a_2, \dots, a_n)^T$ and $b = (b_1, b_2, \dots, b_n)^T$ where a_i and b_i are real and $b_i > 0$. Given θ real, when is the index k that

$$\text{Maximizes}_{i=1, \dots, n} \quad \frac{a_i}{b_i}$$

the same as the index k that

$$\text{Maximizes}_{i=1, \dots, n} \quad a_i + \theta b_i.$$

- Describe a simplex-like algorithm for solving the fractional linear program

$$\begin{array}{ll} \text{Maximize} & \frac{a^T x + \alpha}{b^T x + \beta} \\ \text{subject to} & Ax = d, \\ & x \geq 0 \end{array}$$

assuming $b^T x + \beta$ is positive for all feasible x . Here a and b are $n \times 1$. *Hint:* This problem can be solved by using part (a) or it can be solved by performing a change of variable to obtain an equivalent linear program in $(n+1)$ variables where the linear program $a^T x = \max$, $Ax = d$, $x \geq 0$, is feasible and has a finite upper bound. See if you can develop the theory about how to solve the fractional program when the linear program $a^T x = \max$, $Ax = d$, $x \geq 0$, is feasible and has an infinite upper bound.

- 6.14 *Ph.D. Comprehensive Exam, September 26, 1992, at Stanford.* Consider the linear program

$$\begin{array}{ll} \text{Minimize} & c^T x = z \\ \text{subject to} & Ax = b, \quad A: m \times n, \\ & x \geq 0. \end{array}$$

- (a) Suppose that b is a linear function of a scalar parameter θ . Show, in general, that z is then a convex function of θ .
- (b) Show, however, that the value of some variable, such as x_4 in the following example, need not be either a convex or a concave function of θ .

$$\begin{array}{rccccr}
 4x_1 + 2x_2 & & + x_4 & & = z \text{ (Min)} \\
 x_1 & & - x_3 + x_4 & & = \theta \\
 x_1 + x_2 & & & & = \theta \\
 x_1 & & - x_3 & + x_5 & = 1 \\
 & x_2 & & + x_6 & = 1 \\
 x_j \geq 0 & \text{for } j = 1, \dots, 6.
 \end{array}$$

- 6.15 *Ph.D. Comprehensive Exam, September 23, 1978, at Stanford.* Consider the function $f : \mathbb{R}^1 \rightarrow \mathbb{R}^1$ defined by:

$$f(\theta) = \text{Sup}\{(c^1 + \theta c^2)^T x \mid Ax = b^1 + \theta b^2, \theta \geq 0\}$$

- (a) Describe carefully the nature of f including the cases of $c^2 = 0$ or $b^2 = 0$.
- (b) How would you generate the graph of f ?
- 6.16 *Adapted from Ph.D. Comprehensive Exam, September 1981, at Stanford.*

- (a) Suppose it is known that any feasible solution to a linear program has the property that $x_k > 0$ for some value of k (for example, if in equation p , all $a_{pj} \leq 0$ except $a_{pk} = 1$). Pivot on any nonzero coefficient of x_k in any equation to eliminate x_k from the remaining equations and the objective equation. Prove that the resulting linear program in one less equation and one less variable can be solved instead. Show how to use it to find an optimal solution to the original linear program.
- (b) If it is known in advance that a solution to a linear program cannot be optimal unless $x_k > 0$, show that this variable can be eliminated and the reduced system with one less equation and one less variable can be solved in its place. Show how to use the reduced problem to find an optimal solution to the original linear program.
- (c) If the procedure of (b), to eliminate some x_k , is applied to the linear program $\min c^T x, Ax = b, x \geq 0$, and we obtain a solution in which $x_k < 0$, this implies that either at least one feasible optimal solution of the original problem has $x_k = 0$ or the original problem is infeasible.
- (d) Outline a possible algorithm that could make practical use of the concepts of (a) and (b) to solve a linear programming problem that is a scenario, i.e., a variant of problem already solved and for which it is reasonable to assume that most of the basic variables of the optimum solution of the scenario will turn out to be the same (i.e., positive) as that of the originating problem but a few (not known in advance) will turn out not to be in the optimal solution.

- 6.17 *Ph.D. Comprehensive Exam, Fall 1985, at Stanford.* Consider the polyhedral convex set

$$\mathcal{X} = \{x \mid Ax \leq b\},$$

where A is an $m \times n$ matrix and b is an m -vector. Assume that \mathcal{X} is nonempty. Let θ denote an arbitrary real number. By definition:

$$\theta\mathcal{X} = \{\theta x \mid x \in \mathcal{X}\}.$$

(a) Show that for any *positive* real number θ :

$$\theta\mathcal{X} = \{x \mid Ax \leq \theta b\}.$$

(b) If the linear program

$$\begin{aligned} \text{Maximize} \quad & e^T y = z \\ \text{subject to} \quad & Ax + y = \theta b \\ & x \text{ free} \\ & 0 \leq y \leq e, \theta \geq 1 \end{aligned} \tag{6.74}$$

where $e = (1, 1, \dots, 1)^T \in \mathbb{R}^m$ has a feasible solution for some choice of $\theta \geq 1$, it has an optimal solution.

(c) Write the dual of the linear program (6.74).

(d) Suppose that we want to know whether the set

$$\mathcal{B}(\mathcal{X}) = \{i \mid [Ax]_i = b_i \text{ for all } x \in \mathcal{X}\}$$

is nonempty and, if so, what its elements are. Assuming exact arithmetic, show that this can be done by solving the linear program (6.74) and interpreting its solution.

6.18 Show that if no artificial variables remain in the basic set using the Primal-Dual Algorithm, the solution is optimal.

6.19 The following technique can be used to generate a starting dual feasible solution for the Primal-Dual and the Dual-Simplex Method. For convenience, assume that the variables have been relabeled so that the first m variables are basic and variables x_{m+1}, \dots, x_n are nonbasic. If a dual feasible solution is not available then construct an *augmented problem* by adding an artificial variable x_0 with cost coefficient 0 and an artificial constraint of the form

$$x_0 + x_{m+1} + \dots + x_n = M$$

where M is a very large number. Next pick $\bar{c}_s = \min \bar{c}_j < 0$; we know such a \bar{c}_s exists because the current basis is not dual feasible. Show that by pivoting on x_s we generate a dual feasible solution. Prove that if the augmented problem is primal-infeasible then so is the original problem. Suppose that after applying the Dual-Simplex Method the algorithm terminates with an optimal solution. Discuss the properties of the solution to the original problem in the cases when the optimal solution to the augmented problem contains x_0 and when it does not contain x_0 .

TRANSPORTATION PROBLEM AND VARIATIONS

The general case of the transportation problem (TP) is the minimum-cost capacitated network-flow problem

$$\begin{array}{ll}
 \text{Minimize} & c^T x \\
 \text{subject to} & Ax = b, \quad A : m \times n, \\
 & l \leq x \leq u,
 \end{array} \tag{7.1}$$

where each column $A_{\bullet j}$ has at most one positive coefficient $+1$ and at most one negative coefficient -1 . This matrix structure implies that every basis is *triangular* and that all basic solutions have integer values if the right-hand side and upper and lower bounds have integer values.

7.1 THE CLASSICAL TRANSPORTATION PROBLEM

The *classical* transportation problem, described here, can be shown to belong to the class of the general minimum-cost capacitated network-flow problem.

7.1.1 MATHEMATICAL STATEMENT

The *classical* transportation problem is as follows:

$$\begin{aligned}
 &\text{Minimize} && \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} = z \\
 &\text{subject to} && \sum_{j=1}^n x_{ij} = a_i, \quad i = 1, \dots, m, \\
 & && \sum_{i=1}^m x_{ij} = b_j, \quad j = 1, \dots, n, \\
 & && x_{ij} \geq 0, \quad i = 1, \dots, m, \quad j = 1, \dots, n,
 \end{aligned} \tag{7.2}$$

where

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j = T, \quad a_i \geq 0, \quad b_j \geq 0. \tag{7.3}$$

Note that in this chapter, the symbols m and n denote the number of sources and demand centers, respectively, and are not the symbols used to denote the number of constraints and variables for a general linear program. In this case *the number of equations is $m + n$ and the number of variables are mn* . A network representation of the classical transportation problem is shown in Figure 7-1.

7.1.2 PROPERTIES OF THE SYSTEM

RANK OF THE SYSTEM

LEMMA 7.1 (Rank of the Transportation Problem) *The rank of the system (7.2) is exactly $m + n - 1$. Furthermore, each equation is a linear combination of the other $m + n - 1$ equations so that any one equation may be called redundant and may be discarded if convenient to do so.*

COROLLARY 7.2 (Number of Basic Variables) *There are exactly $m + n - 1$ basic variables x_{ij} .*

▷ **Exercise 7.1** Prove Lemma 7.1 and Corollary 7.2.

COMPACT REPRESENTATION

The special structure of the transportation problem allows us to compactly represent the variables x_{ij} in an $m \times n$ array such that the sum across the rows correspond to the demand constraints and the sums across the columns correspond to the supply constraints. A rectangular array suitable for solving such a transportation problem is shown in Figure 7-2 for a 3×5 case.

In Figure 7-2 the column of cells to the right of the double vertical lines is called the *marginal column* and the row of cells below the double horizontal lines is called the *marginal row*. The rest of the cells is referred to as the *rectangular array*.

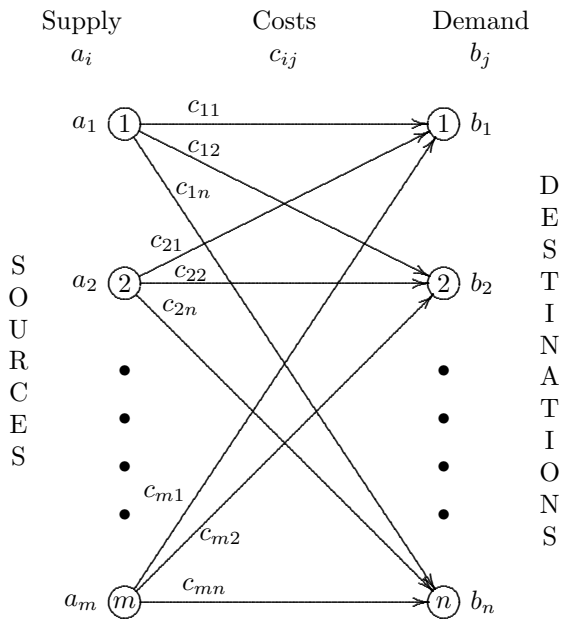


Figure 7-1: Network Representation of the Transportation Problem

x_{11} c_{11}	x_{12} c_{12}	x_{13} c_{13}	x_{14} c_{14}	x_{15} c_{15}	a_1 u_1
x_{21} c_{21}	x_{22} c_{22}	x_{23} c_{23}	x_{24} c_{24}	x_{25} c_{25}	a_2 u_2
x_{31} c_{31}	x_{32} c_{32}	x_{33} c_{33}	x_{34} c_{34}	x_{35} c_{35}	a_3 u_3
b_1 v_1	b_2 v_2	b_3 v_3	b_4 v_4	b_5 v_5	

Figure 7-2: Example of Standard Transportation Array

BASIS TRIANGULARITY

A fundamental property of a transportation (or network-flow) problem is that every basis is triangular.

Definition (Triangular Matrix): We give the following two equivalent definitions of a *triangular* matrix:

1. A square matrix is said to be *triangular* if it satisfies the following properties.
 - (a) The matrix contains at least one row having exactly one nonzero element.
 - (b) If the row with a single nonzero element and its column are deleted, the resulting matrix will once again have this same property.
2. Equivalently, we can define a square matrix to be *triangular* if its rows and columns can be permuted to be either an upper triangular or lower triangular matrix.

Before we prove that every basis is triangular, we establish the following three lemmas.

LEMMA 7.3 (At Least One Basic Entry) *Every row and column has at least one basic variable.*

Proof. By Lemma 7.1 we can drop any one of the equations as redundant without affecting the solution and the rank of the system is $m + n - 1$. It then follows that each of the $m + n - 1$ equalities must have at least one basic variable with a +1 coefficient. ■

LEMMA 7.4 (Exactly One Basic Entry) *There is at least one row or column in the transportation array with exactly one basic entry.*

Proof. Assume, on the contrary, that no row or column has exactly one basic variable. By Lemma 7.3, every row and column has at least one basic entry. Hence, all columns (or rows) under our contrary assumption must have two or more basic entries.

The total number of basic entries in the array is $m + n - 1$ by Lemma 7.1; let

$$k = m + n - 1. \tag{7.4}$$

Then since there are at least two such entries per column, we must have

$$k \geq 2n.$$

Similarly, there are at least two such entries per row, and thus we must have

$$k \geq 2m.$$

Summing these two, we get the number of basic entries k must satisfy

$$k \geq m + n. \quad (7.5)$$

This contradicts (7.4). Therefore the contrary assumption is false, implying that there must be at least one row or column with exactly one basic variable. ■

LEMMA 7.5 (Single Basic Entry Exists after Deletion) *The subsystem obtained by removing any redundant equation from the original system will still contain an equation with exactly one basic variable.*

Proof. Drop some equation as redundant, say the last row equation. Once again make the contrary assumption, that no row or column has exactly one basic variable. Let k' be the total number of basic variables in all but the last row. It is clear that

$$k' \geq 2(m - 1). \quad (7.6)$$

Since there is at least one basic entry in the last row, we have

$$k \geq k' + 1, \quad (7.7)$$

and

$$k \geq 2n. \quad (7.8)$$

Adding the relations we get

$$2k \geq 2m + 2n - 1, \quad (7.9)$$

or

$$k \geq m + n - \frac{1}{2}, \quad (7.10)$$

contradicting the fact that $k = m + n - 1$. This proves the lemma. ■

THEOREM 7.6 (Fundamental Theorem: TP Basis Is Triangular) *Every basis of the transportation problem (7.2) is triangular.*

Proof. Consider a standard transportation array, such as Figure 7-2, with m rows and n columns and with arbitrary marginal constants, a_i and b_j . Consider any particular set of basic variables and substitute the value zero for each of the nonbasic variables. Now, starting with the original array, we set the value of the basic variable in the column or row with a single basic variable (by Lemma 7.4 we know such a row or column exists) equal to its marginal value. We then obtain a *Reduced Array* by deleting the row or column having a single basic entry and reduce the value of the marginal value of its column or row by its value. We next repeat the argument for the reduced array (by Lemma 7.5), thereby establishing the theorem. ■

INTEGER SOLUTION PROPERTY

THEOREM 7.7 (Integrality Property of Basic Variables) *All the basic variables have integer values if the row and column totals a_i and b_j are integers.*

Proof. It is not possible to obtain fractional values when the right-hand sides of the equations have integer values, because the nonzero coefficients of $+1$ imply that all the variables are either set equal to the right-hand side or evaluated by simple subtractions. ■

THEOREM 7.8 (Integral Property of Multipliers) *When the unit costs, c_{ij} , are integers and any one simplex multiplier (u_i or v_j) is given an arbitrary integral value, then all the simplex multipliers will be integers.*

Proof. Since the basis is triangular and of rank $m + n - 1$, so is its transpose. Hence, once we assign an arbitrary integral value to the multiplier of a redundant row, the values of the remaining u_i and v_j satisfying

$$c_{ij} = u_i + v_j \quad \text{for } x_{ij} \text{ basic}$$

can be obtained uniquely in the same manner as the values of the basic variables; i.e., by looking for one equation in one unknown, etc. Since the coefficients in the basis are either unity or zero and one of the multipliers is arbitrarily assigned an integral value, the values of u_i and v_j will be sums and differences of c_{ij} corresponding to basic variables. ■

From the proof it is easy to show that the A matrix is *unimodular* as defined below.

Definition (Unimodular): An $m \times n$ matrix A of rank r is said to be *unimodular* if and only if every element of it is an integer and every square submatrix of size r has a determinant of $+1$, -1 , or 0 .

Definition (Totally Unimodular): An $m \times n$ matrix A is said to be *totally unimodular* if and only if every element of it is an integer and every square submatrix has a determinant of $+1$, -1 , or 0 .

It is obvious that if a matrix is totally unimodular then all the entries are either $+1$, -1 , or 0 .

▷ **Exercise 7.2** Consider the linear program $c^T x$ subject to $Ax = b$, $x \geq 0$. Show that if A is totally unimodular and b is an integer vector then every basic solution is integral.

▷ **Exercise 7.3** For the transportation problem (7.3) or its dual, show that the coefficient matrix is totally unimodular.

- ▷ **Exercise 7.4** Construct a totally unimodular matrix that is not the coefficient matrix of a transportation problem or its dual.
- ▷ **Exercise 7.5** Show that the inverse of a basis for the transportation problem has only -1 , 0 , or $+1$ as its elements.

7.2 FINDING AN INITIAL SOLUTION

We have shown that every basis in the classical transportation problem is triangular. This fact makes it easy to generate a starting basic feasible solution. The simplest way to generate a starting basic feasible solution is by the following triangularity rule (algorithm), also described in *Linear Programming 1*.

Triangularity Rule: Choose arbitrarily any variable x_{pq} as the candidate for the first feasible basic variable. Make x_{pq} as large as possible without violating the row and column totals, i.e., set

$$x_{pq} = \min \{ a_p, b_q \}. \quad (7.11)$$

The next variable to be made basic is determined by this same procedure after *reducing the rectangular array* depending on which of the following three cases arises:

1. If $a_p < b_q$, then all the other variables in the p th row are given the value zero and designated as nonbasic. Next the p th row is deleted, and the value of b_q in column q is reduced to $(b_q - a_p)$.
2. If $a_p > b_q$, then all the other variables in the q th column are given the value zero and designated as nonbasic. Next the q th column is deleted and the value of a_p in row p is reduced to $(a_p - b_q)$.
3. If $a_p = b_q$, then randomly choose either the p th row or the q th column to be deleted, but *not both*. However, if several columns, but only one row, remain in the reduced array, then drop the q th column, and conversely, if several rows and one column remain in the reduced array, drop the p th row. If the p th row is deleted, the value of b_q in column q is reduced to 0. If the q th column is deleted, the value of a_p in row p is reduced to 0.

If after deletion of a row or column there remains only one row or one column, then all remaining cells are basic and are evaluated in turn as equal to the residual amount in the row or column. On the last step exactly one row and one column remain, and both must be dropped after the last variable is evaluated. Thus, this Triangularity Rule will select as many variables for the basic set as there are rows plus columns, less one, i.e., $m + n - 1$.

- ▷ **Exercise 7.6** Show that every reduced array retains the property that the sum of the remaining demand (or marginal row total) is equal to the sum of the remaining supply (or marginal column total). This implies that the last remaining variable can acquire a value consistent with the totals for the single row and column still remaining in a final reduced array.

THEOREM 7.9 (Triangularity Rule Creates a Basic Set) *The set of candidate variables chosen for an initial solution by the Triangularity Rule constitutes a basic set, and conversely every basic set could have been generated by such a rule.*

Proof. The variables picked out by the Triangularity Rule and called “basic” will be true basic variables if we can find values for them when we set the remaining variables equal to zero and arbitrarily choose the values of the right-hand-side equations excluding one that is redundant. The rule actually decides which equation is redundant, namely either row or column associated with the last “basic” variable evaluated. Then arbitrarily choose the values of a_i and b_j except for this redundant row or column in the transportation array. If marginal $b_q - a_p$ was used to evaluate x_{pq} then use the modified $b_q - a_p$ to evaluate x_{pq} even if it is negative, etc. To prove the converse, we note that we have already shown that the basis is triangular and therefore the evaluation process of finding a row or column with a single entry is exactly what the rule would find if we mark in advance the cells that are basic. ■

COROLLARY 7.10 (Totals as Partial Sums) *Every row total of a reduced array is equal to some partial sum of the a_i minus some partial sum of the b_j , whereas every column total of the reduced array is some partial sum of the b_j minus some partial sum of the a_i .*

- ▷ **Exercise 7.7** Use induction to prove Corollary 7.10.

Each basic variable was chosen *arbitrarily* in the original and subsequent reduced arrays. Several authors have suggested Empirical rules that provide a “good” basic solution to start Phase II of the Simplex Method. See *Linear Programming 1* for several such rules, for example: Northwest Corner Rule, The Least Remaining Cost Rule, Vogel’s Approximation Method, and Russel’s Approximation Method.

7.3 FINDING AN IMPROVED BASIC SOLUTION

To distinguish the multipliers corresponding to the rows from those of the columns of the transportation array, let u_i represent the multiplier for the i th-row equation, and let v_j represent the multiplier for the j th-column equation instead of using π_k for all equations k as we did earlier.

In order for a basic column (i, j) to price out to zero, we must have

$$c_{ij} = u_i + v_j \quad \text{for } x_{ij} \text{ basic,} \quad (7.12)$$

because column (i, j) has exactly two nonzero coefficients: $+1$ corresponding to equation i in the demand equations and $+1$ corresponding to equation j in the supply equations; see (7.2).

The reduced costs \bar{c}_{ij} are given by

$$\bar{c}_{ij} = c_{ij} - (u_i + v_j) \quad \text{for } i = 1, \dots, m, \quad j = 1, \dots, n. \quad (7.13)$$

The \bar{c}_{ij} corresponding to the basic variables are all zero by (7.12). The basic feasible solution is optimal if $\bar{c}_{ij} \geq 0$ for all the nonbasic variables, that is

$$c_{ij} \geq u_i + v_j \quad \text{for } i = 1, \dots, m \text{ and } j = 1, \dots, n. \quad (7.14)$$

Thus if for some r and s

$$c_{rs} < u_r + v_s, \quad (7.15)$$

then a new basic feasible solution can be obtained by increasing the value of the nonbasic variable x_{rs} , if possible, and adjusting the values of the basic variable to compensate.

THEOREM 7.11 (Changes in Values of Basic Variables) *In the transportation problem (7.2), if the value of a nonbasic variable x_{rs} is allowed to increase, with the other nonbasic variables remaining at zero, the value of any basic variable x_{pq} will change from x_{pq}^t to*

$$x_{pq}^{t+1} = x_{pq}^t + \delta_{pq} x_{rs}^t, \quad (7.16)$$

where $\delta_{pq} = -1, 0, \text{ or } +1$.

Proof. Given a basis B , for a linear program in standard form, as the nonbasic variable x_s is allowed to increase, the value of the i th basic variable is given by

$$x_{j_i} = \bar{b}_i - \bar{a}_{i_s} x_s$$

where \bar{b} is obtained by solving $B\bar{b} = b$ and $\bar{A}_{\bullet s}$ is obtained by solving $B\bar{A}_{\bullet s} = A_{\bullet s}$.

Note the index pair (r, s) for the incoming basic variable x_{rs} , corresponds to index s for the incoming variable of the general linear program; and the index pair (p, q) for the outgoing variable x_{pq} corresponds to the index j_r for the outgoing basic variable of the general linear program. For the transportation problem, the coefficients of the terms involving x_{rs} are unity in the r th-row equation and in the s th-column equation, and zero elsewhere. Hence, the coefficient of x_{rs} in the canonical form can be obtained by solving for the values imposed on the basic variables when the constants are replaced by $a_r = 1$ and $b_s = 1$, while all other a_i and b_j are zero (this is equivalent to solving $B\bar{A}_{\bullet s} = A_{\bullet s}$ for a linear program in standard form). By Corollary 7.10, the value of a basic variable is the difference (positive or negative) between some partial sum of the a_i (which in this case can only be unity or zero because only $a_r = 1$), and some partial sum of the b_j (also unity or zero because only $b_s = 1$). This difference must clearly be $+1, 0, \text{ or } -1$. This completes the proof. ■

x_{11}	x_{12}	x_{13}	x_{14}	1	
3	5	5	11		u_1
x_{21}	x_{22}	x_{23}	x_{24}	1	
9	7	9	15		u_2
x_{31}	x_{32}	x_{33}	x_{34}	1	
7	7	11	13		u_3
x_{41}	x_{42}	x_{43}	x_{44}	1	
13	13	13	17		u_4
1	1	1	1		
	v_1	v_2	v_3	v_4	

Figure 7-3: Cycling in the Transportation Problem

7.4 DEGENERACY IN THE TRANSPORTATION PROBLEM

Degeneracy may lead to cycling in the transportation problem if the entering variable is always chosen with a negative reduced cost but not necessarily the most negative reduced cost; see Example 7.1. *It is not known if cycling can occur in a transportation problem if the entering variable is always chosen by the usual rule of picking the one which has the most negative reduced cost.* See Section 5.1 for non-transportation examples where cycling occurred using the most negative reduced cost rule.

Example 7.1 (Cycling When Not Using the Most Negative Reduced Cost Rule) Consider the transportation problem defined by the transportation array in Figure 7-3. Recall that the x_{ij} are in the upper-left corner and the specified costs c_{ij} are in the lower-right corner of the cells in the rectangular array; the specified row availabilities a_i are in the upper-left corner and the row multipliers u_i are in the lower-right corner of the marginal column; the specified column demands b_j are in the upper-left corner and the column multipliers v_i are in the lower-right corner of the marginal row.

An initial basic feasible set of variables is $\{x_{11}, x_{22}, x_{33}, x_{44}, x_{12}, x_{23}, x_{34}\}$. All pivot steps in this problem are degenerate pivot steps and the problem cycles after 12 iterations if the entering and leaving variable pairs at each iteration are: $\{x_{13}, x_{23}\}$, $\{x_{42}, x_{12}\}$, $\{x_{32}, x_{34}\}$, $\{x_{41}, x_{42}\}$, $\{x_{43}, x_{13}\}$, $\{x_{21}, x_{41}\}$, $\{x_{31}, x_{32}\}$, $\{x_{24}, x_{21}\}$, $\{x_{23}, x_{43}\}$, $\{x_{14}, x_{24}\}$, $\{x_{34}, x_{31}\}$, $\{x_{12}, x_{14}\}$. Notice that in the last four iterations x_{12} , x_{23} , and x_{34} come back into the basis, thus repeating the initial basis. Note that the first four variables never leave the basic set.

It is interesting from a theoretical perspective to develop a guaranteed anti-cycling scheme for the transportation problem. The earliest such scheme for preventing cycling in the transportation problem is based on a simple perturbation.

If any $a_i = 0$ (or $b_j = 0$) we can drop row i (column j) and the variables in its row (column) from the problem: therefore we can assume that all a_i and b_j are positive. Degeneracy in the transportation problem can be avoided by considering

1		1
ϵ	$1 + \epsilon$	$1 + 2\epsilon$ dropped
$1 + \epsilon$	$1 + \epsilon$	

$1 - \theta$	θ^*	1
$\epsilon + \theta$	$1 + \epsilon - \theta$	$1 + 2\epsilon$ dropped
$1 + \epsilon$	$1 + \epsilon$	

Figure 7-4: Perturbing the Transportation Problem

the class of perturbed problems

$$\begin{aligned}
 &\text{Minimize} && \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} = z \\
 &\text{subject to} && \sum_{j=1}^n x_{ij} = a_i, && i = 1, \dots, m - 1, \\
 &&& \sum_{j=1}^n x_{mj} = a_m + n\epsilon, && \\
 &&& \sum_{i=1}^m x_{ij} = b_j + \epsilon, && j = 1, \dots, n, \\
 &&& x_{ij} \geq 0, && i = 1, \dots, m, j = 1, \dots, n.
 \end{aligned} \tag{7.17}$$

For the discussion that follows, we assume that the last row equation (i.e., the row corresponding to a_m) is dropped as redundant.

Cycling can be prevented as follows. Assume, to initiate the algorithm, that an arbitrary basic solution is chosen by the Triangularity Rule described in Section 7.2, except that no x_{mj} is selected for a basic variable until all other rows $i \neq m$ have been eliminated (see the left part of Figure 7-4). On any subsequent iterative step, if there is a tie as to which basic variable to drop from the basis when $\epsilon = 0$, cycling may occur. Cycling can be prevented by letting $\epsilon \neq 0$ and choosing to drop the basic variable with the smallest coefficient of ϵ among those tied. We will prove that the (i, j) associated with the smallest coefficient is unique. See the right part of Figure 7-4, where the smallest coefficient of ϵ , among those tied, is for x_{11} .

▷ **Exercise 7.8** Work out the details of Example 7.1 to show that the choice of entering variable is *not* based on the most negative reduced cost rule.

▷ **Exercise 7.9** Show that using the most negative reduced cost rule, the algorithm does not cycle on Example 7.1.

- ▷ **Exercise 7.10** Prove that the Triangularity Rule can be applied to generate a basic feasible solution without choosing x_{mj} for basic variables until all other rows $i \neq m$ have been eliminated.

THEOREM 7.12 (Basic Feasible Solution Is Nondegenerate for Perturbed Problem) *Every basic feasible solution for the perturbed problem is nondegenerate for all $0 < \epsilon < \bar{\epsilon}$ for some $\bar{\epsilon} > 0$.*

Proof. Let p be the coefficient of ϵ in any row total (excluding the last row total because the last row has been dropped) of the reduced array and let q be the coefficient of ϵ in any column total of the reduced array.

We first establish that p is either zero or negative and that q is strictly positive in the reduced array. The coefficient of ϵ in any row total of the original or reduced array (excluding row m) is either zero or negative, because it is composed of a nonvacuous partial sum of the a_i , minus a (possibly vacuous) partial sum of the $b_j + \epsilon$. Similarly, the coefficient of ϵ for any column total of the original or reduced array is always positive, because it is composed of a (nonvacuous) partial sum of the $b_j + \epsilon$ minus a (possibly vacuous) partial sum of the a_i (since we are excluding $a_m + n\epsilon$).

We next establish that the *initial* basic solution is nondegenerate for a positive range of ϵ in the neighborhood of $\epsilon = 0$. To do this we will prove that at any stage of evaluating the values of the basic variables in using the Triangularity Rule, the reduced row totals for rows with coefficient of $\epsilon = 0$ are strictly positive and the reduced column totals for columns with coefficient of $\epsilon = 0$ are nonnegative. Initially all a_i and b_j are positive; however for the inductive proof all we need is all a_i positive and all b_j nonnegative. Suppose inductively that for some step of the Triangularity Rule for forming the initial solution it is still true for some reduced array, that $a'_i = \alpha - p\epsilon$ (where α is positive and p is nonnegative), and that $b'_j = \beta + q\epsilon$ (where β is nonnegative and q is positive). If x_{ij} becomes a basic variable, then its value is $\min [(\alpha - p\epsilon), (\beta + q\epsilon)]$. For the case $\alpha \leq \beta$, the row total is satisfied and the new column total becomes $(\beta - \alpha) + (p + q)\epsilon$, where $(\beta - \alpha)$ is nonnegative and $(p + q)$ is positive. On the other hand, if $\beta < \alpha$, then, for ϵ in some range $0 < \epsilon < \tilde{\epsilon}$, the column total is satisfied, and the new row total becomes $(\alpha - \beta) - (p + q)\epsilon$, with $(\alpha - \beta)$ and $(p + q)$ both positive. In either case $x_{ij} > 0$ for $0 < \epsilon < \tilde{\epsilon}$ for some $\tilde{\epsilon} > 0$.

We now show that the basic solution for any subsequent iteration t of the Simplex Method is nondegenerate for some positive range of ϵ . At the start of the iteration we pick the incoming variable and then an outgoing basic variable (we shall see that the latter choice is unique for $0 < \epsilon < \epsilon_t$). We have already seen that the basis (formed by excluding the last row equation) is *triangular*, and, from Theorem 7.9, we can solve for the values by using the Triangularity Rule to obtain a feasible solution. Thus we have that the values of the new basic variables are of the form $\gamma + \nu\epsilon$, where (by a repetition of the same argument) either $\gamma > 0$ and ν arbitrary, or $\gamma = 0$ and $\nu > 0$. Hence, the new basic solution must be nondegenerate for some range $0 < \epsilon < \epsilon_t$.

$+\hat{x}_{11}$	$-x_{12}$	$-x_{13}$	$= \hat{a}_1$
$-x_{21}$	$+\hat{x}_{22}$	$-x_{23}$	$= \hat{a}_2$
$-x_{31}$	$-x_{32}$	$+\hat{x}_{33}$	$= \hat{a}_3$
$\overline{\hat{b}}_1$	$\overline{\hat{b}}_2$	$\overline{\hat{b}}_3$	

Figure 7-5: Example of a Standard Transshipment Array

In general, for any sufficiently small ϵ , there will be a positive (nonzero) decrease in the value of z associated with the basic solution after each iteration. It can be shown that no basic feasible solution can be degenerate if $0 < \epsilon < 1/n$. (See Exercises 7.11 and 7.12.) Thus, no basis can be repeated, and the algorithm will terminate in a finite number of steps. ■

- ▷ **Exercise 7.11 (Orden [1956])** Prove that if a_i, b_j are integers for $i = 1, \dots, m$, $j = 1, \dots, n$ and if b_j are replaced by $b_j + (1/n)$ and a_m by $a_m + 1$, then every basic feasible solution of the new problem is nondegenerate and the corresponding basic solution for the original unperturbed problem is always feasible. How can this be used to guard against the possibility of cycling?
- ▷ **Exercise 7.12 (Orden [1956])** With reference to Exercise 7.11, show that fractions can be avoided in applying the Simplex Algorithm if the original b_j are replaced by $nb_j + 1$ and a_i by na_i except a_m by $na_m + n$.

7.5 TRANSSHIPMENT PROBLEM

7.5.1 Formulation

In the classical Hitchcock transportation problem, shipments are made only from cities where goods are produced (origins) to cities where goods are consumed (destinations); shipments do not take place between origins or between destinations, nor from destinations to origins. In practice, however, the best method of distribution may be through intermediate points (each of which may also serve as a source or as a destination). Shipments through intermediate points are called *transshipments*.

Here we shall consider a generalized transportation model in which transshipment through intermediate cities and local production and local consumption are permitted. See Figure 7-5 for an example of a 3×3 transshipment array. It states that for each city (j, j) there is a material-balance equation stating that the amount

shipped out minus that shipped in is equal to the net amount produced there (if positive), or net amount consumed there (if negative). For every city j this implies for $j = 1, \dots, n$:

$$\begin{aligned} \text{Gross Supply } (\hat{x}_{jj}) &= \text{Total Amount Shipped Out } (x_{jk}) + \text{Consumed } (\hat{b}_j) \\ &= \text{Total Amount Shipped In } (x_{ij}) + \text{Produced } (\hat{a}_j) \end{aligned}$$

or, in equation form, for $j = 1, \dots, n$,

$$\begin{aligned} \hat{x}_{jj} &= \sum_{k \neq j} x_{jk} + \hat{b}_j \\ &= \sum_{i \neq j} x_{ij} + \hat{a}_j, \end{aligned} \tag{7.18}$$

where

$$\begin{aligned} \hat{x}_{jj} &= \text{gross supply at } j, \hat{x}_{jj} \geq 0, c_{jj} = 0, \\ x_{ij} &= \text{total quantity shipped from } i \text{ to } j \text{ for } i \neq j, x_{ij} \geq 0. \\ \hat{b}_j &= \text{the consumption at city } j, \hat{b}_j \geq 0. \\ \hat{a}_j &= \text{the production at city } j, \hat{a}_j \geq 0. \end{aligned}$$

The transshipment problem consists in finding $x_{ij} \geq 0$ and $\min z$ satisfying (7.18) and the objective equation

$$\sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n c_{ij} x_{ij} = z. \tag{7.19}$$

Excluding the cost factor, each column in standard LP format contains only two nonzero coefficients, either both $+1$ or both -1 (or one nonzero coefficient $+1$). The standard transportation model is clearly a special case of this formulation. However, in Section 7.5.3 we shall show that the general transshipment problem is equivalent to the classical transportation problem. More generally, if we allow surplus or shortage, then the rows include slack variables whose corresponding columns contain only one nonzero coefficient $+1$ or -1 .

- ▷ **Exercise 7.13** Show that no feasible solutions exist for the transshipment model shown unless the total production equals the total consumption. Show, however, if we allow surplus or shortage this is no longer true.
- ▷ **Exercise 7.14** Write down the detached coefficient form for the transshipment problem given by equations (7.18) and (7.19).

Let a_j be the net production (referred to as the net amount available) in city j and let b_j be the net consumption (referred to as the net amount required) in

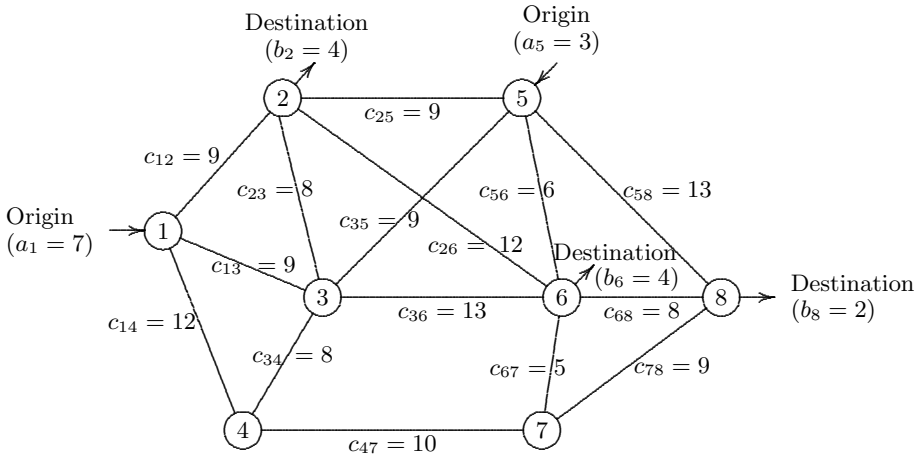


Figure 7-6: The Transshipment Problem

the city. Then the following relationships hold between net production a_j and net consumption b_j and gross production \hat{a}_j and gross consumption \hat{b}_j at city (node) j :

$$a_j = \hat{a}_j - \min(\hat{a}_j, \hat{b}_j), \quad (7.20)$$

$$b_j = \hat{b}_j - \min(\hat{a}_j, \hat{b}_j). \quad (7.21)$$

The transshipment problem defined by equations (7.18) and (7.19), in standard LP format, contains $n(n-1)$ columns corresponding to the number of ways to ship from each city to any other city. If, however, all shipments are routed from one city to another by means of a *chain* of links between neighboring cities, then we need consider only the network composed of such local links and need not need to find minimum path costs from every node i to every node j . All the variables dealing with shipments to non-neighboring cities can be ignored; that is, all variables x_{ij} not corresponding to a local link are inadmissible.

Example 7.2 (Shipping Costs) In the network shown in Figure 7-6, the cost c_{ij} , of shipping a ton of goods from i to a neighboring point j , is shown on the relevant links: thus $c_{36} = 13$ is the cost from 3 to 6. We have not shown c_{63} , the cost from 6 to 3, because in this example we assume each c_{ij} equals c_{ji} . The theory we will develop, however, is valid even when $c_{ij} \neq c_{ji}$.

Although freight rates between two cities are often the same regardless of the direction of shipment, there may be a good economic reason why they might be different. A situation in which c_{ij} is not equal to c_{ji} might actually arise in a pipeline system if i is at the top of a mountain and j is in a valley because it costs less to pump downhill than up. Also, as a stabilizing influence in certain economic

applications, it would be in the public interest to have differing rates to encourage use of a highway at times of least use.

7.5.2 REDUCTION TO THE CLASSICAL CASE BY COMPUTING MINIMUM COST ROUTES

In formulating the transshipment model, we assumed no knowledge of costs except between a subset of pairs of cities (defined to be *neighboring cities*; and the arcs are defined to be *local links*), but we assume that the shipping costs between any pair of non-neighboring cities can be obtained by finding the minimum sum of costs along chains of local links which connect the two cities through all possible *intermediate* points. For small problems, it may not be too difficult to determine all the minimum costs merely by inspecting the network. The actual freight rate between a pair of cities is defined to be the least-cost obtained by this additivity process. If this is less than the local link, then another arc is added to this transshipment network with this lower cost.

Example 7.3 (Cheapest Way to Ship) For instance, the cheapest way to ship from 1 to 6 in the network example of Figure 7-6 is along the link from 1 to 2 and then to 6. Hence, we can set $c_{16} = c_{12} + c_{26} = 9 + 12 = 21$. In this way the transshipment problem can be reduced to the following classical transportation problem:

x_{12}	x_{16}	x_{18}	$a_1 = 7$
9	21	29	
x_{52}	x_{56}	x_{58}	$a_2 = 3$
9	6	13	
$b_1 =$	$b_2 =$	$b_3 =$	
4	4	2	

▷ **Exercise 7.15** Verify that the cheapest cost route in Figure 7-6 from sources 1 and 5 to destinations 2, 6, and 8 is as shown in Example 7.3. Solve the transportation problem and interpret the results to determine the actual shipping routes. Put these amounts back in Figure 7-6 and verify that the conservation-of-flow conditions are satisfied.

7.5.3 REDUCTION TO THE CLASSICAL CASE BY THE TRANSSHIPMENT PROCEDURE

In this section our purpose is to show an alternative approach to the minimum-cost approach of converting the transshipment problem to the classical transportation problem, which has certain advantages:

1. It avoids the necessity of determining a least-cost route for every origin-destination pair.
2. It permits treatment of problems in which certain arcs of the network have fixed capacities bounding the flows over these arcs.

3. It may involve fewer variables, because the number of arcs of a network is often considerably less than the number of origin-destination pairs.

We shall modify the problem to include the amount transshipped x_{jj} through point j instead of the gross supply \hat{x}_{jj} at point j . In order to define x_{jj} , there are two cases to consider: the production at j is greater than the consumption there, i.e., $\hat{a}_j \geq \hat{b}_j$, or vice versa. Note that

$$\begin{aligned} a_j &= \hat{a}_j - \hat{b}_j & \text{if } \hat{a}_j \geq \hat{b}_j, \\ b_j &= \hat{b}_j - \hat{a}_j & \text{if } \hat{a}_j < \hat{b}_j. \end{aligned}$$

It is straightforward to see that the following relations hold:

$$x_{jj} = \hat{x}_{jj} - \hat{a}_j \quad \text{if } \hat{a}_j \geq \hat{b}_j, \quad (7.22)$$

$$x_{jj} = \hat{x}_{jj} - \hat{b}_j \quad \text{if } \hat{a}_j < \hat{b}_j. \quad (7.23)$$

These equations imply that

$$x_{jj} = \hat{x}_{jj} - \hat{a}_j - \hat{b}_j + \min(\hat{a}_j, \hat{b}_j). \quad (7.24)$$

Substituting (7.24) in (7.18), the transshipment problem can be stated as:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} = z \\ \text{subject to} \quad & \sum_{\substack{j=1 \\ j \neq i}}^n x_{ij} - x_{ii} = a_i, \quad i = 1, \dots, n, \\ & \sum_{\substack{i=1 \\ i \neq j}}^n x_{ij} - x_{jj} = b_j, \quad j = 1, \dots, n, \\ & x_{ij} \geq 0, \quad i = 1, \dots, n, \quad j = 1, \dots, n, \end{aligned} \quad (7.25)$$

with $c_{jj} = 0$ for all j . At this point it is still not in the form of a classical transportation problem.

- ▷ **Exercise 7.16** Write down the standard array (see Figure 7-5) for the transshipment problem displayed in (7.25).

THEOREM 7.13 (Triangularity of Basis) *Every basis for the transshipment problem is triangular.*

- ▷ **Exercise 7.17** Prove Theorem 7.13.

THEOREM 7.14 (Diagonal Entries Are Part of the Basic Set) *For the transshipment problem (7.18), the diagonal variables, x_{jj} or \hat{x}_{jj} , are a part of every basic feasible set.*

Proof. Consider a new transshipment problem for which \hat{a}_j and \hat{b}_j in (7.18) is replaced by $\hat{a}_j + \epsilon$ and $\hat{b}_j + \epsilon$ respectively, where ϵ is an arbitrary positive number, and $x'_{ij} = x_{ij}$ are the new values of the variables for $i \neq j$ and $x'_{jj} = \hat{x}_{jj} + \epsilon \geq \epsilon$ for $i = j$. Hence the diagonal variables, \hat{x}_{jj} , are positive and therefore must form part of every basic feasible set. ■

From a procedural point of view, it is not desirable to transform the problem explicitly, since we can accomplish the same end simply by allowing the supply variables \hat{x}_{jj} an *unrestricted range* of values. They will then be retained in the basic set, once they have entered it, even though their values may be zero. The same applies to the transshipment variables x_{jj} since they are in one-to-one correspondence.

THEOREM 7.15 (Relationship For Implicit Prices) *The implicit prices, u_j and v_j , for the transshipment problem can be made to satisfy the relation*

$$-u_j = v_j \quad \text{for } j = 1, \dots, n. \quad (7.26)$$

Proof. The costs $c_{ij} = u_i + v_j$ for all basic variables x_{ij} . Since $c_{jj} = 0$, and according to Theorem 7.14 x_{jj} is basic, it follows that $-u_j = v_j$. ■

Note: It is common practice to use π_j to denote the common values $-u_j$ and v_j .

The transshipment problem (7.25) differs from the classical transportation problem in that feasible solutions exist in which $x_{ij} \rightarrow \infty$. To see this consider the values in a 2×2 diagonal submatrix formed by the intersection of rows i and j with columns i and j . These values can be increased by an arbitrary constant γ , since the row and column sums of the resulting subarray remain unchanged, as in

$$\begin{pmatrix} -x_{ii} & x_{ij} \\ x_{ji} & -x_{jj} \end{pmatrix} \text{ is equivalent to } \begin{pmatrix} -(x_{ii} + \gamma) & (x_{ij} + \gamma) \\ (x_{ji} + \gamma) & -(x_{jj} + \gamma) \end{pmatrix}. \quad (7.27)$$

If all costs $c_{ij} > 0$ for $i \neq j$ (recall that $c_{jj} = 0$), it clearly never pays to transship an amount greater than the total available from all sources. However, if $c_{ij} < 0$ for some $i \neq j$, there may be no lower bound for the objective function z . For example, if $c_{ij} + c_{ji} < 0$, then $z \rightarrow -\infty$ for the class of solutions generated by $\gamma \rightarrow +\infty$ in (7.27). More generally, it would pay to have such a *circulation* in the flow of the network whenever the sum of the c_{ij} around some loop is negative.

▷ **Exercise 7.18** In the transshipment problem (7.25) show that if $\sum_{i=1}^n a_i \neq \sum_{j=1}^n b_j$ then no feasible solutions exist, and that if $\sum_{i=1}^n a_i = \sum_{j=1}^n b_j$, then feasible solutions exist.

THEOREM 7.16 (Optimal Amount Transshipped Is Bounded Case) *If the sum of c_{ij} around every loop in the network is positive, then in any optimal solution, if one exists, the amount transshipped, x_{jj} , is bounded, and*

$$x_{jj} \leq \sum_{i=1}^n a_i = \sum_{j=1}^n b_j = \alpha. \quad (7.28)$$

▷ **Exercise 7.19** Prove Theorem 7.16.

▷ **Exercise 7.20** Defining transshipment slack by

$$\bar{x}_{jj} = \alpha - x_{jj}, \quad (7.29)$$

where $\alpha = \sum_i a_i = \sum_j b_j$, reduce the transshipment problem (7.25) to a classical transportation problem.

▷ **Exercise 7.21** Consider the classical transportation problem with three sources and four destinations

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^3 \sum_{j=4}^7 c_{ij} x_{ij} = z \\ \text{subject to} \quad & \sum_{j=4}^7 x_{ij} = a_i \text{ for } i = 1, 2, 3, \\ & \sum_{i=1}^3 x_{ij} = b_j \text{ for } j = 4, 5, 6, 7. \end{aligned} \quad (7.30)$$

Show how to convert it to a transshipment problem of the form (7.25) with seven transshipment nodes x_{jj} . Show that $x_{jj} = 0$ for all j .

7.6 TRANSPORTATION PROBLEMS WITH BOUNDED PARTIAL SUMS

A transportation problem with upper bound on the variables is called a *capacitated transportation problem*:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} = z \\ \text{subject to} \quad & \sum_{j=1}^n x_{ij} = a_i, \quad i = 1, \dots, m, \\ & \sum_{i=1}^m x_{ij} = b_j, \quad j = 1, \dots, n, \\ & 0 \leq x_{ij} \leq h_{ij}, \quad i = 1, \dots, m, \quad j = 1, \dots, n. \end{aligned} \quad (7.31)$$

A. S. Manne formalized a way to bound partial sums of variables of which simple bounds on variables is a special case. For simplicity, we consider a case with only one such partial sum; for example, in the scheduling of jobs, a condition such as

$x_{11} \leq 40$ might be interpreted to mean that at most one man can be assigned to job 1 in week 1. In some problems a more involved condition might be desired, such as $x_{11} + x_{31} + x_{61} \leq 40$, expressing the circumstance that jobs 1, 3, and 6 can be assigned only to individual 1. Similarly, a condition such as $x_{11} + x_{12} + x_{13} \geq k$ might mean that at least k hours must be worked on job 1 during the first three weeks. Similar to a capacitated transportation problem, a transportation problem subject to a bounded partial sum of variables in a row, or in a column of a transportation array (See Figure 7-2), can be reduced to a standard transportation problem. To see this, consider system (7.31) with the added condition

$$x_{11} + x_{12} + \dots + x_{1k} \leq \alpha. \tag{7.32}$$

If we introduce a slack variable x_{10} and a variable $y_{10} = x_{11} + x_{12} + \dots + x_{1k}$ the problem can be written in the standard form by splitting the first row and adding a new column as follows.

x_{10}	x_{11}	\cdots	x_{1k}		α		
y_{10}			$x_{1,k+1}$	\cdots	x_{1n}	a_1	
	x_{21}	\cdots	x_{2k}	$x_{2,k+1}$	\cdots	x_{2n}	a_2
	\vdots	\cdots	\vdots	\vdots	\cdots	\vdots	\vdots
	x_{m1}	\cdots	x_{mk}	$x_{m,k+1}$	\cdots	x_{mn}	a_m
α	b_1	\cdots	b_k	b_{k+1}	\cdots	b_n	

The squares where there are no variables displayed will be inadmissible squares in the transportation array. In equation form, the problem can be stated as:

$$\begin{aligned}
 \text{Minimize} \quad & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} = z \\
 \text{subject to} \quad & x_{10} + \sum_{j=1}^k x_{1j} = \alpha, \\
 & y_{10} + \sum_{j=k+1}^n x_{1j} = a_1, \\
 & \sum_{j=1}^n x_{ij} = a_i, \quad i = 2, \dots, m, \\
 & x_{10} + y_{10} = \alpha, \\
 & \sum_{i=1}^m x_{ij} = b_j, \quad j = 1, \dots, n, \\
 & 0 \leq x_{ij} \leq h_{ij}, \quad i = 1, \dots, m, \quad j = 1, \dots, n, \\
 & x_{10} \geq 0, \quad y_{10} \geq 0.
 \end{aligned} \tag{7.33}$$

Note that each x_{ij} appears in at most one row equation and in one column equation.

Clearly any number of conditions of the form (7.32) can be added to the system by using the ideas just discussed. For example, an added condition on column 2,

$$x_{12} + x_{42} + x_{72} \leq \beta, \quad (7.34)$$

can be handled by splitting column 2 and using a second slack variable x_{20} and the variable y_{20} . There could also be other conditions on column 2, such as

$$x_{22} + x_{32} + x_{52} \leq \gamma, \quad (7.35)$$

that do not involve the same variables; this could be handled similarly. There could also be more than one condition on the same variables in the same column, for example, condition (7.34) and

$$x_{12} + x_{72} \leq \delta \quad (7.36)$$

can be taken care of by further splitting the column associated with the variables x_{12} , x_{42} , and x_{72} .

THEOREM 7.17 (Reduction to a Transportation Problem) *A transportation problem with added partial sum conditions in rows and columns can be reduced to a standard transportation problem, if any two conditions in a column (or row) either have no variables in common or the variables of one of the conditions are a subset of the variables of the other condition.*

THEOREM 7.18 (Basis Need Not Be Triangular) *Consider a transportation problem with added partial sums of variables. If a bounded partial sum of variables contains two variables x_{ij} and x_{kl} such that $i \neq k$ and $j \neq l$, the basis need not be triangular, so that nonintegral basic solutions can be obtained.*

▷ **Exercise 7.22** Prove Theorems 7.17 and 7.18.

7.7 NOTES & SELECTED BIBLIOGRAPHY

As noted in Section 7.4, it is not known if cycling can occur in transportation problems if the entering variable is chosen based on the usual rule of picking the one that has the most negative reduced cost. The very contrived example of cycling in Section 7.4, due to L. Johnson, can be found in Gassner [1964] and Murty [1983]. Other examples along the same lines can be found in Cunningham [1979] and Cunningham & Klinecicz [1983]; see also Chvátal [1983] for a cycling example due to Cunningham. A. Orden [1956] proposed a first order perturbation scheme to avoid cycling in transportation problems; see Exercises 7.11 and 7.12. Cunningham [1979] also proposed an elegant and simple way to avoid cycling in networks.

A. Orden [1956] first proposed a generalized transportation model in which *transshipment* through intermediate cities is permitted. As a stabilizing influence in certain economic applications, Koopmans [1947] and Koopmans & Reiter [1951], have suggested that it would be in the public interest to have differing rates to encourage demands in the

direction of least use between two cities. For the transshipment problem (7.25), Koopmans & Reiter [1951] call π_j the “potential” of point i in the network, because it is analogous to the electrostatic potential of an electrical network.

7.8 PROBLEMS

- 7.1 (a) Derive a Dual-Simplex Algorithm for the classical transportation problem.
 (b) Derive a Primal-Dual Algorithm for the classical transportation problem.
- 7.2 (a) Generalize the transshipment model to allow for the storing of excesses at a city when the total of amounts shipped-in plus produced may possibly exceed the total of amounts shipped-out plus consumed.
 (b) Show that, in this generalized model, no feasible solution exists if $\sum_i \hat{a}_i < \sum_j \hat{b}_j$. Interpret the result.
 (c) Why is $x_{ii} \geq 0$ implied by the standard transshipment array?
 (d) In any transshipment problem, prove that if x_{jj} exceeds $\sum_i a_i$, then there is a circularity in the flow pattern, and show that such a solution cannot be optimal if all c_{ij} are positive.
- 7.3 Is the following statement true or false?

In a transportation problem, if the demands and supplies are all even integers, then there is always an optimal solution with even integers.

Justify your answer.

- 7.4 Suppose that for an $n \times n$ assignment problem all the unit right-hand-side components are changed to an integer $\gamma > 1$. Prove that in any basic feasible solution, exactly n basic variables are nonzero.
- 7.5 Let P be a $T \times T$ identity matrix. Partition the T rows arbitrarily into m mutually exclusive adjacent sets R_i and partition the columns into n mutually exclusive adjacent sets C_j . Let x_{ij} be the sum of the 1s that are in the intersection of the row set R_i and the column set C_j . Prove that no more than $m+n-1$ of the x_{ij} are nonzero. Replace P by a permutation matrix and prove the same theorem.
- 7.6 *Ph.D. Comprehensive Exam, Fall 1984, at Stanford.* Given a linear program

$$\begin{aligned} \text{Minimize} \quad & \sum_{j=1}^n c_j x_j = z \\ \text{subject to} \quad & \sum_{j=1}^n P_j x_j = 0, \\ & \text{and } l_j \leq x_j \leq h_j, \end{aligned} \tag{7.37}$$

where c_j , l_j , and h_j are known scalars; P_j are known vectors of dimension m ; and x_j are unknown scalars to be determined.

Let $B = [P_1, P_2, \dots, P_m]$ be nonsingular and let $x^o = (x_1^o, \dots, x_m^o; x_{m+1}^o, \dots, x_n^o)$ satisfy $\sum_{j=1}^n P_j x_j^o = 0$, $l_j < x_j^o < h_j$ for $j = 1, \dots, m$ and $x_j^o =$ either l_j or h_j for $j = m+1, \dots, n$.

-
- (a) Someone asserts x^o is the unique minimizer for z . How would you prove this assertion or demonstrate (on the contrary) that it is false?
- (b) Assume for $j = 1, \dots, n$ that $l_j \neq h_j$ and B as defined above is nonsingular. Consider the class of 2^{n-m} (not necessarily primal feasible) solutions, $\sum_{j=1}^n P_j x_j = 0$ where x_j is set equal to either l_j or h_j for $j = m+1, \dots, n$. State conditions that imply that exactly one of the 2^{n-m} solutions is dual-feasible.
- (c) How would you “reduce” a linear program in standard form to the linear program (7.37)?
- (d) How would you “reduce” a capacitated transportation problem to the linear program (7.37)?

This page intentionally left blank

NETWORK FLOW THEORY

Network theory concerns a class of linear programs having a special network structure. The combinatorial nature of this structure has resulted in the development of very efficient algorithms that combine ideas on data structures from computer science with mathematical concepts from operations research.

Networks arise in a wide variety of situations. For example, the transportation problem discussed in Chapter 7 is a network representing the shipment of goods from sources to destinations. Network problems arise naturally in the distribution of electric power in the design of electric circuits, in communications networks, and in hydro-systems in which water flows from conduits from various points to others. Typically, the analysis of a network requires finding a maximal-flow solution when there are capacity constraints on the arcs, a shortest-path solution when there are lengths assigned to arcs, a minimum spanning-tree solution, a least-cost solution, or determining the optimal sequence of tasks to be performed. The ability to obtain, under certain conditions, integer-valued solutions has made it possible to extend network analysis to many different areas such as facilities location, project planning (PERT, CPM), and resource management.

We shall illustrate some definitions and concepts of directed networks by referring to Figure 8-1, which displays a simple directed network.

In the figure, recall that the circles numbered 1, 2, 3, and 4 are called *nodes*; the lines joining them are called *arcs*; and the arrowheads on the arcs show the direction of flow. In all, there are four nodes and six directed arcs.

We shall use the following notation:

$$Af(k) = \{j \in \mathcal{N}d \mid (k, j) \in \mathcal{A}c\}, \quad (8.1)$$

$$Bf(k) = \{i \in \mathcal{N}d \mid (i, k) \in \mathcal{A}c\}, \quad (8.2)$$

where $Af(k)$ stands for “after” (or “out of”) node k , $Bf(k)$ stands for “before” (“into”) node k , $\mathcal{N}d$ is the set of nodes in the network, and $\mathcal{A}c$ is the set of arcs in the network.

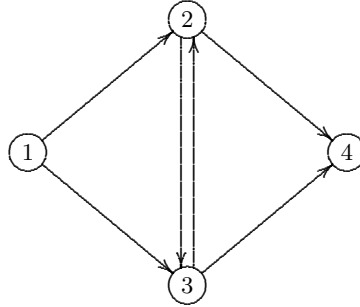


Figure 8-1: A Simple Directed Network

8.1 THE MAXIMAL FLOW PROBLEM

Consider a network with a single source node, $s = 1$, and a single destination node, $t = m$, connected by several intermediate nodes. Except for the nodes 1 and m (the source and destination nodes), the flows into and out of each node k must balance; such relations are called *conservation of flows* (in physics, the condition that the quantity of electrons flowing into a point of an electrical network must equal the amount flowing out is referred to as *Kirchoff's Law*). That is, for an intermediate node k :

$$\sum_{i \in Bf(k)} x_{ik} - \sum_{j \in Af(k)} x_{kj} = 0, \quad \text{for } k = 2, \dots, m-1, \quad (8.3)$$

where the first summation is over all directed arcs that have node k as a head node, and the second summation is over all directed arcs that have node k as a tail node. If we denote F as the *exogenous flow* into the source $s = 1$ from *outside* the network, then

$$F - \sum_{j \in Af(1)} x_{1j} = 0 \quad (8.4)$$

because there are no other flows (or flows on arcs) incoming into the source node. If we denote H as the exogenous flow from its destination node t to outside the network, then

$$\sum_{i \in Bf(m)} x_{im} - H = 0. \quad (8.5)$$

If we sum the $m-2$ relations in (8.3) and (8.4) then each variable x_{ij} appears in exactly two equations with opposite signs (recall the node-arc incidence matrix) and hence cancels, resulting in $F = H$. Therefore:

$$\sum_{i \in Bf(m)} x_{im} - F = 0. \quad (8.6)$$

8.1.1 DECOMPOSITION OF FLOWS

The next theorem shows that a flow can be decomposed into simpler components that involve simple paths and circuits. It is useful because it shows that a solution to a flow problem or a transshipment problem corresponds to our intuitive notion that items start from nodes of surplus and move from one node to the next without losing their identity until arriving finally at some node of deficit.

THEOREM 8.1 (Decomposition of Flow) *Consider a network (N, d, \mathcal{A}) where the capacity constraints are $0 \leq x_{ij} \leq h_{ij}$ for $(i, j) \in \mathcal{A}$. An incoming exogenous flow of $F > 0$ and a set of flows $x_{ij} = x_{ij}^o$ that satisfy the capacity constraints and conservation equations (8.3)–(8.6) can be decomposed into a sum of path flows from source to destination and circuit flows such that the direction of these flows in any common arc is the same as that of the directed arc in \mathcal{A} .*

Proof. By hypothesis the incoming flow is $F > 0$. Begin to generate a path starting at the source node $s = 1$ with an initial arc $(1, i_1)$, satisfying

$$i_1 = \operatorname{argmax}_{i \in Af(1)} x_{1i}^o. \quad (8.7)$$

Note that $x_{1i_1}^o > 0$ follows from $F > 0$ and the conservation relation

$$F = \sum_{i \in Af(1)} x_{1i}^o.$$

Next repeat the procedure starting at node i_1 instead of $s = 1$, and generate i_2 , the second node along the path by

$$i_2 = \operatorname{argmax}_{i \in Af(i_1)} x_{i_1 i}^o. \quad (8.8)$$

Again by conservation of flows at i_1 and the previous result of $x_{1i_1}^o > 0$, it follows that $x_{i_1 i_2}^o > 0$.

If we continue the process of generating nodes along the path, we either

1. generate a path that returns to a node arrived at earlier, thus forming a circuit, or
2. complete a path to the destination.

If a circuit \mathcal{C} is generated we subtract a constant K from each x_{ij}^o for arcs (i, j) that belong to the circuit, where

$$K = \min_{\{(i,j) \in \mathcal{C}\}} x_{ij}^o > 0. \quad (8.9)$$

At the node where the path first formed a circuit there must have been a positive input flow, and therefore starting again at this node, the path-generation procedure can be continued with the adjusted x_{ij}^o values. Only a finite number of circuits can

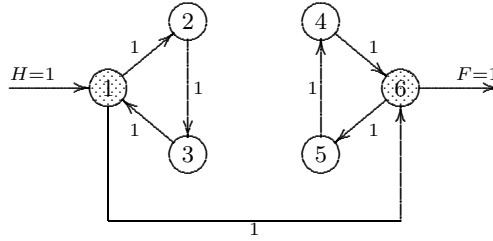


Figure 8-2: Decomposition of Flows

be removed from the solution by this procedure, since each new solution generated by a circuit removal creates at least one more x_{ij}^o whose adjusted value is zero. Hence, after a finite number of circuit removals, a path from origin to destination can finally be constructed with positive flow along it. A value θ then can be assigned to the path \mathcal{P} by setting

$$\theta = \min_{\{(i,j) \in \mathcal{P}\}} x_{ij}^o > 0. \tag{8.10}$$

A new feasible solution is now constructed by setting $x_{ij}^1 = x_{ij}^o - \theta$ for $(i, j) \in \mathcal{P}$ and $x_{ij}^1 = x_{ij}^o$ otherwise; its flow value is $F_1 = F - \theta$.

The entire path augmentation procedure can now be repeated with the new problem if $F_1 > 0$. Again we note that there can only be a finite number of path removals because each new solution has at least one more adjusted x_{ij}^1 that is zero.

Finally, if $F_k = 0$, for some k and some adjusted $x_{ij}^k > 0$, starting with node i and arc (i, j) , the procedure can be followed to construct a circuit that can be removed. In a finite number of steps, all residual circuits can be removed. ■

▷ **Exercise 8.1** Apply the algorithm suggested by the proof of Theorem 8.1 to decompose the flows in Figure 8-2.

▷ **Exercise 8.2** Show that the knowledge of the capacities on the arcs is never used other than to verify that the initial flows do not violate the capacities.

8.1.2 THE AUGMENTING-PATH ALGORITHM FOR MAXIMAL FLOW

The maximal-flow problem for a network is to find the maximum amount that can be transferred from the source to the destination given arc-capacity constraints $0 \leq x_{ij} \leq h_{ij}$ and the existence of a feasible flow $x = x^o$. It is clear that solving the

maximal flow problem is the same as solving the linear program

$$\begin{aligned}
 & \text{Maximize} && F \\
 & \text{subject to} && \sum_{j \in Af(1)} x_{1j} = -F, \\
 & && \sum_{i \in Bf(k)} x_{ik} - \sum_{j \in Af(k)} x_{kj} = 0, \text{ for } k = 2, \dots, m-1, \\
 & && \sum_{i \in Bf(m)} x_{im} = F, \\
 & && 0 \leq x_{ij} \leq h_{ij}, \text{ for all } (i, j) \in \mathcal{A}.
 \end{aligned} \tag{8.11}$$

Definition (Flow Value): The variable F , the exogenous flow into the system, is called the *flow value*.

Before we describe an algorithm for finding the maximal flow, we state and prove two theorems.

THEOREM 8.2 (Existence of a Positive Maximal Flow) *In a network with bounds $0 \leq x_{ij} \leq h_{ij}$ for all $(i, j) \in \mathcal{A}$, the maximal flow is positive if and only if there exists a chain of arcs joining the source to the destination such that a positive θ -flow along the associated flow path is possible.*

Proof. The *if* part of the theorem is obvious. To prove the *only if* part, assume *on the contrary* that the maximal flow $F > 0$ even though there exists no path with a positive flow from the source to the destination. Let $x_{ij} = x_{ij}^o$ with $0 \leq x_{ij}^o \leq h_{ij}$ be the arc flows corresponding to this maximal flow $F > 0$. By Theorem 8.1 it is possible to decompose this positive maximum flow into a sum of path flows and circuit flows, with at least one positive path flow. Along such a positive flow path with positive flow $\theta > 0$ we must have

$$0 < \theta \leq x_{ij}^o \leq h_{ij},$$

because the method of decomposition is such that each $x_{ij}^o > 0$ is represented as a sum of nonnegative path flows along the directed arc joining i to j , contrary to our assumption that no such path exists. ■

THEOREM 8.3 (Existence of an Improving Flow) *Consider a network (Nd, \mathcal{A}) with arc-capacities $0 \leq x_{ij} \leq h_{ij}$ for all $(i, j) \in \mathcal{A}$. Given a feasible flow $x = x_{ij}^o$ with $F = F_o$, a flow value $F > F_o$ can be found if and only if there exists a chain of arcs joining the source to the destination such that a positive θ -flow augmentation along the associated flow path is possible.*

Proof. The *if* part of the theorem is obvious. To prove the *only if* part construct an *associated network* as follows. Subtract x_{ij}^o from the upper bound h_{ij} on arc capacity to obtain a new upper bound $h_{ij} - x_{ij}^o$ on the arc capacity. Add a reverse

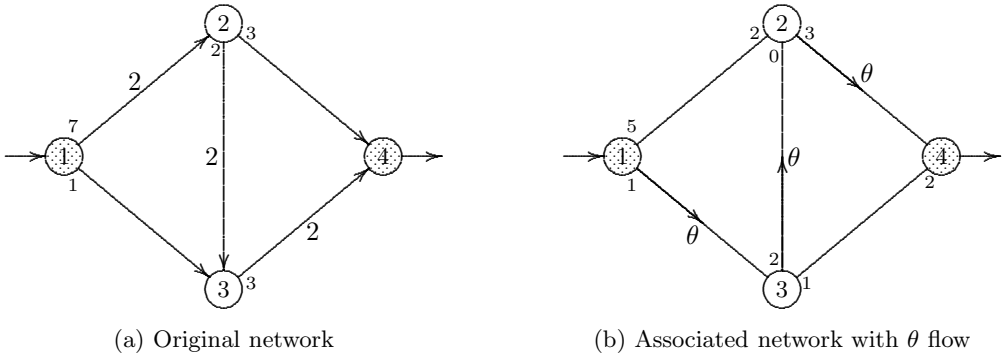


Figure 8-3: Original Network and Associated Network with θ Flow

arc (j, i) with an upper bound x_{ij}^o on arc capacity. (For example, see Figure 8-3 with capacities h_{ij} shown near the nodes, and where the inclusion of a reverse arc is shown with a positive capacity in the reverse direction.) If $x_{ij}^o = 0$, the reverse arc (j, i) may be omitted.

We will now show that the two networks are equivalent. For the associated network, let u_{ij} represent the flow on the arc (i, j) corresponding to the arc (i, j) in the original network; and let v_{ji} represent the flow on arc (j, i) if a reverse arc (j, i) was added to the associated network. Note that corresponding to the flow x_{ij}^o in the original network is the flow $u_{ij} = 0, v_{ji} = 0$ in the associated network. Then if ξ_{ij} is any feasible flow on the original network we can construct a corresponding feasible flow on the associated network as follows:

$$\begin{aligned} u_{ij} &= \xi_{ij} - x_{ij}^o, & v_{ji} &= 0, & \text{if } x_{ij}^o \leq \xi_{ij} \leq h_{ij}, \\ u_{ij} &= 0, & v_{ji} &= x_{ij}^o - \xi_{ij}, & \text{if } \xi_{ij} < x_{ij}^o. \end{aligned} \tag{8.12}$$

Conservation of flow clearly holds in the associated network (see Exercise 8.3). On the other hand, if we are given a feasible flow, u_{ij}, v_{ij} , on the associated network, then we can construct a corresponding feasible flow ξ_{ij} on the original network as follows:

$$\xi_{ij} = u_{ij} - v_{ji} + x_{ij}^o. \tag{8.13}$$

Conservation of flow clearly holds in the original network (see Exercise 8.3).

Now we are ready to prove the *only if* part of the theorem. By equivalence of the two networks, if a feasible flow with value $F' + F_0 > F_0$ exists for the original network, then a feasible flow with value $F' > 0$ exists for the associated network. By the Decomposition of Flow Theorem 8.1 we know that this feasible flow with value F' can be decomposed into sum of positive θ path flows from source to destination and circuit flows such that the direction of these flows in any common arc is the same as that of the directed arc. By equivalence of the two networks, it is easy to see that any positive θ path flow in the associated network corresponds to a positive

θ augmentation flow path along a chain of arcs joining the source to the destination in the original network; where $u_{ij} = \theta$ corresponds to an increase in the flow along arc (i, j) of the original network and $v_{ji} = \theta$ corresponds to a decrease in the flow along arc (i, j) in the original network. ■

▷ **Exercise 8.3** In the proof of Theorem 8.3, show that conservation of flow holds when converting the flows in the original network to flows in the associated network.

An algorithm used to find the maximal flow is the *Augmenting Path Algorithm*. This algorithm is described in *Linear Programming 1*. Instead of adjusting the flows in the network until an optimal flow is obtained, it adjusts the arc capacities by the flow on each augmenting path. Upon termination, the optimal flows can be obtained as the difference between the original and final arc capacities.

THEOREM 8.4 (Finite Termination with Integer Capacities) *If the arc capacities are all integers and a maximal flow exists, the Augmenting Path Algorithm will generate only a finite number of path augmenting flows whose algebraic sum is the maximal flow.*

Proof. Since the arc capacities h_{ij} are assumed to be integers, the path flow $\theta > 0$ at each iteration must also be an integer $\theta \geq 1$. Therefore F must be increased by at least 1 in each iteration. Furthermore, the adjusted arc capacities for each successive associated network must also be integral. We are given that the maximal flow is finite, therefore the algorithm must terminate in a finite number of iterations with the maximal flow. ■

COROLLARY 8.5 (Finite Termination with Rational Capacities) *If the arc capacities are all rational numbers and a maximal flow exists, the Augmenting Path Algorithm will construct only a finite-number of path flows whose algebraic sum is the maximal flow.*

▷ **Exercise 8.4** Prove Corollary 8.5.

A systematic procedure (which is a variation of the shortest-path algorithm discussed in Section 8.2) for finding augmenting paths is the *fanning out* (or *breadth-first unblocked search*) procedure (see *Linear Programming 1*). Note that an arc is said to be *blocked* if the flow on the arc cannot be increased because the arc flow is equal to the arc capacity. This requires forming a tree of all the nodes j that can be reached from the source s by a flow-augmenting path. At the end we obtain an augmenting path with the smallest number of arcs.

THEOREM 8.6 (Edmonds-Karp Max-Flow Theorem) *If a maximal flow exists, the Augmenting Path Algorithm, when used with the Breadth-First Unblocked Search Algorithm to find the augmenting paths, will construct at most $mn/2$ path flows whose algebraic sum is the maximal flow, where n is the number of arcs and m is the number of nodes.*

Proof. Let $l_{ij} = 0$ and h_{ij} be the *original* lower and upper bounds on arc flow x_{ij} . (Note that we assume for convenience a lower bound of $l_{ij} = 0$ on all arc flow; the proof does not rely on this, the lower bound could be any $l_{ij} \leq h_{ij}$.) Let x^o be the feasible initial flow vector and let x^τ , $\tau = 1, 2, \dots$, be the flow vector at the end of iteration τ ; that is, x^τ is the sum of x^o and the τ maximum augmenting path flows. Let AP_τ be the τ th augmenting path.

A breadth-first search implies that for all τ , given $x^{\tau-1}$, the path AP_τ chosen is any augmenting path with the minimum number of arcs. Finding such an augmenting path by the breadth-first unblocked search algorithm is similar to finding the shortest path (see Section 8.2) and is done by assigning arc lengths of 1 to all unblocked arcs and ignoring all blocked arcs.

Define for any iteration τ , given $x^{\tau-1}$,

$$\eta_\tau(k, l) = \begin{array}{l} \text{The smallest number of arcs from node } k \text{ to node } l \text{ in} \\ \text{paths } P \text{ passing through nodes } k \text{ and } l \text{ along which aug-} \\ \text{menting flow is possible from } k \text{ to } l. \text{ If no such } P \text{ exists,} \\ \text{then } \eta_\tau(k, l) = \infty. \end{array}$$

Then for any i and any iterations $(\tau, \tau + 1)$,

$$\begin{aligned} \eta_\tau(s, i) &\leq \eta_{\tau+1}(s, i) \\ \eta_\tau(i, t) &\leq \eta_{\tau+1}(i, t). \end{aligned}$$

The result is clearly true if in iteration $\tau + 1$ the arcs of the augmenting path are either totally different or if these arcs are in the same direction as those in iteration τ . Suppose that in iteration $\tau + 1$ arc (α, β) is traversed in the reverse direction from that in iteration τ and it is the first arc for which this happens on the path. The distance to β is clearly greater for iteration $\tau + 1$. The distance from α to i is also clearly greater for iteration $\tau + 1$ because the argument can be repeated for other arcs traversed in the reverse direction.

We say that (i, j) has become *upper-blocked* on iteration τ if $(i, j) \in AP_\tau$ and $x_{ij}^\tau = h_{ij}$. Suppose $(i, j) \in AP_\tau$ were upper-blocked on iteration τ and $(i, j) \in AP_{\bar{\tau}}$, $x_{ij}^{\bar{\tau}} \leq h_{ij}$ for $\bar{\tau} > \tau$, then there exists an l , where $\tau < l < \bar{\tau}$, such that $(i, j) \in AP_l$ and $x_{ij}^l < h_{ij}$. This is clearly true because if an arc were upper-blocked then it can never be part of the augmenting path $AP_{\bar{\tau}}$ in the same direction unless it was traversed in the opposite direction at some prior iteration and the flow reduced so that it was no longer upper-blocked.

We say that (i, j) has become *lower-blocked* on iteration τ if $(i, j) \in AP_\tau$ and $x_{ij}^\tau = 0$. Analogously suppose $(i, j) \in AP_\tau$ were lower-blocked on iteration τ , and $(i, j) \in AP_{\bar{\tau}}$, $x_{ij}^{\bar{\tau}} \geq 0$ for $\bar{\tau} > \tau$, then there exists an l , where $\tau < l < \bar{\tau}$, such that $(i, j) \in AP_l$ and $x_{ij}^l > 0$. This follows from arguments similar to those give above.

Next we show that if, for τ , some fixed $(i, j) \in AP_\tau$ such that arc (i, j) is upper-blocked (lower-blocked) and for $\bar{\tau} > \tau$, the same $(i, j) \in AP_{\bar{\tau}}$, then

$$\eta_\tau(s, t) + 2 \leq \eta_{\bar{\tau}}(s, t).$$

We will show the result for the upper-blocked case, the other follows by a similar argument. As we have discussed previously, arc (i, j) must then be traversed in the opposite direction at some iteration $\tau < l < \bar{\tau}$ and the path lengths at iteration $\bar{\tau}$ are clearly greater than the path lengths at iteration l which are clearly greater than the path lengths at iteration τ . Clearly

$$\begin{aligned}\eta_{\bar{\tau}}(s, j) &\geq \eta_{\tau}(s, i) + \eta_{\tau}(i, j) = \eta_{\tau}(s, i) + 1 \\ \eta_{\bar{\tau}}(i, t) &\geq \eta_{\tau}(i, j) + \eta_{\tau}(j, t) = 1 + \eta_{\tau}(j, t)\end{aligned}$$

because $\eta_{\tau}(i, j) = 1$ since it is the length of an arc (i, j) . Adding the above two inequalities we get

$$\eta_{\bar{\tau}}(s, j) + \eta_{\bar{\tau}}(i, t) \geq \eta_{\tau}(s, i) + \eta_{\tau}(j, t) + 2.$$

Adding the equality $\eta_{\bar{\tau}}(j, i) = \eta_{\tau}(i, j)$ to the above inequality, we get

$$\eta_{\bar{\tau}}(s, j) + \eta_{\bar{\tau}}(j, i) + \eta_{\bar{\tau}}(i, t) \geq \eta_{\tau}(s, i) + \eta_{\tau}(i, j) + \eta_{\tau}(j, t) + 2$$

or

$$\eta_{\bar{\tau}}(s, t) \geq \eta_{\tau}(s, t) + 2$$

where $\eta_{\bar{\tau}}(s, t)$ is the number of arcs in the shortest path from s to t at iteration $\bar{\tau}$ and $\eta_{\tau}(s, t)$ is the number of arcs in the shortest path from s to t at iteration τ .

We next show that the algorithm terminates by constructing $mn/2$ augmenting paths or less, where n is the number of arcs and m is the number of nodes. If there are n arcs, clearly we can do at most n iterations before one of the arcs must be traversed again (or the maximal flow has been found). In this case the path length must increase by at least 2 from the length at the start of the cycle of n iterations. Because the maximum path length is $m - 1$, the maximum number of cycles of n iterations is $m/2$; thus the algorithm terminates by constructing $mn/2$ augmenting paths or less. ■

8.1.3 CUTS IN A NETWORK

The search for an augmenting path can be time-consuming, especially in large networks. Thus, it would be nice to be able to recognize optimality without doing an exhaustive search for an augmenting path that may not exist. It turns out that it is sometimes possible to prove that no such path exists by verifying that the conditions of the *Ford-Fulkerson Max-Flow Min-Cut* Theorem (Theorem 8.8) are satisfied. These conditions make use of the notion of a cut and its value.

Definition (Cut): A cut $\mathcal{Q} = (\mathcal{X}, \bar{\mathcal{X}})$ in a network is a partition of the node set into two nonempty subsets \mathcal{X} and its complement $\bar{\mathcal{X}} = \mathcal{N}d \setminus \mathcal{X}$. If \mathcal{X} contains the source node s and $\bar{\mathcal{X}}$ contains the destination node t , the cut is said to *separate* node s from node t .

Definition (Cut Value): If $0 \leq x_{ij} \leq h_{ij}$ for all $(i, j) \in \mathcal{A}$, then the *cut value* C of a cut $\mathcal{Q} = (\mathcal{X}, \bar{\mathcal{X}})$ is the sum of the capacities of the arcs that start in set \mathcal{X} and end in set $\bar{\mathcal{X}}$; i.e.,

$$C = \sum_{\{(i,j) \in \mathcal{A} \mid i \in \mathcal{X}, j \in \bar{\mathcal{X}}\}} h_{ij}. \quad (8.14)$$

Definition (Saturated Arc): An arc is said to be *saturated* if it is used to full capacity, i.e., $x_{ij} = h_{ij}$.

LEMMA 8.7 (Flow Value \leq Cut Value) *The flow value F , of any feasible solution, is less than or equal to the value C of any cut separating the source s from the destination t .*

Proof. Let $\mathcal{Q} = (\mathcal{X}, \bar{\mathcal{X}})$ be any cut separating the source node s from the destination node t . Next sum the conservation relations (8.3), (8.4), and (8.6) for all the nodes k of the set \mathcal{X} . By definition, $x_{ji} = -x_{ij}$. Therefore variables x_{ij} and x_{ji} cancel if both i and j are in \mathcal{X} . What remains is only the sum,

$$F = \sum_{\{(k,j) \in \mathcal{A} \mid k \in \mathcal{X}, j \in \bar{\mathcal{X}}\}} x_{kj} - \sum_{\{(i,k) \in \mathcal{A} \mid k \in \mathcal{X}, i \in \bar{\mathcal{X}}\}} x_{ik}. \quad (8.15)$$

Next noting that $0 \leq x_{ij} \leq h_{ij}$, we get

$$F \leq \sum_{\{(k,j) \in \mathcal{A} \mid k \in \mathcal{X}, j \in \bar{\mathcal{X}}\}} h_{kj} - 0 = C. \quad (8.16)$$

■

THEOREM 8.8 (Ford-Fulkerson: Min-Cut = Max-Flow) *The max-flow value is equal to the min-cut value.*

Proof. Lemma 8.7 says that any flow value F is a lower bound for any cut value C and C is an upper bound for F . Therefore it follows that if we can find an $F = F_0$, $C = C_0$ such that $F_0 = C_0$, then $\max F = F_0$ and $\min C = C_0$, the theorem will then follow.

Assume we have found a maximal flow $x = x^o$ with flow value $F = F_0$. Create a cut $\mathcal{Q} = (X, \bar{X})$ by placing in the set X all the nodes that can be reached from the source node by one or more flow-augmenting paths. In the set \bar{X} , place all the remaining nodes. The set \bar{X} must be nonempty and contain t because if t belonged to X there would be a flow-augmenting path from s to t that could be used to augment the flow contrary to the assumption that the flow is maximal. All directed arcs (i, j) , joining a node $i \in X$ to a node $j \in \bar{X}$, must be saturated; i.e., $x_{ij}^o = h_{ij}$, for otherwise j could be reached from the origin via some flow-augmenting path

contrary to the construction of the cut. Moreover all directed arcs (j, i) , joining a node $j \in \bar{X}$ to a node $i \in X$, must have the arc flow $x_{ji}^o = 0$, for otherwise j could have been reached from the origin via some flow-augmenting path passing through i implying that $j \in X$, contrary to the construction of the cut.

We next show that the cut value of \mathcal{Q} is $C_0 = F_0$. Sum the conservation relations (8.3), (8.4), and (8.6) for all the nodes k of the set X . Variables x_{ij} and x_{ji} cancel if both i and j are in X . What remains is only the sum,

$$F_0 = \sum_{\{(k,j) \in \mathcal{A} | k \in X, j \in \bar{X}\}} x_{kj} - \sum_{\{(i,k) \in \mathcal{A} | k \in X, i \in \bar{X}\}} x_{ik}. \quad (8.17)$$

Since the (i, j) are all the arcs of the cut \mathcal{Q} , and since, as we have just shown, $x_{ij}^o = h_{ij}$ for $i \in X$ and $j \in \bar{X}$, and $x_{ij}^o = 0$ for $i \in \bar{X}$ and $j \in X$, we have

$$F_0 = \sum_{\{(k,j) \in \mathcal{A} | k \in X, j \in \bar{X}\}} x_{kj}^o = \sum_{\{(k,j) \in \mathcal{A} | k \in X, j \in \bar{X}\}} h_{kj} = C_0, \quad (8.18)$$

and the theorem $\min C = \max F$ is proved. ■

▷ **Exercise 8.5 (Duality)** Show that the dual of the maximal flow problem is the min-cut problem. *Hint:* Set up the maximal flow problem as a linear program. Set up the dual by letting u_j be the multipliers corresponding to the nodes and let w_{ij} be the multipliers corresponding to the upper bounds on arc flows, and show that the system is redundant. Show that the redundancy is such that we can set $u_t = 0$, where t is the destination node; show that this implies that $u_s = 1$, where s is the source node. Next show that all the remaining multipliers are each 0 or 1. Then show that for arc (i, j) , we have $w_{ij} = 1$ if and only if $u_i = 1$ and $u_j = 0$. Use this last result to define the cut.

▷ **Exercise 8.6** If each such arc (i, j) has a lower bound l_{ij} , not necessarily zero, on the arc flow x_{ij} , then show that the cut value is

$$C = \sum_{\{(i,j) \in \mathcal{A} | i \in X, j \in \bar{X}\}} h_{ij} - \sum_{\{(j,i) \in \mathcal{A} | j \in \bar{X}, i \in X\}} l_{ij}. \quad (8.19)$$

8.2 SHORTEST ROUTE

The shortest-route problem is that of finding the minimum total “distance” along paths in an undirected connected network from the source (or origin) node $s = 1$ to the destination node $t = m$. The distances along arcs in the network can be measured in some units such as actual miles, the cost or time to go between nodes, and so on.

A simple method to solve such a problem assuming all arc distances are *non-negative distances* (or costs) is a branching-out iterative procedure that fans out from the source. Starting from the source it always picks on the next iteration the

closest node i to the source either directly or via a node whose minimum distance to the source has already been determined and records its distance. The algorithm is terminated when the shortest distance from the source node to the destination node is recorded.

LEMMA 8.9 (Validity of Dijkstra's Algorithm) *Dijkstra's Algorithm (see Linear Programming 1) finds the shortest path from the source to all nodes in the network.*

Proof. Recall that at each iteration τ , the nodes are partitioned into two subsets:

- The set of nodes \mathcal{S} such that the shortest path from the source node $s = 1$ to each node $j \in \mathcal{S}$ has been determined, and $p_j \in \mathcal{S}$ is the predecessor to node j along the shortest path to node j .
- The remaining set of nodes $\mathcal{N}d \setminus \mathcal{S}$ such that the shortest path from each $k \in \mathcal{N}d \setminus \mathcal{S}$ to the source s has been determined via the nodes of \mathcal{S} and $p_k \in \mathcal{S}$ is the node along the path that is its predecessor node.

On iteration $\tau + 1$, the algorithm determines $k^* = \operatorname{argmin}_{j \in \mathcal{N}d \setminus \mathcal{S}} z_j$. It next augments the set \mathcal{S} to $\mathcal{S} \cup \{k^*\}$ and deletes k^* from $\mathcal{N}d \setminus \mathcal{S}$. Next it adjusts z_k for the remaining nodes $k \in \mathcal{N}d \setminus \mathcal{S}$

$$\text{if } z_{k^*} + d_{k^*k} < z_k, \text{ then set } z_k = z_{k^*} + d_{k^*k} \text{ and } p_k = k^*.$$

The proof then consists of showing that the shortest path from the source s to k^* is through $p_{k^*} \in \mathcal{S}$. Suppose that the shortest path was not through p_{k^*} but instead through some other node $i \in \mathcal{N}d \setminus \mathcal{S}$. By our selection procedure, $z_{k^*} \leq z_i$ and the distance along the path from i to k^* is nonnegative. Hence it follows that the distance to the source via i is greater than or equal to the distance to the source via p_{k^*} . The algorithm then uses node k^* to modify the labels of adjacent nodes belonging to $\mathcal{N}d \setminus \mathcal{S}$ and hence restores the property of $\mathcal{N}d \setminus \mathcal{S}$. ■

▷ **Exercise 8.7** Construct an example to show that Dijkstra's Algorithm can fail if there are negative arc distances. Construct an example with some negative arc distances but where the sum of distances around every cycle is nonnegative. Construct an example with some negative arc distances but where the sum of distances around every cycle is nonnegative. Demonstrate that Dijkstra's Algorithm in the latter case finds the shortest route from the source to destination.

8.3 MINIMUM COST-FLOW PROBLEM

The minimum cost-flow problem is to find flows x_{ij} through a directed network $G = (\mathcal{N}d, \mathcal{A}c)$ with m nodes indexed $1, \dots, m$ and n arcs such that the total cost

of the flows is minimized. This is a standard linear program with a very special structure:

$$\begin{aligned} \text{Minimize} \quad & \sum_{(i,j) \in \mathcal{A}} c_{ij} x_{ij} = z \\ \text{subject to} \quad & \sum_{i \in Af(k)} x_{ki} - \sum_{j \in Bf(k)} x_{jk} = b_k \quad \text{for all } k \in \mathcal{N}d, \\ & l_{ij} \leq x_{ij} \leq h_{ij} \quad \text{for all } (i,j) \in \mathcal{A}, \end{aligned} \tag{8.20}$$

where c_{ij} is the cost per unit flow on the arc (i, j) ; b_k is the net flow at node k ; l_{ij} is a lower bound on the flow in arc (i, j) ; h_{ij} is an upper bound on the flow in arc (i, j) ; $Af(k) = \{j \in \mathcal{N}d \mid (k, j) \in \mathcal{A}\}$; and $Bf(k) = \{i \in \mathcal{N}d \mid (i, k) \in \mathcal{A}\}$. Note that b_k takes on values that depend on the type of node k :

$$b_k \text{ is } \begin{cases} > 0 & \text{if } k \text{ is a source (supply) node;} \\ < 0 & \text{if } k \text{ is a destination (demand) node;} \\ = 0 & \text{if } k \text{ is a node for transshipment only.} \end{cases}$$

The Network Simplex Method for solving the minimum cost flow applied is described in detail in *Linear Programming 1*.

8.4 NOTES & SELECTED BIBLIOGRAPHY

Network optimization theory is a very beautiful field grounded on graph-theoretical methodology. For further details on networks and their applications, see, for example, Ahuja, Magnanti, & Orlin [1993], Bertsekas [1991], Ford & Fulkerson [1962], Lawler [1976], and *Linear Programming 1*.

The min-cut max-flow theorem was first observed to be true for planar networks at RAND in 1954 and published by Dantzig & Fulkerson [1956]. Soon thereafter the theorem was established by Ford & Fulkerson [1956] for general networks. It was also discovered independently by Elias, Feinstein, & Shannon [1956]. A comprehensive treatment of the maximal-flow problem and related matters can be found in Ford & Fulkerson [1962].

The classical augmenting-path method for finding a maximum flow through a network was developed by Ford & Fulkerson [1957] based on earlier work by Kuhn [1955] and Egerváry [1931]. Fulkerson & Dantzig [1955] and Dantzig & Fulkerson [1956] developed a tree method for solving maximal flow problems which is also described in Dantzig [1963]. The approach constructs two subtrees, one branching out from the source and the other branching out from the destination so that every intermediate node is reached by just one of the trees. Then a connecting arc between the two trees and an associated path from source to destination is found and the maximum flow along the path is determined.

J. Edmonds & R. M. Karp [1972] showed that an augmenting-path method called *first-labeled first-scanned* finds a maximum flow in no more than $mn/2$ iterations, where n is the number of arcs and m is the number of nodes in the network, regardless of what the upper bounds h_{ij} are on the arcs. This method then finds the maximal flow in $O(n^2m)$ operations because it can be shown that each iteration of the augmenting-path method takes only $O(n)$ comparisons to find an augmenting path. A proof of Theorem 8.6 can also

be found in Edmonds & Karp [1972]. Around the same time as Edmonds & Karp's results, Dinic [1970] independently designed a faster algorithm that requires $O(m^2n)$ operations. Later Malhotra, Kumar, & Maheshwari [1978] developed an algorithm that requires $O(m^3)$ operations. For networks that have $n \ll m^2$, an algorithm designed by Galil [1978] takes $O(m^{5/3}n^{2/3})$, and an algorithm designed by Sleator [1980] takes only $O(nm \log m)$ steps.

Shortest-path problems come up often in practice and arise as subproblems in many network problems. Dantzig was among the first to propose a method for finding the shortest path from a source node to a destination node in a network; see Dantzig [1960a] based on an earlier RAND research memorandum. About the same time, Dijkstra [1959] independently proposed a refined version of the same algorithm for finding the shortest directed paths from a node to all other nodes. Both forms of the algorithm require at most $m(m-1)/2$ comparisons; See also Bellman [1958]. Independently, Whiting & Hillier [1960] also developed a shortest route algorithm. Johnson [1977] has shown that this bound can be further reduced to $O(n \log_k m)$ operations, where $k = \max(2, n/m)$; see also Denardo & Fox [1979], Dial [1969], Moore [1959], and Pape [1974]. A summary of various classical algorithms can be found in Gallo & Pallottino [1988]. Improvements have continued to be made in shortest-path algorithms; see, for example, Ahuja, Mehlhorn, Orlin, & Tarjan [1990], Fredman & Willard [1994], Gabow & Tarjan [1989], Goldberg [1993], and Goldberg & Radzik [1993]. Under the assumption that arc lengths are integers between 0 and L where $L \geq 2$, Ahuja, Mehlhorn, Orlin, & Tarjan's algorithm runs in $O(n + m\sqrt{\log L})$. For theory and experimental evaluation of shortest-path algorithms, see Cherkassky, Goldberg, & Radzik [1996]. In their paper they show that some algorithms behave in exactly the same way on two networks, one of which is obtained from the other by replacing the arc lengths by the reduced costs with respect to a potential function; that is, the algorithms are *potential-invariant*. This implies, for example, that a shortest-path problem with no negative cycles is equivalent to one with nonnegative arc lengths.

For additional details, including implementation details on the Network Simplex Method, see, Ali, Helgason, Kennington, & Lall, [1978], Bradley, Brown, & Graves [1977], Chvátal [1983], Cunningham [1979], and Mulvey [1978].

An example of cycling in the Network Simplex Method can be found in Cunningham & Klincewicz [1983]. To the authors' knowledge, cycling, as a result of degeneracy, has not been encountered on any practical problem. It is not known if cycling can occur in minimum-cost network-flow problems if the entering variable is chosen based on the usual rule of picking the one which has the most negative reduced cost. The interested reader can find strategies used to prevent the possibility of cycling, for example, in Bazaraa, Jarvis, & Sherali [1990] and Chvátal [1983].

The Network Simplex Method is very efficient in practice; in fact, this network adaptation of the Simplex Method for networks is typically 200 to 300 times faster than the Simplex Method applied to a general linear programs of the same dimensions encountered in practice. However, pathological examples can be constructed in which the Network Simplex Method can take a very large number of iterations. Zadeh [1973] has constructed a sequence of transshipment problems such that the k th problem has only $2k + 2$ nodes but if we choose the incoming arc by picking the most negative reduced cost, the Network Simplex Method takes $2^k + 2^{k-2} - 2$ iterations.

An area that we have not covered is that of project planning, scheduling, and coordination of various activities. Methods to do this are called PERT (Program Evaluation and Review Techniques) and CPM (Critical Path Method). Many references exist for such methods; see Hillier & Lieberman [1995]. One such reference relating this to networks is

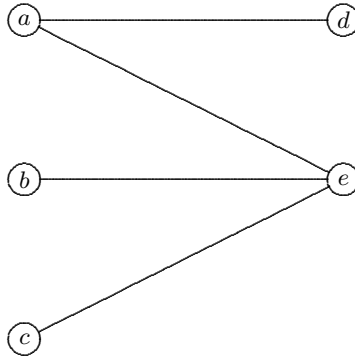


Figure 8-4: Example to Show Matching

by Elmaghraby [1977].

8.5 PROBLEMS

- 8.1 Bertsekas [1991]. Consider a *bipartite graph* that consists of two sets of nodes \mathcal{S} and \mathcal{T} such that every arc in the graph has its tail node in \mathcal{S} and its head node in \mathcal{T} .
- A *matching* is defined to be a subset of the arcs such that no two arcs are incident to the same node; for example, arc (a, e) in Figure 8-4. A *maximal matching* is defined to be a matching with a maximal number of arcs; for example, arcs (a, d) and (c, e) in Figure 8-4. Show that the problem of finding a maximal matching can be formulated as a max-flow problem.
 - A *cover* \mathcal{C} is defined to be a subset of $\mathcal{S} \cup \mathcal{T}$ such that for each arc (i, j) in the graph either $i \in \mathcal{C}$ or $j \in \mathcal{C}$ (or both). A *minimal cover* is defined to be a cover with a minimal number of nodes. Show that the number of arcs in a maximal matching and the number of nodes in a minimal cover are equal. *Hint:* Use the max-flow/min-cut theorem.
 - Consider an $n \times n$ assignment problem of assigning persons to jobs where not every man is eligible for every job. Prove that this assignment problem is not feasible if and only if there is a subset of jobs q that are the only jobs p persons are eligible for, and $p > q$.
- 8.2 *Minimum-Cost Multi-Commodity Flow Problem.* Consider a network with M source-terminal pairs (s_k, t_k) and let the required flow value between s_k and t_k be F_k for $k = 1, \dots, M$; that is, the flow between each pair (s_k, t_k) can be thought of as the flow of a different commodity. Suppose that each arc (i, j) of the network has arc capacity h_{ij} that is an upper bound on the total flow of all commodities on the directed arc (i, j) . Let c_{ij} be the cost per unit total flow on arc (i, j) . Assuming that all arcs are directed arcs and all flows are nonnegative on these arcs, the goal is to find a minimum-cost feasible flow that

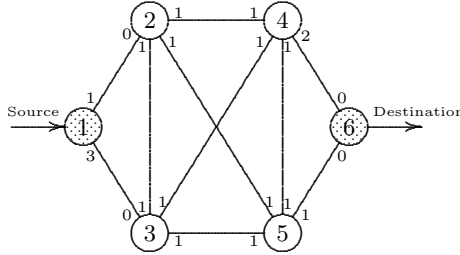


Figure 8-5: Data for a Max-Flow Problem

- can be decomposed into M feasible flows F_k . Formulate this problem.
- 8.3 Set up the maximal-flow problem shown in Figure 8-5 as a minimum cost-flow problem and solve it by the Network Simplex Method.
- 8.4 Given any feasible solution x , not necessarily integral, to the minimum cost-flow problem (8.20) with integer b_i and integer lower and upper bounds on x_{ij} , show that it is possible to find an integral feasible solution y that is a close approximation to x in the sense that

$$|y_{ij} - x_{ij}| \leq 1, \quad \text{for all } (i, j) \in \mathcal{A}.$$

- 8.5 *David Morton [1995].* Let $\mathcal{A}^*c \subseteq \mathcal{A}c$. Let z_1^* be the optimal solution to

$$\begin{aligned} &\text{Maximize} && x_{ts} = z_1 \\ &\text{subject to} && \sum_{j \in Af(k)} x_{kj} - \sum_{i \in Bf(k)} x_{ik} = 0 \quad \text{for all } k \in \mathcal{N}d, \\ &&& 0 \leq x_{ij} \leq u_{ij} \quad \text{for all } (i, j) \in \mathcal{A}c \setminus \mathcal{A}^*c \\ &&& x_{ij} \leq 0 \quad \text{for all } (i, j) \in \mathcal{A}^*c \end{aligned}$$

and let z_2^* be the optimal solution to

$$\begin{aligned} &\text{Maximize} && x_{ts} - \sum_{(i,j) \in \mathcal{A}^*c} x_{ij} = z_2 \\ &\text{subject to} && \sum_{j \in Af(k)} x_{kj} - \sum_{i \in Bf(k)} x_{ik} = 0 \quad \text{for all } k \in \mathcal{N}d, \\ &&& 0 \leq x_{ij} \leq u_{ij} \quad \text{for all } (i, j) \in \mathcal{A}c \setminus \mathcal{A}^*c. \end{aligned}$$

Prove $z_1^* = z_2^*$.

- 8.6 *Ph.D. Comprehensive Exam, September 25, 1976, at Stanford.* Given a network flow problem

$$Ax = b, \quad x \geq 0 \tag{8.21}$$

where each column has exactly two nonzero coefficients of which one is +1 and the other -1. It is assumed b has all integer components and that the program

is feasible. The objective function is of the form

$$\sum_{j=1}^n \phi_j(x_j) = \min, \quad (8.22)$$

where $\phi_j(x_j)$ are *convex functions*.

- (a) Show that in general the optimal solution can have fractional values for x_j .
 (b) Suppose an optimal solution is desired in integers. Prove that if the $\phi_j(x_j)$ are replaced by the broken line fit $\bar{\phi}_j(x_j)$ where the breakpoints occur at integral values of x_j , that system (8.21) and (8.23),

$$\sum_{j=1}^n \bar{\phi}_j(x_j) = \min, \quad (8.23)$$

solves to yield an integer solution if unique.

- (c) How would you solve (8.21) and (8.23) for an optimal integral solution?
 (d) Consider the program

$$\begin{aligned} \sum_{j=1}^n \phi_j(x_j) &= z \quad (\min) \\ F_i(x) &= 0 \quad \text{for } i = 1, \dots, m, \end{aligned}$$

where $F_i(x)$ are general convex functions in x and $\phi_j(x_j)$ are convex in x_j . Let $\bar{\phi}_j(x_j) \geq \phi_j(x_j)$ and $\bar{\phi}_j(x_j) = \phi_j(x_j)$ for all integral x_j . Prove, if the system

$$\begin{aligned} \sum_{j=1}^n \bar{\phi}_j(x_j) &= z \quad (\min) \\ F_i(x) &= 0 \quad \text{for } i = 1, \dots, m, \end{aligned}$$

solves and yields an integer solution, that this is the optimal integral solution to

$$\begin{aligned} \sum_{j=1}^n \phi_j(x_j) &= z \quad (\min) \\ F_i(x) &= 0 \quad \text{for } i = 1, \dots, m. \end{aligned}$$

- (e) What is the relationship between questions (c) and (d).

- 8.7 *Ph.D. Comprehensive Exam, September 24, 1988, at Stanford.* Consider a directed graph $(\mathcal{N}d, \mathcal{A}c)$ with m nodes and n arcs. Assume each arc $(i, j) \in \mathcal{A}c$ can be traversed in unit time at a cost c_{ij} . Invent an $O(m^2n)$ running-time algorithm for finding a simple circuit (i.e., a directed simple cycle) in the graph for which the ratio of the cost to traverse the circuit to the time to traverse it is as small as possible. Be sure to justify your answer.
- 8.8 *Ph.D. Comprehensive Exam, September 23, 1989, at Stanford.* Let $\mathcal{N}d$ be the node set and let $\mathcal{A}c$ be the arc set of a directed network $(\mathcal{N}d, \mathcal{A}c)$. Let s and t be two nodes. For arc $(i, j) \in \mathcal{A}c$ let d_{ij} (which may be positive, zero, or negative) be its specified "length."

We desire to find the shortest directed route from s to t that does not use any arc more than once. Is the following linear program a proper formulation of this shortest-route problem?

$$\begin{aligned} & \text{Minimize} && \sum_{(i,j) \in \mathcal{A}} d_{ij} x_{ij} \\ & \text{subject to} && \sum_i x_{ij} = \sum_k x_{jk}, \quad j \neq s, t \\ & && \sum_k x_{sk} = 1 \\ & && x_{ij} \geq 0. \end{aligned}$$

If you are not satisfied with the formulation, state why. Can you do better? Argue your case. Comment on the difficulty of the problem; under what conditions is the problem easier?

- 8.9 *Ph.D. Comprehensive Exam, September 22, 1990, at Stanford.* Given a network $G = (V, E)$ with node-set V and edge-set E , and given specified nodes s and t , a flow x from s to t means $x = (x_j \mid j \in E) \geq 0$ such that

$$\sum_{\{j \mid H(j)=i\}} x_j - \sum_{\{j \mid T(j)=i\}} x_j = 0 \quad \text{for all } i \in V \setminus \{s, t\}, \quad (8.24)$$

where $H(j)$ denotes the head of edge j and $T(j)$ denotes the tail of edge j . The amount $f(x)$ of the flow is defined to be either the net flow out of the terminal node

$$f(x) = \sum_{\{j \mid H(j)=t\}} x_j - \sum_{\{j \mid T(j)=t\}} x_j$$

or the net flow into the source node

$$f(x) = - \sum_{\{j \mid H(j)=s\}} x_j + \sum_{\{j \mid T(j)=s\}} x_j.$$

Given lower bounds $\beta = (\beta_j \mid j \in E) \geq 0$, we say that flow x satisfies the lower bounds if $x_j \geq \beta_j$ for all $j \in E$. For $S \subseteq V$, let $\delta(S)$ be the set of directed arcs from S to $V \setminus S$ and let $\delta(V \setminus S)$ be the set of directed arcs from $V \setminus S$ to S . By definition

$$\begin{aligned} \delta(S) &= \{j \in E \mid T(j) \in S, H(j) \in V \setminus S\} \\ \delta(V \setminus S) &= \{j \in E \mid T(j) \in V \setminus S, H(j) \in S\} \end{aligned}$$

- (a) Prove algebraically that if $x = (x_j \mid j \in E)$ is a flow from s to t satisfying the lower bounds, and if $\delta(V \setminus S) = \emptyset$

$$f(x) \geq \sum_{j \in \delta(S)} \beta_j.$$

- (b) Assume that there is a directed path from t to s (i.e., there is no cut separating s from t such that $\delta(V \setminus S) = \emptyset$) and assume $x = (x_j \mid j \in E)$ is a flow from s to t satisfying the lower bounds. How can you obtain flows x' satisfying the lower bounds such that $f(x) \rightarrow -\infty$.

- (c) Give an algorithm for the following: Given a flow x from s to t of amount $f(x) > 0$ satisfying the lower bounds, find a flow x' from s to t of amount $f(x') < f(x)$ satisfying the lower bounds or else find a cut $(S, V \setminus S)$ separating s from t such that $\delta(V \setminus S) = \emptyset$ and

$$f(x') = \sum_{j \in \delta(S)} \beta_j. \quad (8.25)$$

- (d) Use (a) and (c) to prove: If G has a flow from s to t satisfying the lower bounds, and a cut $(S, V \setminus S)$ separating s from t such that $\delta(V \setminus S) = \emptyset$, then

$$F^* = \max \sum_{j \in \delta(S)} \beta_j, \quad (8.26)$$

where F^* is the min flow from s to t satisfying the lower bounds.

- (e) Given costs $c = (c_j \mid j \in E)$, the cost of flow x is

$$c^T x = \sum_{\{j \in E\}} c_j x_j$$

by definition. Given dual prices (multipliers) $y = (y_i \mid i \in V)$, define reduced costs \bar{c}_j by

$$\bar{c}_j = c_j + y_{T(j)} - y_{H(j)}$$

for $j \in E$. Let $x^* = (x_j^* \mid j \in E)$ be a flow from s to t satisfying the lower bounds, i.e., $x_j^* \geq \beta_j$. Suppose there are prices $y = (y_i \mid i \in V)$, $\bar{c}_j = c_j + y_{T(j)} - y_{H(j)}$ such that

- (•) $\bar{c}_j \geq 0$ for all $j \in E$,
- (•) $x_j^* > \beta_j \Rightarrow \bar{c}_j = 0$.

- Prove that x^* minimizes $\bar{c}^T x$ over all flows x from s to t of amount $f(x^*)$ satisfying the lower bounds.
- Prove that x^* also minimizes $c^T x$ over all flows x from s to t of amount $f(x^*)$ satisfying the lower bounds.

8.10 *Ph.D. Comprehensive Exam, September 21, 1991, at Stanford.*

König-Egerváry Theorem: Let M be a $(0, 1)$ matrix with m rows and n columns. The König-Egerváry theorem states that the largest cardinality of a set of 1s in M , no pair of which is in the same row or column, is equal to the smallest cardinality of a set of rows and columns containing all 1s in M .

Philip Hall's Theorem of Distinct Representatives: Let X_1, X_2, \dots, X_k be k given finite sets. The collection $\{x_1, x_2, \dots, x_k\}$ of one element x_i from each X_i is a *system of distinct representatives* if $x_i \in X_i$ for all i and $x_i \neq x_j$ for $i \neq j$. No such system of distinct representatives exists if and only if there exists an index set $\mathcal{N} \subset \{1, \dots, k\}$ such that the number of elements in $\cup_{i \in \mathcal{N}} X_i$ is less than the number of elements in \mathcal{N} .

Dilworth's Theorem on the Decomposition of a Finite Partially Ordered Set into Chains: For a given partial ordering \mathcal{S} , a chain of elements is a sequence s_1, s_2, \dots, s_m where the elements satisfy $s_i \leq s_{i+1}$. The minimum number of

chains, such that every element is contained in at least one chain, is equal to the maximum number of incomparable elements; i.e., where by definition s_i and s_j are incomparable elements if neither $s_i \leq s_j$ nor $s_j \leq s_i$.

(a) Illustrate the König-Egervály theorem for the matrix

$$M = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b) Do three of the following:

- Prove the König-Egervály theorem from the max-flow min-cut theorem.
- Sketch a proof of the König-Egervály theorem using linear programming duality.
- Show how Philip Hall's theorem on systems of distinct representatives is a consequence of the König-Egervály theorem.
- Show how Dilworth's theorem on the decomposition of a finite partially ordered set into chains is a consequence of the König-Egervály theorem.

GENERALIZED UPPER BOUNDS

Large-scale systems typically have a special structure that can be exploited to gain computational advantage. It is routine, in practice, to take advantage of the large percentage of zeros in a large-scale problem. In other instances, there are linear programs that have upper bounds on subsets of variables such that each variable appears in at most one subset. Such constraints are called *generalized upper bounds*.

Quite often, linear programs have a set of general constraints and a set of constraints that are upper bounds on partial sums of variables such that each variable appears in at most one of these partial sums. Such partial sum upper-bound constraints are called *generalized upper bounds* or *GUB constraints*. In practice, the GUB constraints are typically much more than the other constraints.

9.1 PROBLEM STATEMENT

Consider a linear program with $m + l$ constraints with the properties that:

1. Each variable has at most one nonzero coefficient in the last l constraints.
2. All of these nonzero coefficients are positive.
3. The last l constant terms (right-hand sides) are positive.

▷ **Exercise 9.1** Show that the variables can be rescaled so that all the coefficients in the last l equations are 1 and the constants in the last l equations are also 1.

9.2 BASIC THEORY

LEMMA 9.1 (Each GUB Set Contains at Least One Basic Variable) Given a basic solution to (9.2), then for each GUB set S_i , for $i = 1, \dots, l$, there exists at least one $j \in S_i$ such that x_j is basic.

▷ **Exercise 9.2** Prove Lemma 9.1.

▷ **Exercise 9.3** Construct an example that demonstrates that it is not necessary to have any basic variable x_j such that $j \in S_0$.

THEOREM 9.2 (Bound on the Number of GUB Sets with More Than One Basic Index) The number of sets S_i for $1 \leq i \leq l$ containing two or more basic indices is at most m .

Proof. There are $m+l$ variables that are basic. Of these, l indices are in different sets by Lemma 9.1. Thus, of the remaining m basic indices, p can be distributed to S_0 and at most $m-p$ to S_i , $i = 1 \leq i \leq l$. Thus at most $m-p$ GUB sets can have two or more basic indices. By Exercise 9.3, p can be zero. ■

Definition (Plural (or Essential) and Singleton (or Inessential)): For $1 \leq i \leq l$, the GUB set S_i is said to be a *plural* (or *essential*) set with respect to a basic set \mathcal{B} if it contains two or more basic variable indices; all other GUB sets are *singleton* (or *inessential*) sets since by Lemma 9.1 they have exactly one basic variable index. With this definition, Theorem 9.2 can be restated: the maximum number of plural sets is m .

Definition (Key Basic Index/Variable, Artificial): For each GUB set S_i we choose one basic variable index $j \in S_i$ to be designated as a *key basic index* and its corresponding basic variable x_j as a *key basic variable*. For the singleton sets, there is exactly one basic index so this choice for key basic is unique. For plural sets, the choice as to which one is designated as the key is *arbitrary*.

For an example in detached coefficient form see Table 9-1 where the basic variable are denoted by • above and the key variables are denoted by “key” below. Notice that if the columns of the basis are rearranged so that the columns corresponding to the key basic variables are moved to the right, then the basis has an identity in the lower-right corner.

9.3 SOLVING SYSTEMS WITH GUB EQUATIONS

In practice, (9.2) will have many GUB constraints; in fact, often $l \gg m$; for example, $m = 50$ and $l = 1000$. By taking advantage of the structure we shall see that

\bullet	\bullet	\bullet	\bullet	\bullet	\bullet	\bullet	\bullet	\bullet	\bullet	\bullet	\bullet
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	b	
1	0	2	0	3	4	5	1	-1	-12	15	
1	1	-1	0	2	1	4	2	-3	6	7	
0	0	0	1	0	0	0	0	0	0	0	
	1	1	1							1	
				1						1	
					1					1	
						1	1			1	
								1	1	1	
	k			k	k	k		k			
	e			e	e	e		e			
	y			y	y	y		y			

Table 9-1: An Example of GUB Constraints and Key Basic Variables

x_1	x_3	x_4	x_2	x_5	x_6	x_7	x_9
1	2	0	0	3	4	5	-1
1	-1	0	1	2	1	4	-3
0	0	1	0	0	0	0	0
	1	1	1				
				1			
					1		
						1	
							1

Table 9-2: An Example of a Reordered Basis for GUB Constraints

we solve $m \times m$ equations on each iteration of the Simplex Algorithm instead of $(m + l) \times (m + l)$ equations.

Suppose the columns of a basis for (9.2) have been reordered so that the last l columns $\begin{pmatrix} G \\ I_l \end{pmatrix}$ correspond to the key basic variables and the first m columns $\begin{pmatrix} F \\ E \end{pmatrix}$ correspond to the nonkey basic variables. The basis then has the following special form:

$$B = \begin{matrix} m & l \\ m : & \begin{pmatrix} F & G \\ E & I_l \end{pmatrix}, \end{matrix} \tag{9.3}$$

where I_l is an identity matrix of dimension l because it corresponds to the key basic variables, and E has a special form, i.e., each column has at most one 1 and all the rest 0; see, for example, Table 9-2.

Note that E can be eliminated by multiplying B on the right by the matrix

$$V = \begin{matrix} & m & l \\ m: & \left(\begin{array}{cc} I_m & 0 \\ -E & I_l \end{array} \right), \\ l: & & \end{matrix} \quad (9.4)$$

This results in

$$BV = \begin{pmatrix} F - GE & G \\ 0 & I_l \end{pmatrix} = \begin{pmatrix} W & G \\ 0 & I_l \end{pmatrix}. \quad (9.5)$$

▷ **Exercise 9.4** Show that

$$V^{-1} = \begin{pmatrix} I_m & 0 \\ E & I_l \end{pmatrix}. \quad (9.6)$$

▷ **Exercise 9.5** Show that because of the special structure of E , the matrix multiplication GE can be performed very efficiently on a computer.

Let the basic indices be j_k for $k = 1, \dots, m + l$; S_q be the q th GUB set for $1 \leq q \leq l$, S_0 be the set of indices not in any of the GUB sets; and $\bar{k}_q \in S_q$ be the key basic index in set S_q . Then the matrix $W = F - GE$, called the *working basis*, has the following form for any column k of W :

$$W_{\bullet k} = \begin{cases} A_{\bullet j_k} - A_{\bullet \bar{k}_q} & \text{if } j_k \in S_q \text{ for some } 1 \leq q \leq l; \\ A_{\bullet j_k} & \text{if } j_k \in S_0. \end{cases} \quad (9.7)$$

THEOREM 9.3 (Working Basis Is Nonsingular) *The matrix $W = F - GE$ is an $m \times m$ nonsingular matrix.*

Proof. Since B is nonsingular and V is nonsingular, their product BV is nonsingular, implying that W is nonsingular. ■

Using equation (9.5), a system of equations whose matrix of coefficients is either B or B^T can be solved very efficiently. First consider the system:

$$Bx_B = \begin{pmatrix} b \\ e \end{pmatrix}, \quad e = e = (1, 1, \dots, 1)^T, \quad (9.8)$$

where x_B , the basic solution, is of dimension $m + l$. Multiplying B on the right by V and letting $y = V^{-1}x_B$, we get

$$BVV^{-1}x_B = \begin{pmatrix} W & G \\ 0 & I_l \end{pmatrix} y = \begin{pmatrix} b \\ e \end{pmatrix}. \quad (9.9)$$

Let $y = (y_M, y_L)$ where y_M consists of the first m components of y , and y_L consists of the last l components of y . Given W^{-1} or some factorization of the working basis W , we can easily solve for y by setting $y_L = e$ and then solving

$$Wy_M = b - Ge = b - \sum_{j \in K} A_{\bullet j} \quad (9.10)$$

where K is the set of l indices corresponding to the key basic variables. Then x_B is obtained from:

$$x_B = Vy = \begin{pmatrix} y_M \\ y_L - Ey_M \end{pmatrix}. \tag{9.11}$$

Because of the special structure of E , the operation Ey_M amounts to computing partial sums of the components of y_M ; that is, the k th component of the product Ey_M is given by

$$[Ey_M]_k = \sum_{i \in \mathcal{I}_k} [y_M]_i, \tag{9.12}$$

where $\mathcal{I}_k = \{i \mid j_i \in \mathcal{B} \cap S_k, j_i \text{ not key}, i = 1, \dots, m + l\}$, and j_1, \dots, j_{m+l} are all the indices of the basic variables.

Using the Revised Simplex Method, if the incoming column $s \in S_q$ for some $1 \leq q \leq l$, determine the representation of $\begin{pmatrix} A_{\bullet s} \\ e_q \end{pmatrix}$ in terms of the basis by \bar{p} obtained as the solution to

$$B\bar{p} = \begin{pmatrix} A_{\bullet s} \\ e_q \end{pmatrix}.$$

This computation can be done in a manner similar to that for x_B . Let $\bar{p} = Vy$; then, in this case, $y_L = e_q$ and y_M is the solution to

$$Wy_M = A_{\bullet s} - Ge_q = A_{\bullet s} - G_{\bullet q}. \tag{9.13}$$

The vector \bar{p} is then obtained by computing Vy .

▷ **Exercise 9.6** How would the computations for determining \bar{p} change if the incoming column s is in the set S_0 ?

Similarly we can easily compute the reduced costs $\sigma_N = c_N - N^T\pi$ by first solving $B^T\pi = c_B$. In order to do this, first multiply through by V^T to give $V^TB^T\pi = V^Tc_B$. This gives

$$\begin{pmatrix} W^T & 0 \\ G^T & I_l \end{pmatrix} \begin{pmatrix} \pi_M \\ \pi_L \end{pmatrix} = \begin{pmatrix} I_m & -E^T \\ 0 & I_l \end{pmatrix} c_B = \begin{pmatrix} [c_B]_M - E^T[c_B]_L \\ [c_B]_L \end{pmatrix}, \tag{9.14}$$

where once again the subscripts M and L on c_B imply the first m components and the last l components of c_B , respectively. As before, the special structure of E makes the multiplication $E^T[c_B]_L$ on the right-hand side of (9.14) easy, because each column of E (or row of E^T) has at most one nonzero component, which is unity. For notational convenience, let $y = E^T[c_B]_L$, then

$$y_i = \begin{cases} [c_B]_{m+k} & \text{if } j_i \in S_k; \\ 0 & \text{otherwise;} \end{cases} \tag{9.15}$$

for $i = 1, \dots, l$. Given W^{-1} , or some factorization of W , the system (9.14) is easy to solve. The vector π_M is first obtained by solving $W^T\pi_M = [c_B]_M - E^T[c_B]_L$ and then π_L is obtained by substitution from the last l equations in (9.14) as

$$\pi_L = [c_B]_L - G^T\pi_M. \tag{9.16}$$

Once π is obtained, the reduced costs σ_N can be obtained as follows:

$$\sigma_N = c_N - N^T \pi = c_N - A_{\bullet N}^T \pi_M - H_{\bullet N}^T \pi_L, \quad (9.17)$$

where H is the matrix of coefficients of the last l equations. Since the columns of H are all zeros except possibly one equal to $+1$, the third term in Equation (9.17) can be computed very efficiently.

▷ **Exercise 9.7** Show how to compute $(H_{\bullet N})^T \pi_L$ efficiently.

9.4 UPDATING THE BASIS AND WORKING BASIS

So far, we have shown, given a factorization of W , that the steps of the Simplex Algorithm can be performed very efficiently. At each iteration of the Simplex Algorithm we need only maintain a factorization of W that is a matrix of size $m \times m$ instead of a factorization of B that is a matrix of size $(m+l) \times (m+l)$ where l is the number of the GUB constraints, which can be very large relative to m .

Now we need only show that we can efficiently obtain a new basis representation \bar{B} from the old basis $B = \begin{pmatrix} F & G \\ E & I_l \end{pmatrix}$ at the next iteration such that the inverse (or factorization) of \bar{W} can be computed efficiently from the inverse (or factorization) of W . For simplicity of exposition, let j_r be the index of the outgoing variable where r is the r th column of B and let s be the index of the incoming variable where $s \in S_i$ for some $i = 1, \dots, l$ or $s \in S_0$. Then $\begin{pmatrix} A_{\bullet s} \\ d^s \end{pmatrix}$ is the incoming column where

$$d^j = \begin{cases} e_k & \text{a unit } l\text{-vector if } j \in S_k \text{ for some } k = 1, \dots, l, \\ 0 & \text{a zero } l\text{-vector if } j \in S_0. \end{cases} \quad (9.18)$$

There are three cases to consider depending on whether x_{j_r} is a non-key basic variable, x_{j_r} is a key basic variable belonging to a singleton set, or x_{j_r} is a key basic variable belonging to a plural set.

Case 1: x_{j_r} leaving the basis is non-key. In this case $1 \leq r \leq m$ (because of the reordering of the basic columns, the non-key basics are in the first m columns of B). The new basis \bar{B} is given by

$$\bar{B} = \begin{pmatrix} \bar{F} & G \\ \bar{E} & I_l \end{pmatrix}, \quad \text{with} \quad \begin{aligned} \bar{F} &= F + (A_{\bullet s} - A_{\bullet j_r})e_r^T, \\ \bar{E} &= E + (d^s - d^{j_r})e_r^T, \end{aligned} \quad (9.19)$$

where e_r is an m -dimensional unit vector whose r th component is 1 and all other components are 0. Now \bar{E} can be eliminated from the matrix \bar{B} in (9.19) by multiplying \bar{B} on the right by the matrix

$$\bar{V} = \begin{pmatrix} I_m & 0 \\ -\bar{E} & I_l \end{pmatrix}. \quad (9.20)$$

Thus

$$\bar{B}\bar{V} = \begin{pmatrix} \bar{F} - G\bar{E} & G \\ 0 & I_l \end{pmatrix} = \begin{pmatrix} \bar{W} & G \\ 0 & I_l \end{pmatrix}, \quad (9.21)$$

where

$$\begin{aligned} \bar{W} &= \bar{F} - G\bar{E} \\ &= W + [(A_{\bullet s} - Gd^s) - W_{\bullet r}]e_r^T, \quad W = F - GE \end{aligned} \quad (9.22)$$

Equation (9.22) implies that \bar{W} is obtained from W by substituting for its r th column $W_{\bullet r}$, the column $A_{\bullet s} - Gd^s$, a rank-one change. Thus, the inverse of \bar{W} can be easily obtained from the inverse of W ; or the LU factors of \bar{W} can be easily obtained from the LU factors of W (see *Linear Programming 4* for details). For the purpose of examples in this section, we shall use W^{-1} to obtain \bar{W}^{-1} . Letting

$$\bar{h} = W^{-1}(A_{\bullet s} - Gd^s) = W^{-1}h, \quad (9.23)$$

we have

$$\bar{W}^{-1} = (I + (\bar{h} - e_r)e_r^T)^{-1}W^{-1} = \left(I - \frac{1}{\bar{h}_r}(\bar{h} - e_r)e_r^T\right)W^{-1}. \quad (9.24)$$

▷ **Exercise 9.8** Show that $W_{\bullet r} = F_{\bullet r} - Gd^{j_r}$ and verify (9.22).

▷ **Exercise 9.9** Verify (9.24).

Case 2: x_{j_r} leaving the basis is key in a singleton set. In this case the index r satisfies $m < r \leq m + l$ (because of the reordering of the basic columns, the key basics are in the last l columns). Let $\hat{r} = r - m$, then $j_r \in S_{\hat{r}}$, the r th GUB set. Since r is from a singleton set, we know that $E_{\hat{r}\bullet}$ is a zero row. Now we also know by Lemma 9.1 that at least one variable from each GUB set S_i must be basic for $i = 1, \dots, l$. Therefore, because x_{j_r} is from a singleton set, the incoming variable x_s must be such that $s \in S_{\hat{r}}$. Then \bar{B} is the same as B except column \hat{r} of G is replaced by $A_{\bullet s}$; thus

$$\bar{B} = \begin{pmatrix} F & \bar{G} \\ E & I_l \end{pmatrix}, \quad \text{with} \quad \bar{G} = G + (A_{\bullet s} - G_{\bullet \hat{r}})e_{\hat{r}}^T \quad (9.25)$$

where $e_{\hat{r}}$ is an l -dimensional unit vector whose \hat{r} th component is 1 and all other components are 0. Note that $G_{\bullet \hat{r}} = A_{\bullet j_r}$ in the equation (9.25). Because, in this case, there is no change to E , the matrix \bar{V} used to multiply \bar{B} on the right in order to eliminate E from \bar{B} is the same as V , i.e.,

$$\bar{V} = V = \begin{pmatrix} I_m & 0 \\ -E & I_l \end{pmatrix}. \quad (9.26)$$

Thus

$$\bar{B}\bar{V} = \begin{pmatrix} F - \bar{G}E & \bar{G} \\ 0 & I_l \end{pmatrix} = \begin{pmatrix} \bar{W} & \bar{G} \\ 0 & I_l \end{pmatrix}, \quad (9.27)$$

where

$$\bar{W} = F - GE - (A_{\bullet s} - A_{\bullet j_r})e_{\hat{r}}^T E = W$$

since $e_{\hat{r}}^T E = E_{\hat{r}\bullet} = 0$, a zero-row vector because x_{j_r} is a key basic variable in a singleton set. Thus, there is no change to the working basis W .

Case 3: x_{j_r} leaving the basis is key in a plural set. In this case the index r satisfies $m < r \leq m + l$ (because of the reordering of the basic columns, the key basics are in the last l columns). Let $\hat{r} = r - m$, then $j_r \in S_{\hat{r}}$, the r th GUB set. Recall that the selection of a variable as a key basic variable was arbitrary. Therefore this case can be handled by redesignating one of the other basic variables belonging to the plural set as the key basic and then applying Case 1. To do this first select a column k from the first m columns of B with the property that $j_k \in S_{\hat{r}}$ and designate j_k as the new key basic variable for set $S_{\hat{r}}$. Then x_{j_r} is no longer a key basic variable and the basis can be updated by Case 1.

To redesignate $j_k \in S_{\hat{r}}$ as the key basic variable instead of $j_r \in S_{\hat{r}}$, we interchange columns k and r of the basis B to obtain

$$\tilde{B} = \begin{pmatrix} \tilde{F} & \tilde{G} \\ E & I_l \end{pmatrix}, \quad \text{where} \quad \begin{aligned} \tilde{G} &= G + (F_{\bullet k} - G_{\bullet \hat{r}})e_{\hat{r}}^T \\ \tilde{F} &= F + (G_{\bullet \hat{r}} - F_{\bullet k})e_k^T, \end{aligned} \quad (9.28)$$

where $e_{\hat{r}}$ is an l -dimensional unit vector whose \hat{r} th component is 1 and all other components are 0, and e_k is an m -dimensional unit vector whose k th component is 1 and all other components are 0. The updated matrix E is the same E because a 1 in column k is replaced by another 1 from the identity matrix column \hat{r} . Thus the matrix \tilde{V} used to multiply \tilde{B} on the right to eliminate E from \tilde{B} is the same as V , i.e.,

$$\tilde{V} = V = \begin{pmatrix} I_m & 0 \\ -E & I_l \end{pmatrix}. \quad (9.29)$$

Multiplying \tilde{B} on the right by $\tilde{V} = V$ we get

$$\tilde{B}\tilde{V} = \begin{pmatrix} \tilde{F} - \tilde{G}E & \tilde{G} \\ 0 & I_l \end{pmatrix} = \begin{pmatrix} \tilde{W} & \tilde{G} \\ 0 & I_l \end{pmatrix}. \quad (9.30)$$

The matrix \tilde{W} is related to the original matrix W in a very simple way. The original nonkey columns $W_{\bullet i}$ for $j_i \in S_{\hat{r}}$ are:

$$W_{\bullet i} = F_{\bullet i} - G_{\bullet \hat{r}} \quad \text{for } j_i \in S_{\hat{r}}.$$

including non-key column k , which will be designated as the new key variable. Switching columns k and r in the basis B means that we switch columns $F_{\bullet k}$ and $G_{\bullet \hat{r}}$. This results in

$$\begin{aligned}\widetilde{W}_{\bullet k} &= G_{\bullet \hat{r}} - F_{\bullet k} = -W_{\bullet k} \\ \widetilde{W}_{\bullet i} &= F_{\bullet i} - F_{\bullet k} = W_{\bullet i} - W_{\bullet k} \quad \text{for } j_i \in S_{\hat{r}} \text{ and } i \neq k.\end{aligned}$$

In matrix notation, the matrix \widetilde{W} can be represented as

$$\widetilde{W} = WT \tag{9.31}$$

where T is a row elementary matrix whose k th row, $T_{k\bullet}$, is defined by:

$$T_{ki} = \begin{cases} -1 & \text{if } j_i \in S_{\hat{r}}; \\ 0 & \text{otherwise.} \end{cases} \tag{9.32}$$

That is, T is an elementary matrix of the row form with -1 at the intersection of the k th row and diagonal and -1 or 0 elsewhere in the k th row.

Once the new \widetilde{B} is obtained, we apply Case 1 since now the outgoing variable is no longer a key basic variable.

Once again, it is possible to update W^{-1} to \widetilde{W}^{-1} or obtain the LU factors of \widetilde{W} from the LU factors of W . In order to obtain \widetilde{W}^{-1} , first compute $T^{-1}W^{-1} = TW^{-1}$ (see Exercise 9.11). Next use Equation (9.24) to apply Case 1 to obtain the final updated inverse.

For updating the LU factors, let $h = A_{\bullet s} - Gd^s$, apply Case 1 with r being replaced by k , $1 \leq k \leq m$, to obtain

$$\bar{W} = \left(W + (h - [WT]_{\bullet k})e_k^T \right) T, \tag{9.33}$$

which is in a form suitable for updating the LU factorization (see *Linear Programming 4*).

▷ **Exercise 9.10** Derive the general form $\widetilde{W} = WT$ of equation (9.31).

▷ **Exercise 9.11** Show that $T^{-1} = T$.

LEMMA 9.4 ($T^2 = I$) *Every matrix T that is a row (or column) elementary matrix where the diagonal element is -1 has the property that $T^2 = I$.*

▷ **Exercise 9.12** Prove that the column elementary matrix of Lemma 9.4 is of the form

$$T = I - 2e_k e_k^T + u e_k^T - u_k e_k e_k^T \tag{9.34}$$

for some vector u .

- ▷ **Exercise 9.13** Prove Lemma 9.4 using Exercise 9.12.
- ▷ **Exercise 9.14** Prove (9.33).
- ▷ **Exercise 9.15** Suppose that the variables in the last l constraints have at most one nonzero coefficient each. Show how to generalize the GUB theory to this class of problems.
- ▷ **Exercise 9.16** Suppose that instead of the last l constraints we have l sets of the constraints, each of the form

$$\sum_{j \in S_i} a_{m+i,j}^k x_j \leq b_{m+i}^k, \text{ for } k = 1, \dots, K_i, i = 1, \dots, l.$$

Show how to generalize the GUB theory to this class of problems. Illustrate for $K_i = 2$ for $i = 1, \dots, l$.

Example 9.1 (Illustration of GUB Procedure) Consider an example with $m = 3$ and $l = 5$ to minimize $z = -x_1$ subject to the following constraints in detached coefficient form where the basic columns are designated by \bullet above and the key basic columns are denoted by “key” below in (9.35).

•	•	•	•	•	•	•	•	•	•	•		
x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	b		
1	0	2	0	3	4	5	1	-1	-12	15		
1	1	-1	0	2	1	4	2	-3	6	7		
0	0	0	1	0	0	0	0	0	0	0		
										1		
										1		
										1		
										1		
										1	1	
										1	1	
										1	1	
										1	1	
	k			k	k	k		k				
	e			e	e	e		e				
	y			y	y	y		y				

The columns associated with the GUB sets S_i are:

S_0	S_1			S_2	S_3	S_4		S_5		b
$A_{\bullet 1}$	$A_{\bullet 2}$	$A_{\bullet 3}$	$A_{\bullet 4}$	$A_{\bullet 5}$	$A_{\bullet 6}$	$A_{\bullet 7}$	$A_{\bullet 8}$	$A_{\bullet 9}$	$A_{\bullet 10}$	b
0	e_1	e_1	e_1	e_2	e_3	e_4	e_4	e_5	e_5	e

Let the initial basic set be $\mathcal{B} = \{1, 2, 3, 4, 5, 6, 7, 9\}$ and let the key basic variables be x_2, x_5, x_6, x_7, x_9 . Then the basic set can be rewritten as

$$\mathcal{B} = \{j_1, j_2, \dots, j_8\} = \{1, 3, 4; 2, 5, 6, 7, 9\};$$

and, the basis is written as:

$$B = \left(\begin{array}{c|c} F & G \\ \hline E & I_5 \end{array} \right) = \left(\begin{array}{ccc|cccc} A_{\bullet 1} & A_{\bullet 3} & A_{\bullet 4} & A_{\bullet 2} & A_{\bullet 5} & A_{\bullet 6} & A_{\bullet 7} & A_{\bullet 9} \\ \hline 0 & e_1 & e_1 & e_1 & e_2 & e_3 & e_4 & e_5 \end{array} \right).$$

The working basis W is then given by

$$W = F - GE = (A_{\bullet 1} \quad A_{\bullet 3} - A_{\bullet 2} \quad A_{\bullet 4} - A_{\bullet 2}) = \begin{pmatrix} 1 & 2 & 0 \\ 1 & -2 & -1 \\ 0 & 0 & 1 \end{pmatrix}.$$

Its inverse is

$$W^{-1} = \begin{pmatrix} 1/2 & 1/2 & 1/2 \\ 1/4 & -1/4 & -1/4 \\ 0 & 0 & 1 \end{pmatrix}.$$

On solving $Bx_B = b$ using equations (9.8), (9.9), (9.10), and (9.11), we get

$$x_B = \left(3, \frac{1}{2}, 0; \frac{1}{2}, 1, 1, 1, 1 \right)^T.$$

Next using (9.14) we get the prices

$$\pi = (\pi_M, \pi_L) = \left(-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}; \frac{1}{2}, \frac{5}{2}, \frac{5}{2}, \frac{9}{2}, -2 \right)^T,$$

and from (9.17) we find that the reduced costs corresponding to x_8 and x_{10} are $(-3, -1)$, thus the smallest reduced cost occurs corresponding to x_8 . The representation, \bar{p} , of $\begin{pmatrix} A_{\bullet 8} \\ e_4 \end{pmatrix}$ in terms of the basis is obtained by solving $B\bar{p} = \begin{pmatrix} A_{\bullet 8} \\ e_4 \end{pmatrix}$. Using equation (9.13) we get

$$\bar{p} = \left(-3, -\frac{1}{2}, 0; \frac{1}{2}, 0, 0, 0, 1, 0 \right)^T.$$

Next we determine the variable leaving the basis by

$$\theta_r = \min_{[\bar{p}]_i > 0} \frac{[x_B]_i}{[\bar{p}]_i} = 1; \quad \text{which gives } r = 4 \text{ or } r = 7.$$

There is a tie, and we break it by arbitrarily choosing $r = 7$ and, in this case, $j_r = 7$. Now $j_r \in S_4$, where S_4 is a singleton set for the current basis. This is Case 2, where the working basis does not change.

We replace x_7 by x_8 as the key basic variable and thus obtain the new basic set as $\mathcal{B} = \{1, 3, 4; 2, 5, 6, 8, 9\}$. The new G matrix is given by

$$G = [A_{\bullet 2}, A_{\bullet 5}, A_{\bullet 6}, A_{\bullet 8}, A_{\bullet 9}].$$

The new basic solution is then

$$x_B = (6, 1, 0; 0, 1, 1, 1, 1).$$

The new multipliers π are given by:

$$\pi = (\pi_M, \pi_L) = \left(-\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}; \frac{1}{2}, \frac{5}{2}, \frac{5}{2}, \frac{3}{2}, -2 \right)^T.$$

Notice that π_M does not change because W has not changed and $[c_B]_M$ has not changed; also notice that only one component (the 4th) of π_L changes. After computing the reduced

costs for the nonbasic variables x_7 and x_{10} , we find that they are $(3, -1)$ and thus x_{10} is a candidate to enter the basis. Solving $B\bar{p} = \begin{pmatrix} A_{\bullet 10} \\ e_5 \end{pmatrix}$, using equation (9.13) we get

$$\bar{p} = (-1, -5, 0; 5, 0, 0, 0, 1)^T.$$

Next we determine the variable leaving the basis by

$$\theta_r = \min_{[\bar{p}]_i > 0} \frac{[x_B]_i}{[\bar{p}]_i} = 0, \quad \text{with } r = 4 \text{ or } j_r = 2.$$

The new basic set is $\mathcal{B} = \{1, 3, 4; 10, 5, 6, 8, 9\}$. However, x_2 is a key basic variable; this is Case 3. We first designate another variable in the set, x_3 , as the key basic variable (in place of x_2) so that the new basic set will be $\mathcal{B} = \{1, 10, 4; 3, 5, 6, 8, 9\}$. This is done by postmultiplying W by T , which is defined by (9.32) and in this case is

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & -1 \\ 0 & 0 & 1 \end{pmatrix}.$$

We can easily update W^{-1} by premultiplying it by T^{-1} which is the same as premultiplying it by T . The new W^{-1} matrix is

$$TW^{-1} = \begin{pmatrix} 1/2 & 1/2 & 1/2 \\ -1/4 & 1/4 & -3/4 \\ 0 & 0 & 1 \end{pmatrix}.$$

From (9.23) we compute \bar{h} (with the new $r = 2$ and $j_r = 2$) by

$$\bar{h} = TW^{-1}(A_{\bullet 10} - Gd^{10}) = (-1, 5, 0)^T.$$

Then from equation (9.24) we get

$$\bar{W}^{-1} = \begin{pmatrix} 1 & 1/5 & 0 \\ 0 & 1/5 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1/2 & 1/2 & 1/2 \\ -1/4 & 1/4 & -3/4 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 9/20 & 11/20 & 7/20 \\ -1/20 & 1/20 & -3/20 \\ 0 & 0 & 1 \end{pmatrix}.$$

The new G matrix is

$$G = [A_{\bullet 3}, A_{\bullet 5}, A_{\bullet 6}, A_{\bullet 8}, A_{\bullet 9}].$$

The new basic feasible solution is

$$x_B = (6, 0, 0; 1, 1, 1, 1, 1)^T \quad \text{where } \mathcal{B} = \{1, 10, 4; 3, 5, 6, 8, 9\}.$$

The new prices are

$$\pi = \left(-\frac{9}{20}, -\frac{11}{20}, -\frac{7}{20}, \frac{7}{20}, \frac{49}{20}, \frac{47}{20}, \frac{31}{20}, -\frac{42}{20} \right)^T.$$

After pricing out, we find that the reduced costs corresponding to the nonbasic variables x_2 and x_7 are, respectively, $(4/5, 13/2) \geq 0$. Hence the above solution x_B is optimal.

9.5 NOTES & SELECTED BIBLIOGRAPHY

The generalized upper bounding (GUB) technique was developed by Dantzig & Van Slyke [1967]. Example 9.1 is the same as the one discussed in their paper. A generalization of this method, which replaces each of the GUB constraints by a rectangular block of constraints involving the same subset of variables, has been carried out by Kaul [1965]. Similar approaches have been investigated by Bennet [1963], Charnes & Lemke [1960], Rosen [1964], Sakarovitch & Saigal [1967], and others. Brown & Thomen [1980] developed an algorithm for automatically identifying generalized upper bounds in a linear program.

In all our discussions we have assumed that the bounds or generalized bounds are fixed. Several interesting papers have appeared that handle variable upper bounds and generalized variable upper bounds (GVUB). See, for example, Bastian [1984], Schrage [1975], and Todd [1982, 1983].

9.6 PROBLEMS

- 9.1 Consider a linear program with GUB constraints having some negative coefficients -1 instead of all $+1$. How would the development of the solution procedure and the updates to the working basis change?
- 9.2 Extend the GUB technique where the nonnegativity constraints on (9.2) are replaced by upper and lower bounds on x , i.e., $l \leq x \leq u$.

DECOMPOSITION OF LARGE-SCALE SYSTEMS

Large-scale linear programming systems typically have special structures that can be exploited to gain computational advantage, for example, those having a very large percentage of zero coefficients. It is routine to take advantage of this *sparsity* of nonzero coefficients. In this Chapter we discuss how specialized versions of the Simplex Method can be used to solve systems that have a special *block-matrix* structure. We begin with the simple case of a system that, except for the objective function, consists of two independent subsystems that have no variables in common, for example:

$$\begin{aligned}
 \text{Minimize} \quad & \sum_{j=1}^{n_1} c_j x_j + \sum_{j=n_1+1}^n c_j x_j = z \\
 \text{subject to} \quad & \sum_{j=1}^{n_1} A_{ij} x_j = b_i, \quad i = 1, \dots, m_1 \\
 & \sum_{j=n_1+1}^n A_{ij} x_j = b_i, \quad i = m_1 + 1, \dots, m \\
 & x_j \geq 0, \quad j = 1, \dots, n.
 \end{aligned} \tag{10.1}$$

Since there is no connection between the blocks except for the objective function, it is obvious that the solution to the linear program (10.1) can be found by solving the two linear programs (one for each block) separately and adding the objectives to obtain z .

▷ **Exercise 10.1** Show that if there are K subsystems, each of dimension K , that are independent of each other, except for the objective function, then it would take approximately $1/K^2$ times the effort to solve the full system, assuming, in general it takes less than γm^3 arithmetic operations to solve a linear program in m equations for some constant γ .

The *block-angular* system (10.2) is a generalization of (10.1); it has K independent blocks and one set of coupling constraints:

$$\begin{array}{ll}
 \text{Minimize} & (c^o)^T x^o + (c^1)^T x^1 + \cdots + (c^K)^T x^K = z \\
 \text{subject to} & A^0 x^o + \begin{array}{l} A^1 x^1 + \cdots + A^K x^K = b \\ F^1 x^1 \\ \vdots \\ F^K x^K = f^K \end{array} \\
 & x^o \geq 0, x^1 \geq 0, \dots, x^K \geq 0.
 \end{array} \tag{10.2}$$

A possible application of a block angular system might be for a company with K almost independent factories $k = 1, \dots, K$. Each factory has many constraints that are independent of the constraints of the other factories. There are a few constraints, however, such as shared budget, skilled labor, and a profit function that the K factories must share. In (10.2), x^k is the vector of activity levels of the k th factory and x^o is the set of activity levels of the headquarters that are not a part of the activities of any factory. The first equation is the objective function, the second line the m constraints expressing the sharing of m scarce resources across the board, the third line the m_1 constraints that involve the first factory only, and the last line the m_K constraints that involve the K th factory only.

The structure of the block-angular system (10.2) suggests that we try to break the problem into K independent parts and then adjust the solution to take into account the interconnections. One way to do this, popular with economists, is to begin by arbitrarily assigning prices to the scarce resources and let each factory optimize its activities assuming that it has to pay for scarce resources according to these prices. The scarce resources that the headquarters and each factory will demand in general will be *out of kilter* with b , the resources available to the system, and the problem becomes one of finding *an efficient algorithm* to adjust the prices (Lagrange multipliers). In this chapter we show how to do this in a finite number of iterations using the *Dantzig-Wolfe (D-W) Decomposition Principle*.

This algorithm was first proposed in technical papers of the RAND Corporation around 1958 and first published in technical journals in 1960. Because of its potential application to decentralized planning it was enthusiastically received by economists. However, contrary to expectations, in initial trials on certain classes of practical problems, the algorithm turned out to be disappointingly slow. Later, when the initial implementations were replaced by codes prepared by skilled mathematical programmers, it turned out to be very efficient. Unfortunately, the initial incorrect reports that the algorithm is inefficient still persist in the literature.

Another class of problems encountered in practice, amenable to decomposition methods, are the *staircase* systems, which differ from block-angular systems (10.2)

in that the activities associated with any step of the staircase share input/output resources with those on the step below it on the staircase. For example, (10.3) depicts a staircase system with four steps:

$$\begin{array}{rcll}
 \text{Minimize} & (c^1)^T x^1 + (c^2)^T x^2 + (c^3)^T x^3 + (c^4)^T x^4 & = & z \\
 \text{subject to} & A^{11} x^1 & = & b^1 \\
 & A^{21} x^1 + A^{22} x^2 & = & b^2 \\
 & A^{32} x^2 + A^{33} x^3 & = & b^3 \\
 & A^{43} x^3 + A^{44} x^4 & = & b^4 \\
 & x^k \geq 0, & k = 1, \dots, 4. &
 \end{array} \tag{10.3}$$

Staircase systems often arise in the study of processes through time in which the activities of one period (or stage) directly affect or are affected by the preceding and following periods (or stages) but with no others. Such systems arise in manufacturing where the production at one stage of the process is affected by that of the previous stage and affects the products of the following stage only. In such problems, it is often the case that several of the submatrices A^{ii} along the diagonal are all the same and several of the $A^{i,i-1}$ along the subdiagonal are also all the same; when true it is possible to take advantage of it.

Another more general type of system that can be solved using decomposition methods are the *lower block-triangular* systems such as the four-stage one displayed in (10.4) below:

$$\begin{array}{rcll}
 \text{Minimize} & (c^1)^T x^1 + (c^2)^T x^2 + (c^3)^T x^3 + (c^4)^T x^4 & = & z \\
 \text{subject to} & A^{11} x^1 & = & b^1 \\
 & A^{21} x^1 + A^{22} x^2 & = & b^2 \\
 & A^{31} x^1 + A^{32} x^2 + A^{33} x^3 & = & b^3 \\
 & A^{41} x^1 + A^{42} x^2 + A^{43} x^3 + A^{44} x^4 & = & b^4 \\
 & x^1 \geq 0, x^2 \geq 0, x^3 \geq 0, x^4 \geq 0, & &
 \end{array} \tag{10.4}$$

where stage 1 directly affects stages 2, 3, and 4; stage 2 directly affects stages 3 and 4; and stage 3 directly affects stage 4.

10.1 WOLFE'S GENERALIZED LINEAR PROGRAM

When a production system is being modeled, it may happen that the input and output coefficients of one or more activities are not in fixed proportions (as is the case for linear programs), but each column of coefficients may be freely chosen as any point from a convex set \mathcal{C}_j . This important class of problems is called a "Generalized Linear Program." These were first studied by Philip Wolfe.

Definition: A *generalized linear program* is a problem stated in standard linear programming problem format:

$$\begin{aligned} &\text{Minimize} && c^T x = z \\ &\text{subject to} && Ax = b, \quad A : m \times n, \\ &&& x \geq 0, \end{aligned} \tag{10.5}$$

where each column $\begin{pmatrix} c_j \\ A_{\bullet j} \end{pmatrix}$ may be freely chosen to be any point in a given convex set $\mathcal{C}_j, j = 1, \dots, n$.

By simple extension, the fixed right-hand-side vector b may also be replaced by a vector picked from a convex set \mathcal{C}_b .

THEOREM 10.1 (Equivalent Generalized Linear Program) *The generalized linear program (10.6) is equivalent to the generalized program (10.7) generated at some iteration of Wolfe’s algorithm.*

Original Generalized Linear Program:

$$\begin{aligned} &\text{Minimize} && \sum_{j=1}^n c_j \hat{x}_j = \hat{z} \\ &\text{subject to} && \sum_{j=1}^n A_{ij} \hat{x}_j = b_i, \quad \text{for } i = 1, \dots, m \\ &&& \hat{x}_j \geq 0, \quad \text{for } j = 1, \dots, n, \end{aligned} \tag{10.6}$$

where $\begin{pmatrix} c_j \\ A_{\bullet j} \end{pmatrix} \in \mathcal{C}_j$ are freely chosen vectors in convex sets \mathcal{C}_j .

Equivalent Generalized Linear Program:

$$\begin{aligned} &\text{Minimize} && \sum_{j=1}^n \left(c_j x_j + \sum_{t=1}^{T_j} c_j^t x_j^t \right) = z \\ &\text{subject to} && \sum_{j=1}^n \left(A_{ij} x_j + \sum_{t=1}^{T_j} A_{ij}^t x_j^t \right) = b_i, \quad \text{for } i = 1, \dots, m \\ &&& x_j \geq 0, \quad \text{for } j = 1, \dots, n, \end{aligned} \tag{10.7}$$

where $\begin{pmatrix} c_j \\ A_{\bullet j} \end{pmatrix} \in \mathcal{C}_j$ are freely chosen vectors in convex sets \mathcal{C}_j and $\begin{pmatrix} c_j^t \\ A_{\bullet j}^t \end{pmatrix} \in \mathcal{C}_j, t = 1, \dots, T_j$ are T_j fixed points in \mathcal{C}_j .

Proof. Let

$$\hat{x}_j = u_j, \quad \begin{pmatrix} c_j \\ A_{\bullet j} \end{pmatrix} = \begin{pmatrix} c_j^o \\ A_{\bullet j}^o \end{pmatrix}, \quad \text{Min } \hat{z}$$

be an optimal solution to (10.6). Let

$$x_j = v_j, \quad x_j^t = v_j^t, \quad \begin{pmatrix} c_j \\ A_{\bullet j} \end{pmatrix} = \begin{pmatrix} c_j^t \\ A_{\bullet j}^t \end{pmatrix}, \quad \text{Min } z$$

be an optimal solution to (10.7). Note $(x_j = u_j, x_j^t = 0, \text{Min } \hat{z})$ is a feasible solution to (10.7). Therefore $\text{Min } z \leq \text{Min } \hat{z}$. Note that the optimal solution to (10.7) can be rewritten as a feasible solution to (10.6):

$$\begin{aligned} \text{Minimize} \quad & \sum_{j=1}^n \bar{c}_j \bar{v}_j = \bar{z} \\ \text{subject to} \quad & \sum_{j=1}^n \bar{A}_{ij} \bar{v}_j = b_i, \quad \text{for } i = 1, \dots, m \\ & \bar{v}_j \geq 0, \quad \text{for } j = 1, \dots, n, \end{aligned} \tag{10.8}$$

where

$$\begin{aligned} \bar{v}_j &= v_j + \sum_{t=1}^{T_j} v_j^t \\ \begin{pmatrix} \bar{c}_j \\ \bar{A}_{\bullet j} \end{pmatrix} &= \begin{cases} \begin{pmatrix} c_j^* \\ A_{\bullet j}^* \end{pmatrix} \frac{v_j}{\bar{v}_j} + \begin{pmatrix} c_j^t \\ A_{\bullet j}^t \end{pmatrix} \frac{v_j^t}{\bar{v}_j} & \text{if } \bar{v}_j \neq 0 \\ \begin{pmatrix} c_j^* \\ A_{\bullet j}^* \end{pmatrix} & \text{if } \bar{v}_j = 0 \end{cases} \end{aligned}$$

Therefore $\text{Min } z \geq \text{Min } \hat{z}$. This and $\text{Min } z \leq \text{Min } \hat{z}$ imply $\text{Min } z = \text{Min } \hat{z}$. ■

In the following discussion, we further assume that the convex sets are defined by systems of linear inequalities; however, the method can be easily extended to the situation where the convex sets are general.

Example 10.1 (Generalized Linear Program) An example of a generalized linear program is to find $x_j \geq 0, y_{i4}, \text{Min } z$ such that

$$\begin{aligned} 6x_1 + 4x_2 + x_3 + y_{04}x_4 &= z \quad (\text{min}) \\ x_1 + x_2 - 4x_3 + y_{14}x_4 &= 5 \\ -x_1 + x_2 - x_3 + y_{24}x_4 &= 1 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, \end{aligned} \tag{10.9}$$

where the coefficients $y_{\bullet 4} = (y_{04}, y_{14}, y_{24})$, are not fixed but must be chosen to be a point in the convex set:

$$C_4 = \{ y_{\bullet 4} \mid 3y_{04} + y_{14} + 2y_{24} = 2 \text{ with } y_{i4} \geq 0 \text{ for } i = 0, 1, 2 \}. \tag{10.10}$$

This is an example of a nonlinear system. However, we will show that by a suitable change of variables, the problem can be reformulated as a linear program. Multiply $3y_{04} + y_{14} + 2y_{24} = 2$ by x_4 to obtain

$$3y_{04}x_4 + y_{14}x_4 + 2y_{24}x_4 = 2x_4. \tag{10.11}$$

Substitute variables $u_0 = y_{04}x_4$, $u_1 = y_{14}x_4$, $u_2 = y_{24}x_4$ into (10.9) and (10.11) to obtain the linear program:

$$\begin{aligned}
 6x_1 + 4x_2 + x_3 + u_0 &= \bar{z} \text{ (min)} \\
 x_1 + x_2 - 4x_3 + u_1 &= 5 \\
 -x_1 + x_2 - x_3 + u_2 &= 1 \\
 3u_0 + u_1 + 2u_2 &= 2x_4 \\
 x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, u_0 \geq 0, u_1 \geq 0, u_2 \geq 0.
 \end{aligned} \tag{10.12}$$

Our immediate goal is to prove that Wolfe’s nonlinear program (10.10) and the linear program (10.12) are equivalent in the sense that if $\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4, \hat{u}_0, \hat{u}_1, \hat{u}_2, \hat{z}$ is optimal for (10.12) and $\hat{x}_4 \neq 0$, then $x_1 = \hat{x}_1, x_2 = \hat{x}_2, x_3 = \hat{x}_3, x_4 = \hat{x}_4, y_{04} = \hat{u}_0/\hat{x}_4, y_{14} = \hat{u}_1/\hat{x}_4, y_{24} = \hat{u}_2/\hat{x}_4, z = \hat{z}$ is optimal for the generalized program (10.9).

Proof. Assume an optimal solution to Wolfe’s nonlinear program (10.9) is $z^*, x_1^*, x_2^*, x_3^*, x_4^*, y_{04}^*, y_{14}^*, y_{24}^*$, then a feasible solution to the linear program (10.12) is $x_1^*, x_2^*, x_3^*, x_4^*, u_0^* = y_{04}^*x_4^*, u_1^* = y_{14}^*x_4^*, u_2^* = y_{24}^*x_4^*, \bar{z} = z^*$.

Conversely assume the minimal solution to the linear program (10.12) is $\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4, \hat{u}_0, \hat{u}_1, \hat{u}_2, \bar{z} = \hat{z}$. Then a feasible solution to Wolfe’s generalized LP is $x_1 = \hat{x}_1, x_2 = \hat{x}_2, x_3 = \hat{x}_3, x_4 = \hat{x}_4, y_{04} = \hat{u}_0/\hat{x}_4, y_{14} = \hat{u}_1/\hat{x}_4, y_{24} = \hat{u}_2/\hat{x}_4$ under the assumption of $\hat{x}_4 \neq 0$. We conclude that (10.9) and (10.12) have the same optimal solution. ■

▷ **Exercise 10.2** If it turns out that in optimizing (10.12) $x_4 = \hat{x}_4 = 0$, then show by (10.13) that the optimal solution to the generalized program is not necessarily found by setting $x_4 = 0$ in (10.9) and optimizing (10.9). Find $x_j \geq 0, y_{i4}$, Min z such that

$$\begin{aligned}
 6x_1 + 4x_2 + x_3 + y_{04}x_4 &= z \text{ (min)} \\
 x_1 + x_2 - 4x_3 + y_{14}x_4 &= 5 \\
 -x_1 + x_2 - x_3 + y_{24}x_4 &= 1 \\
 x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0,
 \end{aligned} \tag{10.13}$$

where the coefficients $y_{\bullet 4} = (y_{04}, y_{14}, y_{24})$, are not fixed but must be chosen to be a point in the convex set:

$$C_4 = \{ y_{\bullet 4} \mid 3y_{04} - y_{14} + 2y_{24} = 2 \text{ with } y_{i4} \geq 0 \text{ for } i = 0, 1, 2 \}. \tag{10.14}$$

Show that the associated linear programming solution cannot be used to solve the original system.

▷ **Exercise 10.3** In Example 10.1 multiply (10.10) by x_4 , replace each $y_{i4}x_4$ by u_i , reduce the system to a linear program, and solve by hand or by using the DTZG Simplex Primal (*Linear Programming 1*) software option. Can this solution be used to solve the original system? Show that in this case the linear program is equivalent to the original generalized linear program in a certain sense.

THEOREM 10.2 (Generalized LP and Equivalent LP) *The generalized linear program*

$$\begin{aligned}
 \text{Minimize} \quad & c^T x + y_{0,n+1}x_{n+1} = z \\
 \text{subject to} \quad & Ax + y_{\bullet,n+1}x_{n+1} = b, \quad A : m \times n, \\
 & (x, x_{n+1}) = (x_1, x_2, \dots, x_{n+1}) \geq 0,
 \end{aligned} \tag{10.15}$$

where the coefficients $(y_{0,n+1}, y_{\bullet,n+1})$ are not fixed but must be chosen to be a point in the convex set:

$$C_{n+1} = \left\{ y_{i,n+1} \mid \sum_{i=0}^m \alpha_i y_{i,n+1} = 1, y_{i,n+1} \geq 0 \text{ for } i = 0, \dots, m \right\}. \quad (10.16)$$

has the same optimal solution as the linear program

$$\begin{aligned} \text{Minimize} \quad & c^T x + u_0 & & = w \\ \text{subject to} \quad & Ax & + & u = b, & & A : m \times n, \\ & & & \sum_{i=0}^m \alpha_i u_{i,n+1} = x_{n+1} & & (10.17) \\ & & & x_1, x_2, \dots, x_{n+1} \geq 0, u_0, u_1, \dots, u_m \geq 0 \end{aligned}$$

where $u_i = y_{i,n+1} x_{n+1}$ under the assumption that the optimal solution to (10.15) has $x_{n+1} = x_{n+1}^* > 0$.

Proof. We will prove that the optimal solutions are the same for (10.15) and (10.17) by showing that the optimal solution for (10.15) is a feasible solution to (10.17) and vice versa.

Let $z = z^*$, $x_j = x_j^*$, for $j = 1, \dots, n + 1$ with $x_{n+1}^* > 0$, and $y_{i,n+1} = y_{i,n+1}^*$, for $i = 0, \dots, m$ be an optimal feasible solution to (10.15). The values $x_j = x_j^*$ for $j = 1, \dots, n + 1$ and $u_i = y_{i,n+1}^* x_{n+1}^*$ $i = 0, \dots, m + 1$, are clearly feasible for (10.17); i.e.,

$$\begin{aligned} \text{Minimize} \quad & c^T x^* + y_{0,n+1}^* x_{n+1}^* & & = w^* \\ \text{subject to} \quad & Ax^* & + & y_{\bullet,n+1}^* x_{n+1}^* = b, & & A : m \times n \\ & & & \sum_{i=0}^m \alpha_i y_{i,n+1}^* x_{n+1}^* = x_{n+1}^* & & (10.18) \end{aligned}$$

Therefore $z^* \leq w^*$.

Next suppose that w^* is not optimal but there exists another set of values $w = \bar{w}$, $x_j = \bar{x}_j$ for $j = 1, \dots, n + 1$ and $u_i = \bar{u}_i$, $i = 0, \dots, m + 1$, that result in an optimal feasible solution for (10.17). Since $\bar{x}_{n+1} > 0$ by our assumption, we compute $\bar{y}_{i,n+1} = \bar{u}_{i,n+1} / \bar{x}_{n+1}$ for $i = 0, \dots, m$. The solution $x_j = \bar{x}_j$ for $j = 1, \dots, n + 1$ with $y_{i,n+1} = \bar{y}_{i,n+1}$ for $i = 0, \dots, m$ is clearly feasible for (10.15) and therefore $\bar{w} \leq \bar{z}$. Hence the optimal solutions for the two systems are the same. ■

▷ **Exercise 10.4** Suppose that the optimal solution to (10.15) has $x_{n+1} = x_{n+1}^* = 0$. Show that systems (10.15) and (10.17) are not equivalent in this case.

Returning to the general case (10.5), for now assume that

$$\begin{pmatrix} c_j \\ A_{\bullet j} \end{pmatrix} = y_{\bullet j} = (y_{0j}, y_{1j}, \dots, y_{mj})^T$$

may be freely chosen to be any point in a convex set \mathcal{C}_j defined by a system of linear inequalities in the variable coefficient parameters y_{ij} for $i = 0, 1, \dots, m$, and possibly auxiliary variables $y_{m+1,j}, y_{m+2,j}, \dots, y_{m+k,j}$ independent of the rest of the system. Problem (10.5) can be restated as a linear program in x_j and u_{ij} by multiplying the relations of \mathcal{C}_j by $x_j \geq 0$ and substituting $u_{ij} = y_{ij}x_j$, unrestricted in sign, as new variables. If this substitution generates a linear program that yields a solution in which $x_j \neq 0$ then it is easy to back-substitute and get an optimal feasible solution to the original system.

▷ **Exercise 10.5** Suppose after the substitution, that the resulting linear program in u_{ij} and x_j generates a solution with the property that whenever $x_j = 0$ that $u_{ij} = 0$ for $i = 1, \dots, m$. Show how this property can be used to find a solution to the original system (10.5).

▷ **Exercise 10.6** Suppose that a solution is obtained for the new linear system in u_{ij} and x_j with $x_t = 0$; show that a solution $y_{it}x_t \neq 0$ is still possible. Construct an example to show how this can happen. Show that this case can only happen if \mathcal{C}_j is an unbounded convex set. Furthermore show that we can get a sequence of solutions to the original problem in which $y_{it} \not\rightarrow 0$ and $x_t \rightarrow 0$.

The reduction of a generalized program to a linear program is not recommended as a solution technique. Instead we recommend solving it by a series of adjustments of the values of y_{ij} obtained by sequentially solving certain auxiliary programs or *subprograms* in y_{ij} . In effect, a large linear program with variable coefficients is *decomposed* into smaller linear programs, each of which can be solved very efficiently. We shall now illustrate the method.

Example 10.2 (Illustration of Wolfe's Method) Consider the problem defined by equations (10.9) and (10.10) and restated here for convenience.

$$\begin{aligned} 6x_1 + 4x_2 + x_3 + y_{04}x_4 &= z \text{ (min)} \\ x_1 + x_2 - 4x_3 + y_{14}x_4 &= 5 \\ -x_1 + x_2 - x_3 + y_{24}x_4 &= 1 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, \end{aligned} \tag{10.19}$$

where the coefficients $y_{\bullet 4} = (y_{04}, y_{14}, y_{24})$, are not fixed but must be chosen to be a point in the convex set:

$$\mathcal{C}_4 = \{ y_{\bullet 4} \mid 3y_{04} + y_{14} + 2y_{24} = 2 \text{ with } y_{i4} \geq 0 \text{ for } i = 0, 1, 2 \}. \tag{10.20}$$

If we pick a starting basic set of variables $(-z, x_1, x_2)$, the basic feasible solution corresponding to this set is

$$(-z) = 24, x_1 = 2, x_2 = 3, x_3 = x_4 = 0. \tag{10.21}$$

To test if this basic feasible solution is optimal, we compute the simplex multipliers by solving $B^T\pi = c_B$ to obtain

$$\pi = (5, -1)^T.$$

We use π to compute the reduced costs \bar{c}_3 and \bar{c}_4 , where $\bar{c}_j = c_j - \pi^T A_{\bullet j}$:

$$\begin{aligned}\bar{c}_3 &= 20 \\ \bar{c}_4 &= y_{04} - 5y_{14} + y_{24}.\end{aligned}$$

where $(y_{04}, y_{14}, y_{24}) \in \mathcal{C}_4$; see (10.10).

Note this basic solution (10.21) is minimal, if $\bar{c}_j \geq 0$ for all j . In this case $\bar{c}_3 > 0$ and \bar{c}_4 may be less than zero depending on the choice of the variable coefficient parameters y_{04} , y_{14} , and y_{24} . In order to determine if \bar{c}_4 can be < 0 , we minimize the value for \bar{c}_4 subject to conditions (10.10); i.e., we solve the subprogram associated with subset \mathcal{C}_4 :

$$\begin{aligned}\text{Minimize} \quad & y_{04} - 5y_{14} + y_{24} = \bar{c}_4 \\ \text{subject to} \quad & 3y_{04} + y_{14} + 2y_{24} = 2, \\ & y_{04} \geq 0, y_{14} \geq 0, y_{24} \geq 0.\end{aligned}\tag{10.22}$$

Observe in this case that the minimal solution is $y_{04} = 0$, $y_{14} = 2$, $y_{24} = 0$, and $\bar{c}_4 = -10$. Therefore the basic solution (10.21) fails to pass the test for optimality for the original problem.

We next obtain an improved solution to (10.19) by allowing x_4 , with column coefficients $y_{04}^{(1)} = 0$, $y_{14}^{(1)} = 2$, $y_{24}^{(1)} = 0$, to become an incoming basic variable. Once we introduce x_4 into the basic set of (10.19), x_1 drops out, resulting in the new basic feasible solution:

$$(-z) = -4, \quad x_2 = 1, \quad x_4 = 2, \quad x_1 = x_3 = 0.$$

However, we need to provide for the possibility of revising the values of y_{i4} to obtain a still lower value of z . This is done by rewriting the program (10.9) in the equivalent form:

$$\begin{aligned}6x_1 + 4x_2 + x_3 + 0x_4^{(1)} + y_{04}x_4 &= z \text{ (min)} \\ x_1 + x_2 - 4x_3 + 2x_4^{(1)} + y_{14}x_4 &= 5 \\ -x_1 + x_2 - x_3 + 0x_4^{(1)} + y_{24}x_4 &= 1 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4^{(1)} \geq 0, x_4 &\geq 0,\end{aligned}\tag{10.23}$$

where y_{i4} satisfy the same relations (10.10) as before, i.e.,

$$3y_{04} + y_{14} + 2y_{24} = 2 \quad \text{and} \quad y_{i4} \geq 0 \text{ for } i = 0, 1, 2.\tag{10.24}$$

The column $(y_{04}, y_{14}, y_{24})^T$ will be referred to as the *generic column*. It may seem that we have changed our original problem; however, we know by Theorem 10.1 that the new problem (10.23) is equivalent to the original one (10.19).

The new basic feasible solution to (10.23) is

$$(-z) = -4, \quad x_2 = 1, \quad x_4^{(1)} = 2, \quad x_1 = x_3 = x_4 = 0.\tag{10.25}$$

The new simplex multipliers obtained from $B^T\pi = c_B$ are:

$$\pi = (0, 4)^T.$$

We next obtain the reduced costs from $\bar{c}_N = c_N - N^T\pi$ as

$$\bar{c}_1 = 10, \quad \bar{c}_3 = 5, \quad \bar{c}_4 = y_{04} - 4y_{24}.$$

The reduced costs \bar{c}_1 and \bar{c}_3 are nonnegative; in order to determine if $\bar{c}_4 < 0$ is possible, we solve the subprogram

$$\begin{aligned} \text{Minimize} \quad & y_{04} && - 4y_{24} = \bar{c}_4 \\ \text{subject to} \quad & 3y_{04} + y_{14} + 2y_{24} = 2, \\ & y_{04} \geq 0, y_{14} \geq 0, y_{24} \geq 0, \end{aligned} \tag{10.26}$$

which is the same problem as (10.22) except it has a different objective expression for \bar{c}_4 . On solving (10.26) we obtain $\bar{c}_4 = -4$, $y_{04} = 0$, $y_{14} = 0$, and $y_{24} = 1$. Thus, the solution (10.25) fails again to pass the test for optimality of the equivalent problem. The variable x_4 is the only nonbasic with a negative cost; bringing it into the basis causes x_2 to drop out and the new basic feasible solution is

$$(-z) = -4, x_4^{(1)} = 5/2, x_4 = 1, x_1 = x_2 = x_3 = 0.$$

However, again we need to be careful. We construct a new augmented equivalent program (10.27), which allows for the possibility of y_{i4} to be revised:

$$\begin{aligned} 6x_1 + 4x_2 + x_3 + 0x_4^{(1)} + 0x_4^{(2)} + y_{04}x_4 &= z \text{ (min)} \\ x_1 + x_2 - 4x_3 + 2x_4^{(1)} + 0x_4^{(2)} + y_{14}x_4 &= 5 \\ -x_1 + x_2 - x_3 + 0x_4^{(1)} + 1x_4^{(2)} + y_{24}x_4 &= 1 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4^{(1)} \geq 0, x_4^{(2)} \geq 0, x_4 \geq 0, \end{aligned} \tag{10.27}$$

The new basic feasible solution is

$$(-z) = 0, x_4^{(1)} = 5/2, x_4^{(2)} = 1, x_1 = x_2 = x_3 = x_4 = 0. \tag{10.28}$$

The new simplex multipliers obtained from $B^T\pi = c_B$ are:

$$\pi = (0, 0)^T.$$

We once again obtain the reduced costs from $\bar{c} = c_N - N^T\pi$;

$$\bar{c}_1 = 6, \bar{c}_2 = 4, \bar{c}_3 = 1, \bar{c}_4 = y_{04}.$$

As before, in order to see if $\bar{c}_4 < 0$ is possible, we solve the subprogram with the revised objective:

$$\begin{aligned} \text{Minimize} \quad & y_{04} && = \bar{c}_4 \\ \text{subject to} \quad & 3y_{04} + y_{14} + 2y_{24} = 2 \\ & y_{04} \geq 0, y_{14} \geq 0, y_{24} \geq 0. \end{aligned} \tag{10.29}$$

On solving (10.29) we obtain $\bar{c}_4 = 0$, $y_{04} = 0$, $y_{14} = 2$, and $y_{24} = 0$. Since $\bar{c}_4 = 0$, the solution (10.28) is optimal because all the reduced costs are nonnegative for all feasible values of $y_{\bullet 4}$. The optimal solution to (10.9) and (10.10) can be derived from the solution to (10.27) and (10.29) as follows: $z = 0$, $x_1 = x_2 = x_3 = 0$, $x_4 = x_4^{(1)} + x_4^{(2)} = 5/2 + 1 = 7/2$,

$$\begin{aligned} \begin{pmatrix} y_{04} \\ y_{14} \\ y_{24} \end{pmatrix} &= y_{\bullet 4}^{(1)} \frac{x_4^{(1)}}{x_4^{(1)} + x_4^{(2)}} + y_{\bullet 4}^{(2)} \frac{x_4^{(2)}}{x_4^{(1)} + x_4^{(2)}} \\ &= \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} \frac{5}{7} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \frac{2}{7} = \begin{pmatrix} 0 \\ 10/7 \\ 2/7 \end{pmatrix}. \end{aligned} \tag{10.30}$$

What to Do If the Subprogram has an Unbounded Solution

In general we must allow for the possibility for some incoming column $j = s$ such that $y_{\bullet s}$ is a convex combination of one or more extreme points plus a nonnegative combination of one or more homogenous solutions (see Theorem 10.5). Thus, in general, when we solve the subprogram, two cases can arise:

1. If an extreme-point solution $y_{\bullet s} = y_{\bullet s}^e \in C_s$ is obtained to the subprogram that prices out negative, bring it into the basis of the original problem.
2. If a class of solutions $y_{\bullet s} = y_{\bullet s}^e + \theta y_{\bullet s}^h \in C_s$, where $y_{\bullet s}^e$ is an extreme-point solution and $y_{\bullet s}^h$ is a homogeneous solution with $\theta \geq 0$ a scalar parameter, is obtained, then introduce only the homogeneous part $y_{\bullet s}^h$ into the basis of the original problem.

The intuitive reason for this is as follows. It is clear that increasing θ makes $y_{\bullet s}$ price out more negative and hence more attractive to introduce $y_{\bullet s}$ as a column into the basis of the original problem. If we rewrite $y_{\bullet s}x_s$ as

$$y_{\bullet s}x_s = (y_{\bullet s}^e + \theta y_{\bullet s}^h)x_s = \left(\frac{1}{\theta}y_{\bullet s}^e + y_{\bullet s}^h\right)\theta x_s \tag{10.31}$$

it is clear that that the nonhomogeneous part becomes negligible relative to the homogeneous part as θ increases. Example 10.3 illustrates this.

▷ **Exercise 10.7** Suppose the “optimal” solution to the generalized program example using Wolfe’s procedure has some positive weights x_s^t on $y_{\bullet s}^h$ associated with homogeneous solutions to the subproblem C_s and has zero weights x_s^r on each extreme-point solution $y_{\bullet s}^e$ associated with the subproblem C_s . Let $z = z^*$. Prove, if the final basis is nondegenerate, there exists a class of feasible solutions with positive weights on all extreme solutions and extreme homogeneous solutions such that $z \rightarrow z^*$. Show, if the basis is degenerate, that none of the class of solutions need to be feasible but they tend to feasibility in the limit.

Example 10.3 (Subprogram Has an Unbounded Solution) We illustrate the situation where the subprogram has an unbounded solution. Consider the following example of a generalized linear program:

$$\begin{aligned} 0x_1 + 0x_2 + 3x_3 + y_{04}x_4 &= z \text{ (min)} \\ x_1 &+ y_{14}x_4 = 1 \\ &x_2 + y_{24}x_4 = 1 \\ &x_3 + y_{34}x_4 = 1 \\ x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0 \end{aligned} \tag{10.32}$$

where y_{04} , y_{14} , y_{24} , and y_{34} may be chosen to have any values satisfying

$$\begin{aligned} y_{04} &= 2 \\ -0.5y_{14} + y_{24} &= 0 \\ -0.5y_{14} + y_{34} &= 1 \\ y_{04} \geq 0, y_{14} \geq 0, y_{24} \geq 0, y_{34} \geq 0. \end{aligned} \tag{10.33}$$

If we pick a starting basic set of variables to be $(-z, x_1, x_2, x_3)$, the basic feasible solution is

$$(-z) = -3, \quad x_1 = 3, \quad x_2 = 1, \quad x_3 = 1, \quad x_4 = 0. \tag{10.34}$$

In order to determine if this basic feasible solution is optimal, we first compute the simplex multipliers by solving $B^T\pi = c_B$ to obtain

$$\pi = (0, 0, 3)^T.$$

We next obtain the reduced costs \bar{c}_4 from $\bar{c}_N = c_N - N^T\pi$:

$$\bar{c}_4 = y_{04} - 3y_{34} \quad \text{with } y_{04} = 2. \tag{10.35}$$

The test for a minimum is $\bar{c}_j \geq 0$ for all j . In this case, only \bar{c}_4 , as defined by (10.35), may be less than zero depending on the values of the parameters y_{04} and y_{34} . In order to determine if $\bar{c}_4 < 0$ is possible, we minimize the value for \bar{c}_4 subject to (10.33); i.e., we solve the subprogram:

$$\begin{aligned} \text{Minimize} \quad & y_{04} && - 3y_{34} = \bar{c}_4 \\ \text{subject to} \quad & - (1/2)y_{14} + y_{24} && = 0 \\ & - (1/2)y_{14} && + y_{34} = 1 \\ & y_{04} = 2, \quad y_{14} \geq 0, \quad y_{24} \geq 0, \quad y_{34} \geq 0. \end{aligned} \tag{10.36}$$

On applying the Simplex Method, we find the class of solutions

$$y_{\bullet 4} = y_{\bullet 4}^e + \theta y_{\bullet 4}^h = \begin{pmatrix} 2 \\ 0 \\ 0 \\ 1 \end{pmatrix} + \theta \begin{pmatrix} 0 \\ 1 \\ 1/2 \\ 1/2 \end{pmatrix} \tag{10.37}$$

and $\bar{c}_4 = -1 - (3/2)\theta \rightarrow -\infty$ as $\theta \rightarrow \infty$. Since $\bar{c}_4 < 0$ for all $\theta \geq 0$, the basic solution (10.34) does not satisfy the test for optimality for the original problem (10.32) whatever be $\theta \geq 0$.

We obtain an improved solution by introducing the homogeneous part to (10.37) as a new column $y_{\bullet 4}^h = (y_{04}^h = 0, y_{14}^h = 1, y_{24}^h = 1/2, y_{34}^h = 1/2)^T$ into the basis of (10.32):

$$\begin{aligned} 0x_1 + 0x_2 + 3x_3 + \quad & 0x_4^h = z \quad (\text{min}) \\ x_1 & + \quad 1x_4^h = 1 \\ & x_2 + (1/2)x_4^h = 1 \\ & x_3 + (1/2)x_4^h = 1 \\ x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4 \geq 0. \end{aligned} \tag{10.38}$$

It is straightforward to see that the variable x_1 leaves the basis and the new basic feasible solution is:

$$(-z) = -3/2, \quad x_2 = 1/2, \quad x_3 = 1/2, \quad x_4^h = 1, \quad x_1 = 0. \tag{10.39}$$

However, again we need to be careful. We construct a new augmented program (10.40) that allows for the possibility that the values of $y_{\bullet 4}$ may need to be revised to obtain a lower value for z :

$$\begin{aligned} 0x_1 + 0x_2 + 3x_3 + \quad & 0x_4^h + y_{04}x_4 = z \quad (\text{min}) \\ x_1 & + \quad 1x_4^h + y_{14}x_4 = 1 \\ & x_2 + (1/2)x_4^h + y_{24}x_4 = 1 \\ & x_3 + (1/2)x_4^h + y_{34}x_4 = 1 \\ x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad x_4^h \geq 0, \quad x_4 \geq 0, \end{aligned} \tag{10.40}$$

where the $y_{\bullet 4}$ satisfy the same relations (10.33) as before. From (10.39) the new basic feasible solution is:

$$(-z) = -3/2, \quad x_2 = 1/2, \quad x_3 = 1/2, \quad x_4^h = 1, \quad x_1 = x_4 = 0.$$

In order to determine if it is optimal we compute the simplex multipliers:

$$\pi = (-3/2, 0, 3)^T.$$

We next obtain the reduced costs \bar{c}_1 and \bar{c}_4 as

$$\bar{c}_1 = 3/2, \quad \bar{c}_4 = y_{04} + (3/2)y_{14} - 3y_{34}, \quad \text{with } y_{04} = 2. \quad (10.41)$$

The test for a minimum is $\bar{c}_j \geq 0$ for all j . In this case, only \bar{c}_4 may possibly be less than zero depending on the values of the parameters y_{04} , y_{14} , and y_{34} . In order to determine if $\bar{c}_4 < 0$ is possible, we replace only the objective of (10.36) by the updated \bar{c}_4 from (10.41) and solve the new subprogram:

$$\begin{aligned} \text{Minimize} \quad & y_{04} + (3/2)y_{14} && - 3y_{34} = \bar{c}_4 \\ \text{subject to} \quad & - (1/2)y_{14} + y_{24} && = 0 \\ & - (1/2)y_{14} && + y_{34} = 1 \\ & y_{04} = 2, \quad y_{14} \geq 0, \quad y_{24} \geq 0, \quad y_{34} \geq 0. \end{aligned} \quad (10.42)$$

On applying the Simplex Method, we find the minimal value is $\bar{c}_4 = -1 < 0$ with $y_{04}^e = 2$, $y_{14}^e = 0$, $y_{24}^e = 0$, and $y_{34}^e = 1$. Thus, once again the solution fails to pass the optimality test for the original problem. The variable x_4 is the only nonbasic with a negative cost; bringing it into the basis causes x_3 to drop out and the new basic feasible solution is

$$(-z) = -1, \quad x_2 = 1/2, \quad x_4^h = 1, \quad x_4^e = 1/2, \quad x_1 = x_3 = 0.$$

However, again we need to be careful here. We construct a new augmented program (10.43), which allows for the possibility of $y_{\bullet 4}$ to be revised:

$$\begin{aligned} 0x_1 + 0x_2 + 3x_3 + \quad & 0x_4^h + 2x_4^e + y_{04}x_4 = z \quad (\text{min}) \\ x_1 \quad & + 1x_4^h + 0x_4^e + y_{14}x_4 = 1 \\ x_2 \quad & + (1/2)x_4^h + 0x_4^e + y_{24}x_4 = 1 \\ & x_3 + (1/2)x_4^h + 1x_4^e + y_{34}x_4 = 1 \\ x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \quad & x_4^h \geq 0, \quad x_4^e \geq 0, \quad x_4 \geq 0, \end{aligned} \quad (10.43)$$

where the $y_{\bullet 4}$ satisfy the same relations (10.33) as before. The new basic feasible solution is:

$$(-z) = -1, \quad x_2 = 1/2, \quad x_4^h = 1, \quad x_4^e = 1/2, \quad x_1 = x_3 = x_4 = 0. \quad (10.44)$$

In order to determine if it is optimal we compute the simplex multipliers

$$\pi = (-1, 0, 2)^T.$$

We next obtain the reduced costs \bar{c}_1 , \bar{c}_2 , and \bar{c}_4 as

$$\bar{c}_1 = 1, \quad \bar{c}_3 = 1, \quad \bar{c}_4 = y_{04} + y_{14} - 2y_{34} \quad \text{with } y_{04} = 2.$$

The test for a minimum is $\bar{c}_j \geq 0$ for all j . In this case, only \bar{c}_4 may possibly be less than zero depending on the values of the parameters y_{04} , y_{14} , and y_{34} . In order to determine if $\bar{c}_4 < 0$ is possible, we solve the subprogram:

$$\begin{aligned} \text{Minimize} \quad & y_{04} + y_{14} - 2y_{34} = \bar{c}_4 \\ \text{subject to} \quad & - (1/2)y_{14} + y_{24} = 0 \\ & - (1/2)y_{14} + y_{34} = 0 \\ & y_{04} = 2, \quad y_{14} \geq 0, \quad y_{24} \geq 0, \quad y_{34} \geq 0. \end{aligned} \tag{10.45}$$

On applying the Simplex Method, we find the the minimal value is $\bar{c}_4 = 0$ with $y_{04} = 2$, $y_{14} = 0$, $y_{24} = 0$, and $y_{34} = 2$. Since $\bar{c}_4 = 0$, the solution (10.44) is optimal because all the reduced costs are nonnegative for all feasible values of $y_{\bullet 4}$.

▷ **Exercise 10.8** From the final solution (10.44) derive the optimal solution to (10.32) and (10.33).

▷ **Exercise 10.9** In Example 10.3 change $y_{04} = 2$ to be $y_{04} = 8$ in equations 10.33 and re-solve the problem.

▷ **Exercise 10.10** This exercise is designed to show how the homogeneous part of the solution can dominate and drive out the extreme-point part of the solution. Solve the following generalized linear program:

$$\begin{aligned} x_1 + 0x_2 + 0x_3 &= z \quad (\text{min}) \\ x_1 + y_{13}x_3 &= 3 \\ x_2 + y_{23}x_3 &= 1 \\ x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0, \end{aligned} \tag{10.46}$$

where $y_{03} = c_3 = 0$, y_{13} , and y_{23} may be chosen to have any values satisfying

$$-y_{13} + 2y_{23} = 2 \quad \text{with } y_{i3} \geq 0 \text{ for } i = 1, 2 \tag{10.47}$$

by the following three approaches:

1. Use the method of substituting $u_{i4} = y_{i4}x_{i4}$ as described in Example 10.1.
2. Use the method of Example 10.3.
3. Use the method of Example 10.3, except that when a class of solutions $y_{\bullet t} = y_{\bullet t}^e + \theta y_{\bullet t}^h \in \mathcal{C}_t$, where $y_{\bullet t}^e$ is an extreme-point solution and $y_{\bullet t}^h$ is a homogeneous solution with $\theta \geq 0$ a scalar parameter is obtained, insert it into the basis of the original problem together with the parameter θ ; see (10.31).

Comment on the three methods and the solutions thus obtained.

▷ **Exercise 10.11** Modify the steps of the algorithm to take care of the case when for some $j = s$ the subproblem \mathcal{C}_s turns out to be infeasible.

We shall now formalize the concepts discussed so far and prove that the algorithms just described converge.

Definition: The *Restricted Master Program* at the k th stage of the algorithm consists of variables $x_j^{(k)}$ with specified columns of $y_{\bullet j} = \begin{pmatrix} c_j^{(k)} \\ A_{\bullet j}^{(k)} \end{pmatrix}$ drawn from the convex set \mathcal{C}_j .

The optimal solution of the restricted master program determines values for the simplex multipliers, $\pi = \pi^o$, for use in subprograms.

Definition: The j th subprogram at any stage is to find $y_{\bullet j} \in \mathcal{C}_j$ which minimizes the linear form $\bar{c}_j = y_{0j} - \sum_{i=1}^m y_{ij} \pi_i^o$ where $\pi = \pi^o$ are the known simplex multipliers from the restricted master program.

If the j th subprogram has a finite optimal solution $y_{\bullet j} = y_{\bullet j}^*$, it generates an additional specified column of coefficients for the next restricted master program. If it has an unbounded class of solutions, $y_{\bullet j} = y_{\bullet t}^e + \theta y_{\bullet t}^h$, $\theta \geq 0$, then $y_{\bullet t}^h$ is used to generate an additional specified column of coefficients for the next restricted master program. If the subproblem turns out to be infeasible, set $x_j = 0$ permanently and set a flag not to solve the subproblem j on any subsequent iteration or find some other way to tell the computer to *drop* x_j and its column of coefficients from the problem.

THEOREM 10.3 (Optimality Check) *A solution $(x_j^*, y_{\bullet j}^*)$ for $j = 1, \dots, n$ is optimal if there exists multipliers π , such that $\bar{c}_j \geq 0$ for all $y_{\bullet j} \in \mathcal{C}_j$ and $\bar{c}_j = 0$ for all $x_j^e > 0$ or $x_j^h > 0$.*

THEOREM 10.4 (Finite Termination) *The Simplex Algorithm will terminate in a finite number of iterations if each basic feasible solution is improved until it is no longer possible to find either an extreme point $y_{\bullet s} = y_{\bullet s}^e \in \mathcal{C}_s$ or a homogeneous extreme direction $y_{\bullet s} = y_{\bullet s}^h$ of \mathcal{C}_s to introduce into the basis such that*

$$\bar{c}_s = \min_{\substack{y_{\bullet j} \in \mathcal{C}_j \\ j=1, \dots, n}} \bar{c}_j = y_{0j} - \sum_{i=1}^m y_{ij} \pi_i^o < 0, \quad (10.48)$$

where π are the simplex multipliers of the basis in the master program.

Proof. In order to show finiteness of the algorithm it is easy to show that the columns of any basis of the master program must be drawn from a finite class. Each \mathcal{C}_j is a convex set defined by a finite number of linear inequalities; therefore it can be represented by a finite number of extreme points and a finite number of extreme homogeneous directions (see Theorem 10.5 on Page 281). ■

▷ **Exercise 10.12** In the discussion so far we have assumed that b is fixed. Show how to modify the approach so that b can be chosen freely from a convex set C_b .

- ▷ **Exercise 10.13** Reduce to a generalized linear program the problem: Find vector $y_{\bullet j} \in C_j \subseteq \mathbb{R}^m$ and $b \in C_b \subseteq \mathbb{R}^n$ such that

$$\sum_{j=1}^n y_{\bullet j} = b. \quad (10.49)$$

- ▷ **Exercise 10.14** Referring to Example 10.1 on Page 269, suppose that

$$C_4 = \{ y_{\bullet 4} \mid y_{04}^2 + y_{14}^2 + y_{24}^2 \leq 1 \}. \quad (10.50)$$

Show that this condition also results in a generalized linear program. Apply the methods of this section to solve the problem; contrast it with the polyhedral case. Suppose that instead of (10.50)

$$C_4 = \{ y_{\bullet 4} \mid y_{04}^2 + y_{14}^2 + y_{24}^2 = 1 \}. \quad (10.51)$$

Prove that the method of this section would still be applicable even though C_4 is no longer a convex set.

Comment: Once a generated column of the basis of the master program is dropped, we have the *option* of keeping it as a nonbasic column of the master program or dropping it because it can always be regenerated if needed later on as an extreme point or as the direction of an extreme half-line of some C_j . Experiments show that if the restricted master problem has not grown too large it is clearly advantageous to keep all such dropped generated columns as nonbasic columns of the master. However, the size of the restricted master program may grow too large and, if it does, we can reduce the size by dropping some of its nonbasic columns by using some rule such as dropping those that price out the most positive or dropping those that have remained nonbasic for the longest consecutive number of iterations.

10.2 DANTZIG-WOLFE (D-W) DECOMPOSITION PRINCIPLE

The Dantzig-Wolfe Decomposition Principle is based on the *Resolution* or *Representation* theorem for convex polyhedra (see Theorem 10.5). Before stating and proving the resolution (or representation) theorem, we define a *normalized extreme homogeneous solution*.

Definition: The *normalized extreme homogeneous solutions* associated with the basic feasible solutions of a convex polyhedral set

$$\begin{aligned} Ax &= b \\ x &\geq 0 \end{aligned} \quad (10.52)$$

are the basic feasible solutions to the following system of equations

$$\begin{aligned} Ay &= 0 \\ e^T y &= 1 \\ y &\geq 0 \end{aligned} \quad (10.53)$$

where $e = (1, 1, \dots, 1)^T$.

In (10.53), the first set of m equations, $Ay = 0$, implies that y is a homogeneous solution; the convexity and nonnegativity constraints on y state that the solutions are nonzero and the variables are normalized to sum to 1. The number of basic feasible solutions are clearly finite for (10.53). (Note that the reason that we normalize the homogeneous solutions is to get a finite number of them).

▷ **Exercise 10.15** If the set of feasible solutions to (10.52) is unbounded, prove it has at least one extreme homogeneous solution.

▷ **Exercise 10.16** Construct an example to show it is possible for (10.53) to have a feasible solution while (10.52) has none.

THEOREM 10.5 (Resolution) *Every feasible solution of a convex polyhedral set of the form $Ax = b$, $x \geq 0$, can be represented as a convex combination of the finite set of its extreme points and a nonnegative linear combination of the finite set of its normalized extreme homogeneous solutions (i.e., the finite set of directions of its extreme half-line solutions).*

Proof. The theorem states that every feasible solution $x = \bar{x}$ of

$$\begin{aligned} Ax &= b \\ x &\geq 0, \end{aligned} \tag{10.54}$$

can be represented in the form

$$\begin{aligned} \bar{x} &= \sum_{i=1}^L \alpha_i u^i + \sum_{j=1}^M \beta_j v^j \\ 1 &= \sum_{i=1}^L \alpha_i \\ \alpha_i &\geq 0 \text{ for } i = 1, \dots, L, \beta_j \geq 0 \text{ for } j = 1, \dots, M, \end{aligned} \tag{10.55}$$

where $\{u^i\}$ are the finite set of all extreme points, and $\{v^j\}$ are the finite set of all normalized extreme homogeneous solutions.

First, suppose that $x = \bar{x}$ is defined by (10.55) for some choice of α_i , $i = 1, \dots, L$ and β_j , $j = 1, \dots, M$. Note that $x = \bar{x} \geq 0$ because $\alpha_i \geq 0$, $\beta_j \geq 0$, $u^i \geq 0$, $v^j \geq 0$ for all i, j . Moreover, since $Au^i = b$ for $i = 1, \dots, L$ and $Av^j = 0$ for $j = 1, \dots, M$ we have

$$\begin{aligned} A\bar{x} &= A \sum_{i=1}^L \alpha_i u^i + A \sum_{j=1}^M \beta_j v^j = \sum_{i=1}^L \alpha_i (Au^i) + \sum_{j=1}^M \beta_j (Av^j) \\ &= \sum_{i=1}^L \alpha_i b + \sum_{j=1}^M \beta_j (0) = b. \end{aligned}$$

Next, we show conversely that given any feasible solution $x = \bar{x} \geq 0$ of (10.54), there exist $\alpha_i \geq 0$, $\beta_j \geq 0$ satisfying (10.55). To show this *assume on the contrary*, that there exist no α_i , β_j that satisfy

$$\begin{aligned} \sum_{i=1}^L \alpha_i u^i + \sum_{j=1}^M \beta_j v^j &= \bar{x} \\ \sum_{i=1}^L \alpha_i &= 1 \\ \alpha_i &\geq 0, \quad i = 1, \dots, L, \\ \beta_j &\geq 0, \quad j = 1, \dots, M. \end{aligned} \tag{10.56}$$

This implies, by the Infeasibility Theorem (see *Linear Programming 1*), that there exists at least one set of multipliers $\bar{\pi} = (\bar{\pi}_1, \bar{\pi}_2, \dots, \bar{\pi}_n)^T$ and $\gamma = \bar{\gamma}$, not all zero, such that

$$\begin{aligned} \text{(a)} \quad & \bar{\pi}^T u^i + \bar{\gamma} \geq 0 \quad \text{for } i = 1, \dots, L \\ \text{(b)} \quad & \bar{\pi}^T v^j \geq 0 \quad \text{for } j = 1, \dots, M \\ \text{(c)} \quad & \bar{\pi}^T \bar{x} + \bar{\gamma} < 0, \end{aligned} \tag{10.57}$$

has a feasible solution.

We now show that (10.57) is in fact infeasible, contrary to our assumption that (10.56) is infeasible. We do this by examining the following linear program

$$\begin{aligned} \text{Minimize} \quad & \bar{\pi}^T w \\ \text{subject to} \quad & Aw = b, \\ & w \geq 0, \end{aligned} \tag{10.58}$$

whose objective coefficients are the $\bar{\pi}$ satisfying (10.57), and whose set of extreme point solutions $\{u^i\}$, satisfying $Au^i = b$, and extreme homogeneous solutions $\{v^j\}$, satisfying $Av^j = 0$, are the same as those for (10.54). This linear program (10.58) is feasible since $w = \bar{x}$ is a feasible solution by hypothesis.

The linear program (10.58) clearly has a finite minimum because every extreme homogeneous solution satisfies $\bar{\pi}^T v^j \geq 0$ by (10.57b); i.e., the objective is nondecreasing along every extreme homogeneous solution. *This implies that the minimum value of the objective must occur at an extreme point.* On subtracting the third relation of (10.57c) from the first set of relations (10.57a) for each $i = 1, \dots, M$ we obtain the relations

$$\bar{\pi}^T u^i > \bar{\pi}^T \bar{x} \quad \text{for } i = 1, \dots, M.$$

Clearly a contradiction since this states that objective value at the given feasible point $w = \bar{x} \geq 0$ is strictly smaller than the objective value at every extreme point. Therefore, we conclude that there must exist $\alpha_i \geq 0$, $\beta_j \geq 0$ satisfying (10.56). ■

Example 10.4 (Illustration of the Resolution Theorem) This illustrates how a feasible solution of a convex polyhedral set can be represented as a convex combination of the finite set of extreme points and a nonnegative linear combination of the finite set

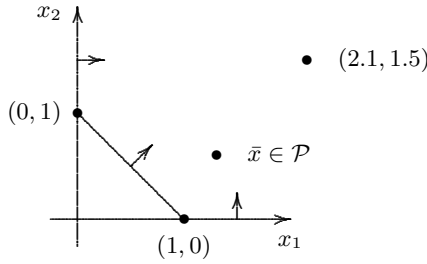


Figure 10-1: Illustration of the Resolution Theorem

of normalized extreme homogeneous solutions as proved in Theorem 10.5. Consider the polyhedral set

$$\mathcal{P} = \{ x \in \mathbb{R}^2 \mid x_1 + x_2 \geq 0, x_1 \geq 0, x_2 \geq 0 \} \tag{10.59}$$

shown in Figure 10-1. From the figure we see:

- Extreme Points: $(1, 0)^T, (0, 1)^T$
- Half-Lines: $(x_1 \geq 0, x_2 = 0), (x_1 = 0, x_2 \geq 0)$
- Extreme Directions Away from Origin (Normalized): $(1, 0)^T, (0, 1)^T$

The Resolution Theorem 10.5, in this case, states that any point $\bar{x} \in \mathcal{P}$ can be expressed as

$$\bar{x} = \alpha_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \alpha_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \beta_1 \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

where $\alpha_1 + \alpha_2 = 1, \alpha_1 \geq 0, \alpha_2 \geq 0, \beta_1 \geq 0, \beta_2 \geq 0$. For example $\bar{x} = (2.1, 1.5)^T$ can be represented with $\alpha_1 = 0, \alpha_2 = 1, \beta_1 = 2.1, \beta_2 = 0.5$ or $\alpha_1 = 1, \alpha_2 = 0, \beta_1 = 1.1, \beta_2 = 1.5$ or any convex combination of these two.

▷ **Exercise 10.17** Consider the convex polyhedral set in \mathbb{R}^3 given by $x_1 \geq 0, x_2 \geq 0$, and x_3 unrestricted. Show that it has no extreme points. Write it in an equivalent form $Ay = b, y \geq 0$, with $y \in \mathbb{R}^4$.

THEOREM 10.6 (D-W Transformation) *Every feasible solution of a convex polyhedral set in \mathbb{R}^n can be represented as a convex combination of a finite set of feasible solutions and a nonnegative linear combination of the finite set of normalized extreme homogeneous solutions (i.e., the finite set of directions of the extreme half-line solutions).*

▷ **Exercise 10.18** Prove Theorem 10.6 by showing that every convex polyhedral set of $y \in \mathbb{R}^n$ can be represented in the form

$$\begin{aligned} Ax &= b \\ x &\geq 0, \end{aligned} \tag{10.60}$$

where $x \in \mathbb{R}^{\bar{n}}$ with $\bar{n} \geq n$ and applying the Resolution Theorem 10.5.

10.2.1 D-W PRINCIPLE

We illustrate the Dantzig-Wolfe Decomposition Principle by applying it to a general linear program in standard form:

$$\begin{array}{ll} \text{Minimize} & c^T x = z \\ \text{subject to} & Ax = b, \quad A : m \times n, \\ & x \geq 0. \end{array} \quad (10.61)$$

Let $Ax = b$ be arbitrarily partitioned into two sets of equations $A^1x = b^1$ and $A^2x = b^2$ where A^1 is $m_1 \times n$ and A^2 is $m_2 \times n$ and $m = m_1 + m_2$. That is,

$$\begin{array}{ll} \text{Minimize} & c^T x = z \\ \text{subject to} & A^1x = b^1, \quad A^1 : m_1 \times n, \\ & A^2x = b^2, \quad A^2 : m_2 \times n, \\ & x \geq 0. \end{array} \quad (10.62)$$

In this case, we view the problem as

$$\begin{array}{ll} \text{Minimize} & c^T x = z, \\ \text{subject to} & A^1x = b^1, \quad A^1 : m_1 \times n, \end{array} \quad (10.63)$$

subject to the additional constraints

$$\begin{array}{ll} A^2x = b^2, & A^2 : m_2 \times n, \\ x \geq 0. \end{array} \quad (10.64)$$

From the Resolution Theorem 10.5, we know that any feasible solution to (10.64) can be written as a convex linear combination of the L possible extreme points (basic feasible solutions) $x = u^i$ and a nonnegative linear combination of the M possible normalized extreme homogeneous solutions $x^h = v^j$ of (10.64), i.e.,

$$x = \sum_{i=1}^L \alpha_i u^i + \sum_{j=1}^M \beta_j v^j, \quad (10.65)$$

where $\sum_{i=1}^L \alpha_i = 1$, $\alpha_i \geq 0$, $i = 1, \dots, L$, and $\beta_j \geq 0$, $j = 1, \dots, M$.

THEOREM 10.7 (Equivalent Full Master Program) *Substituting (10.65) into (10.63) transforms the original $m \times n$ linear program into an equivalent linear program with fewer rows ($m_1 + 1$) and possibly many more columns ($L + M$), where m_1 is the number of rows of A^1 , and where L is the number of extreme solutions and M is the number of normalized homogeneous solutions of $\{x \mid A^2x = b^2, x \geq 0\}$;*

that is:

FULL MASTER PROGRAM

Find minimum z , $\alpha_i \geq 0$, $i = 1, \dots, L$, $\beta_j \geq 0$, $j = 1, \dots, M$, such that

$$\begin{aligned} \sum_{i=1}^L (c^T u^i) \alpha_i + \sum_{j=1}^M (c^T v^j) \beta_j &= z \\ \sum_{i=1}^L (A^1 u^i) \alpha_i + \sum_{j=1}^M (A^1 v^j) \beta_j &= b^1 \\ \sum_{i=1}^L \alpha_i &= 1 \end{aligned} \tag{10.66}$$

▷ **Exercise 10.19** Prove Theorem 10.7.

Definition (Full Master Program or the Extremal Problem): The linear program obtained from the extreme point solutions $x = u^i$ and normalized extreme homogeneous solutions $x^h = v^j$ of (10.64) is called the equivalent *full master program* or the *extremal problem*. Note that the rows, other than the convexity constraint, of this master program are in one-to-one correspondence with the rows of the first partition.

THEOREM 10.8 (Feasible and Optimal Solution) Any α_i and β_j satisfying (10.66), determines an x by (10.65) which is a feasible solution to (10.62). If, in addition, z is the minimum of (10.66) for $\alpha_i = \alpha_i^*$ and $\beta_j = \beta_j^*$, then by (10.65) these generate an optimal feasible solution $x = x^*$ to (10.61). If this optimal feasible solution to (10.61) is not unique, then x^* need not be basic feasible solution to the original problem (10.61).

▷ **Exercise 10.20** Prove Theorem 10.8.

To simplify notation, denote the linear transforms G^i and H^j of u^i and v^j by:

$$\begin{aligned} G^i &= A^1 u^i \\ H^j &= A^1 v^j \end{aligned} \tag{10.67}$$

and the associated scalar costs g_i and h_j by:

$$\begin{aligned} g_i &= c^T u^i \\ h_j &= c^T v^j. \end{aligned} \tag{10.68}$$

The original linear program is then equivalent to:

FULL MASTER PROGRAM

Find minimum $z, \alpha_i \geq 0, i = 1, \dots, L, \beta_j \geq 0, j = 1, \dots, M$, such that

$$\begin{aligned} \sum_{i=1}^L g_i \alpha_i + \sum_{j=1}^M h_j \beta_j &= z && : \text{Dual Variables} \\ \sum_{i=1}^L G^i \alpha_i + \sum_{j=1}^M H^j \beta_j &= b^1 && : \pi \\ \sum_{i=1}^L \alpha_i &= 1 && : \gamma \end{aligned} \tag{10.69}$$

So far we have transformed the original problem (10.61) to (10.69), a linear program, called the full master program, of lower row dimension. Unfortunately this transformation usually results, in practice, with many, many more columns corresponding to *all* M basic feasible solutions and *all* L normalized extreme nonnegative homogeneous directions of (10.64). In practice it is usually impractical to generate a full master program consisting of L extreme points plus M extreme homogeneous directions. What is done instead is to generate at each iteration of the Simplex Algorithm just that column of the full master program, which the Algorithm would have selected to try to bring into the basis.

Definition:

The program obtained by dropping all but a subset \bar{L} of the L columns associated with α_i and a subset \bar{M} of the M columns associated with β_j for (10.69) is called a *Restricted Master Program*.

Let us suppose after a number of iterations that we have generated a restricted master program whose columns correspond to a basic feasible solution to the full master problem with values of the basic variables $\alpha_i = \alpha_i^0, i = 1, \dots, k$, and $\beta_j = \beta_j^0, j = 1, \dots, l$. Since (10.69) has $m_1 + 1$ rows (excluding the objective), $k + l = m_1 + 1$. (Later, in Section 10.2.4, we will discuss how to obtain such a starting basic feasible solution).

Next, let the simplex multipliers associated with this basic feasible solution be $(\bar{\pi}, \bar{\gamma})$ where the components of $\bar{\pi}$ correspond to m_1 rows of b_1 and $\bar{\gamma}$ is the the simplex multiplier of the convexity constraint be denoted by γ . To test whether this basic feasible solution is an optimal solution of the Full Master problem we use these multipliers to “price out” all $M + N$ of its columns.

Key Idea: We will now show how to generate *only* the column having the most negative reduced cost *without* having to generate and price out all the remaining columns of the master.

The simplex multipliers by definition satisfy the following equations:

$$\begin{aligned} \gamma + (G^i)^T \pi &= g_i && i = 1, \dots, k \\ (H^j)^T \pi &= h_j && j = 1, \dots, l \end{aligned} \tag{10.70}$$

If we used the commonly used rule of selecting the most negative reduced cost we would first need to determine $i = i^*$ and $j = j^*$ such that

$$g_{i^*} - (G^{i^*})^T \bar{\pi} - \bar{\gamma} = \min_{i=1, \dots, L} \{g_i - (G^i)^T \bar{\pi}\} - \bar{\gamma} \quad (10.71)$$

and

$$h_{j^*} - (H^{j^*})^T \bar{\pi} = \min_{j=1, \dots, M} \{h_j - (H^j)^T \bar{\pi}\}. \quad (10.72)$$

where i^* and j^* are the indices i and j at which these minima are achieved. We would then choose the minimum of (10.71) and (10.72) to be the index of the column to introduce into the basis. If we define as *Adjusted Costs*

$$\bar{\rho} = c - (A^1)^T \bar{\pi} \quad (10.73)$$

and substitute $G^i = A^1 u^i$ and $H^j = A^1 v^j$, we can rewrite the above two reduced-cost pricing-out equations (10.71) and (10.72) as

$$\begin{aligned} \min_{i=1, \dots, L} \{g_i - (G^i)^T \bar{\pi}\} - \bar{\gamma} &= \min_{i=1, \dots, L} \{c^T u^i - (A^1 u^i)^T \bar{\pi}\} - \bar{\gamma} \\ &= \min_{i=1, \dots, L} \{(c - (A^1)^T \bar{\pi})^T u^i\} - \bar{\gamma} \\ &= \min_{i=1, \dots, L} \{\bar{\rho}^T u^i\} - \bar{\gamma} \end{aligned} \quad (10.74)$$

and

$$\begin{aligned} \min_{j=1, \dots, M} \{h_j - (H^j)^T \bar{\pi}\} &= \min_{j=1, \dots, M} \{c^T v^j - (A^1 v^j)^T \bar{\pi}\} \\ &= \min_{j=1, \dots, M} \{(c - (A^1)^T \bar{\pi})^T v^j\} \\ &= \min_{j=1, \dots, M} \{\bar{\rho}^T v^j\}. \end{aligned} \quad (10.75)$$

At this point in the algorithm we do not know if there are any extreme points or normalized extreme solutions that price out negative. To determine (10.74) *without having to evaluate all the extreme point solutions to (10.64) that price out negative*, we determine instead $x = u^*$, which solves

$$\begin{aligned} \text{Minimize} \quad & \bar{\rho}^T x = z_2 + \bar{\gamma} \\ \text{subject to} \quad & A^2 x = b^2 \\ & x \geq 0, \end{aligned} \quad (10.76)$$

where $\bar{\rho} = c - (A^1)^T \bar{\pi}$ satisfies (10.73) and where $\bar{\gamma}$ is the value of the simplex multiplier of the convexity constraint of the Restricted Master Problem.

If after using the Simplex Method, the optimal solution is a basic feasible solution $x = u^*$, then we have found a column with the smallest reduced cost for the Full Master Program. If u^* is optimal, and if

$$\min z_2 < 0 \quad (10.77)$$

we augment the columns of the restricted master program by

$$\begin{pmatrix} G^* \\ g^* \\ 1 \end{pmatrix} = \begin{pmatrix} A^1 u^* \\ c^T u^* \\ 1 \end{pmatrix} \quad (10.78)$$

with index $*$, relabeled appropriately, then augment the restricted master program by this column and reoptimize the augmented restricted master program. On the other hand, if $\min z_2 = 0$ then all the reduced costs for the full master program (10.69) will be nonnegative. In this case we are at an optimal solution of the full master program (see Theorem 10.9), and (10.65) can be used to compute an optimal solution to (10.61).

Earlier we set aside the possibility that there are *extreme homogeneous solutions*. We now assume that after solving the subprogram (10.76) by the Simplex Method an extreme homogeneous solution $x^h = v^*$ is obtained. In this case, according to the theory, we are interested in obtaining the best normalized extreme homogeneous solution. This can be done *without having to evaluate all the normalized extreme homogeneous solutions* to (10.64) that price out negative by solving the linear program:

$$\begin{array}{ll} \text{Minimize} & \bar{\rho}^T x = z_2^h \\ \text{subject to} & A^2 x = 0 \\ & e^T x = 1, \\ & x \geq 0, \end{array} \quad (10.79)$$

where $e = (1, 1, \dots, 1)^T$, $\bar{\rho}$ satisfies (10.73), and z_2^h is the value associated with the optimal normalized homogeneous solution. Because setting up and solving this new subprogram (10.79) requires additional work, typically this linear program is not solved. Instead we accept any homogeneous solution $x^h = v^*$ of (10.76) that prices out negative without bothering to normalize it and compute and augment the restricted master program by

$$\begin{pmatrix} H^* \\ h^* \\ 0 \end{pmatrix} = \begin{pmatrix} A^1 v^* \\ c^T v^* \\ 0 \end{pmatrix} \quad (10.80)$$

with index $*$, relabeled appropriately, and bring this column into the basis of the augmented restricted master program and then reoptimize the augmented restricted master program.

- ▷ **Exercise 10.21** Why is it not necessary to normalize the homogeneous solution v^* before augmenting the restricted master program?

The D-W decomposition algorithm is sometimes also referred to in the literature as a *Delayed Column Generation Procedure* (we prefer to call it *Wait-and-See Column Generation Procedure*) since we generate only the column of the full master program that is coming into the basis. In summary, at each iteration we solve a

restricted master program that provides the information used to generate the new objective for the subprogram. The subprogram is then re-solved in order to generate a new incoming column for the next iteration of the restricted master.

10.2.2 D-W DECOMPOSITION ALGORITHM AND VARIANTS

10.2.2.1 The D-W ALGORITHM

So far we have described one iteration of the Dantzig-Wolfe algorithm. In this section we formalize its steps.

Algorithm 10.1 (Dantzig-Wolfe Algorithm)

1. An initial restricted master program with a starting basic feasible solution is given with one nonbasic column.
2. Solve the restricted master program. If a finite optimal solution is obtained go to Step 3. Otherwise report the original problem as unbounded (see Exercise 10.22) and stop.
3. The optimal basic feasible solution of the restricted master program provides us with simplex multipliers $\begin{pmatrix} \bar{\pi} \\ \bar{\gamma} \end{pmatrix}$. The nonbasic column is dropped.
4. The subprogram (10.76) is then solved.
5. If an optimal basic feasible solution is obtained to (10.76) and $\min z_2 < 0$, see (10.77); then a new column (10.78) is added to the restricted master program and the process is continued by going to Step 2.
6. If an extreme nonnegative homogeneous solution is obtained to (10.76), a new column (10.80) is added to the restricted master and the process is continued by going to Step 2.
7. If $\min z_2 = 0$, the solution is declared to be optimal for the original problem. The optimal solution is then given by (10.65) where $\alpha_1, \alpha_2, \alpha_3, \dots$ and $\beta_1, \beta_2, \beta_3, \dots$ is an optimal feasible solution to the final restricted master program, which is also an optimal feasible solution to the full restricted master program.

- ▷ **Exercise 10.22** Show how to display an unbounded solution to the original problem.
- ▷ **Exercise 10.23** Given an optimal solution $\alpha_1, \alpha_2, \alpha_3, \dots$ and $\beta_1, \beta_2, \beta_3, \dots$ to the Master Problem (10.69), show how to construct an optimal solution to the original problem.
- ▷ **Exercise 10.24** Show that if the original problem does not have a unique optimal solution, then the optimal solution to the Master Problem (10.69) may be a convex combination of several optimal basic feasible solutions to the original problem.

Computational Note: On each iteration of the D-W algorithm, we can start the solution of the new subprogram from the last basic solution of the preceding iteration. Thus, no Phase I procedure is necessary for any subprogram except the first. It has been observed in practice that most of the time only a few iterations are necessary to re-solve the subprogram. Some authors, especially Beale, question the need at each iteration to fully optimize each subproblem with respect to the current optimal prices $\pi = \bar{\pi}$ of the Restricted Master. They report good results, returning to the Restricted Master with improving, but not necessarily optimal, basic feasible solutions to the subproblems.

10.2.2.2 VARIANTS OF THE D-W ALGORITHM

As various columns of the full master program are generated for the restricted master program, each column that drops from the current basis of a restricted master is also dropped from the current restricted master program. Instead, one of the following two variants can be used.

1. Each column that drops from the current basis of a restricted master is retained as a supplementary column in the current restricted master program. This variant of retaining the nonoptimal “dropped” columns is recommended unless these retained columns become too numerous.
2. The restricted master program is augmented by each new column and each column that drops out of the basis is retained until the available computer memory is used up. At this point, a subset of the columns that price out the most positive is dropped from the current restricted master program.

10.2.3 OPTIMALITY AND DUAL PRICES

THEOREM 10.9 (Optimality and Finiteness under Nondegeneracy) *An optimal basic feasible solution of the restricted master program is also optimal for the full master program if*

$$\min z_2 = \bar{\gamma}, \quad (10.81)$$

see (10.76). If each restricted master program is nondegenerate such an optimum will be reached in a finite number of iterations.

Proof. The first part follows from the optimality conditions for a linear program. If the restricted master programs are nondegenerate the introduction of a new column into the restricted master program will decrease the objective function by a finite amount. Hence none of the finite number of bases of the full master program (10.69) can reappear, implying the iterative procedure is finite. ■

COROLLARY 10.10 (Finiteness under Degeneracy) *If some anticycling scheme is used, Theorem 10.9 also holds if the restricted master programs are degenerate.*

▷ **Exercise 10.25** Prove Corollary 10.10.

THEOREM 10.11 (Lower Bound on Optimal Objective Value) *Let $\alpha = \alpha^o$, $\beta = \beta^o$ be the current basic feasible solution and let $\pi = \bar{\pi}$, $\gamma = \bar{\gamma}$ be the multipliers at this solution. Then a lower bound on the optimal objective value is given by:*

$$\min z \geq z_o + \min z_2 \quad (10.82)$$

where $\min z_2$ is an extreme-point solution to (10.76).

▷ **Exercise 10.26** Prove Theorem 10.11.

LEMMA 10.12 (Dual-Feasible Solution) *Let π^1 be the multipliers on the first set of m_1 constraints of the master program and let π^2 be the multipliers for the m_2 constraints of the subprogram (10.64). Then $\begin{pmatrix} \pi^1 \\ \pi^2 \end{pmatrix}$ constitutes a feasible dual solution of the original linear program (10.61).*

▷ **Exercise 10.27** Prove Lemma 10.12.

THEOREM 10.13 (Optimal Basic Feasible Solution Representation) *An optimal basic feasible solution to the original linear program (10.61) can be represented as a convex combination of k basic feasible solutions and a nonnegative combination of l extreme homogeneous solutions of the subprogram where $m_1 = k + l$ is the dimension of b^1 .*

▷ **Exercise 10.28** Prove Theorem 10.13.

10.2.4 D-W INITIAL SOLUTION

So far we have assumed that an initial feasible solution is available to the full master program. In this section we show how to obtain an initial feasible solution by a Phase I procedure.

To start the process, we obtain a feasible solution u^1 to the subprogram (10.64). If no such feasible solution exists, quit. Defining $G^1 = A^1 u^1$ we set up the following restricted master program with artificial variables ξ_i .

$$\begin{aligned} & \text{Minimize} && \sum_{i=1}^{m_1} \xi_i = w \\ & \text{subject to} && G^1 \alpha_1 + \sum_{i=1}^{m_1} \pm e^i \xi_i = b^1 \\ & && \alpha_1 = 1 \\ & && \alpha_1 \geq 0, \xi_i \geq 0, i = 1, \dots, m_1, \end{aligned} \quad (10.83)$$

where e^i is the i th column of an $m_1 \times m_1$ identity matrix for $i = 1, \dots, m_1$. Note that for feasibility α_1 must equal 1. Hence, to ensure feasibility $+e_i$ is used if $b_i^1 - G_i^1 \geq 0$ and $-e_i$ is used if $b_i^1 - G_i^1 < 0$. The variables $\alpha_1, \xi_1, \xi_2, \dots, \xi_{m_1}$ constitute a basic set of variables for the restricted master (10.83). This Phase I problem is then solved by the procedure discussed in Section 10.2.2.1. At the end of Phase I either a feasible solution is obtained to the original problem or the problem is infeasible. If Phase I ends with a feasible solution, all nonbasic artificial variables are dropped and all basic artificial variables are maintained at 0 by setting their upper bounds to 0. Next the Phase I objective coefficients are replaced for the columns in the Restricted Master and the Restricted Master is re-solved. The process then continues.

10.2.5 D-W ALGORITHM ILLUSTRATED

In this section, we illustrate the D-W algorithm using first the full master problem and then the steps of the D-W algorithm by generating the columns of the D-W algorithm using the prices generated by the Restricted Master Problem.

Example 10.5 (Illustration of the Full Master Problem) Consider the following linear program:

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}		
1	2	3	4	5	6	1	2	3	4	5	6	7	-10	=	z (min)
3	2	1	6	5	4	8	5	7	3	4	1	1	2	=	64
1	8	3	7	1	4	5	2	5	3	2	6	3	4	=	63
1	1	1												=	3
			1	1	1									=	4
1			1											=	2
	1			1										=	1
		1			1									=	4
						1	1	1						=	4
									1	1	1			=	5
						1			1					=	3
							1			1				=	3
								1			1			=	3
												1	-1	=	1

where $x_j \geq 0$ for $j = 1, \dots, 14$. The problem can be thought of as consisting of the following partitions: the objective function, followed by two equality constraints, two transportation problems (called Sub1 and Sub2) whose variables are

$$\begin{bmatrix} x_1 & x_2 & x_3 \\ x_4 & x_5 & x_6 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x_7 & x_8 & x_9 \\ x_{10} & x_{11} & x_{12} \end{bmatrix}$$

and an equality constraint (called Sub3). That is, the problem can be partitioned into the form

$$\begin{aligned}
 (c^1)^T x^1 + (c^2)^T x^2 + (c^3)^T x^3 &= z \text{ (min)} \\
 A^1 x^1 + A^2 x^2 + A^3 x^3 &= b \\
 F^1 x^1 &= f^1 \\
 F^2 x^2 &= f^2 \\
 F^3 &= f^3
 \end{aligned}
 \tag{10.84}$$

$x^1 \geq 0, x^2 \geq 0, x^3 \geq 0.$

which is redisplayed below.

x_1^1	x_2^1	x_3^1	x_4^1	x_5^1	x_6^1	x_1^2	x_2^2	x_3^2	x_4^2	x_5^2	x_6^2	x_1^3	x_2^3	$z \text{ (min)}$	
1	2	3	4	5	6	1	2	3	4	5	6	7	-10	=	64
3	2	1	6	5	4	8	5	7	3	4	1	1	2	=	63
1	8	3	7	1	4	5	2	5	3	2	6	3	4	=	3
1	1	1												=	4
			1	1	1									=	2
1			1											=	1
	1			1										=	4
		1			1									=	4
						1	1	1						=	5
									1	1	1			=	3
						1			1					=	3
							1			1				=	3
								1			1			=	3
												1	-1	=	1

The entire set of basic solutions for Sub1 is

-2 1 4 4	2 -3 4 4	2 1 0 4
2 1 0 4	2 1 0 4	2 1 1 3
3 -1 1 4	-1 4 2 2	1 2 2 2
-1 4 3 1	3 2 -2 4	3 2 1 1

where the double-lined boxes are the basic feasible solutions excluding one degenerate case. (For example,

$$\begin{bmatrix} x_1 = -2, x_2 = 1, x_3 = 4, \\ x_4 = 4, x_5 = 0, x_6 = 0 \end{bmatrix}$$

is a basic (infeasible) solution.)

The entire set of basic solutions for Sub2 are

$\begin{matrix} -2 & 3 & 3 \\ & & 5 \end{matrix}$	$\begin{matrix} 3 & -2 & 3 \\ & 5 & \end{matrix}$	$\begin{matrix} 3 & 3 & -2 \\ & & 5 \end{matrix}$
$\begin{matrix} 1 & 3 \\ 2 & & 3 \end{matrix}$	$\begin{matrix} 3 & 1 \\ & 2 & 3 \end{matrix}$	$\begin{matrix} 3 & 1 \\ & 3 & 2 \end{matrix}$
$\begin{matrix} 4 \\ -1 & 3 & 3 \end{matrix}$	$\begin{matrix} & 1 & 3 \\ 3 & 2 & \end{matrix}$	$\begin{matrix} & 3 & 1 \\ 3 & 2 & \end{matrix}$
$\begin{matrix} 1 & & 3 \\ 2 & 3 & \end{matrix}$	$\begin{matrix} & 4 \\ 3 & -1 & 3 \end{matrix}$	$\begin{matrix} & & 4 \\ 3 & 3 & -1 \end{matrix}$

where again the double lined boxes are the basic feasible solutions. The third subproblem has one basic solution $x_{13} = 1, x_{14} = 0$, and one homogeneous solution $x_{13} = 1, x_{14} = 1$. The full master problem in the order corresponding to the basic feasible cases is:

α_1^1	α_2^1	α_3^1	α_4^1	α_1^2	α_2^2	α_3^2	α_4^2	α_5^2	α_6^2	α_1^3	β_1^3	
28	28	28	28	33	33	33	33	38	33	7	-3	= z (min)
24	24	24	24	32	40	45	43	33	47	1	3	= 64
26	18	36	28	35	39	38	30	32	32	3	7	= 63
1	1	1	1									= 1
				1	1	1	1	1	1			= 1
										1		= 1

▷ **Exercise 10.29** In Example 10.5, solve the original problem and the full master problem to verify that they produce the same optimal solution.

Example 10.6 (Illustration of the D-W Algorithm) We will now show how to solve the linear program shown in Example 10.5 by the D-W algorithm. We start by trying to generate a basic feasible solution to each of the three subproblems. If no such solution exists to any of the subproblems then the entire linear program is infeasible. The particular objective function we choose at this stage is not important; we assume that we use the actual objective costs for each subproblem. That is we solve:

$$\begin{aligned}
 & \text{Minimize} && 1x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 + 6x_6 = z_1 \\
 & \text{subject to} && x_1 + x_2 + x_3 = 3 \\
 & && x_4 + x_5 + x_6 = 4 \\
 \text{(Sub 1) :} & && x_1 + x_4 = 2 \tag{10.85} \\
 & && x_2 + x_5 = 1 \\
 & && x_3 + x_6 = 4 \\
 & && x_j \geq 0, \text{ for } j = 1, \dots, 6,
 \end{aligned}$$

$$\begin{aligned}
 & \text{Minimize} && 1x_7 + 2x_8 + 3x_9 + 4x_{10} + 5x_{11} + 6x_{12} = z_2 \\
 & \text{subject to} && x_7 + x_8 + x_9 = 4 \\
 & && x_{10} + x_{11} + x_{12} = 5 \\
 \text{(Sub 2) :} & && x_7 + x_{10} = 3 \tag{10.86} \\
 & && x_8 + x_{11} = 3 \\
 & && x_9 + x_{12} = 3 \\
 & && x_j \geq 0, \text{ for } j = 7, \dots, 12,
 \end{aligned}$$

and

$$\begin{aligned} & \text{Minimize} && 7x_{13} - 10x_{14} = z_3 \\ \text{(Sub 3):} & \text{subject to} && x_{13} - x_{14} = 1 \\ & && x_{13} \geq 0, x_{14} \geq 0. \end{aligned} \quad (10.87)$$

The optimal basic feasible solutions to the three subproblems are:

$$\begin{aligned} \text{Sub 1: } & u_1^{11} = 2, u_2^{11} = 1, u_4^{11} = 0, u_6^{11} = 4. \\ \text{Sub 2: } & u_2^{21} = 1, u_3^{21} = 3, u_4^{21} = 3, u_5^{21} = 2. \\ \text{Sub 3: } & u_1^{31} = 1. \end{aligned} \quad (10.88)$$

As noted on Page 285, the solutions obtained from the subproblems are transformed when put into the Master Problem. *The Phase II* objective evaluation will be done using as objectives:

$$c^1 = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}, \quad c^2 = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}, \quad c^3 = \begin{pmatrix} 7 \\ -10 \end{pmatrix}. \quad (10.89)$$

The Phase I objective evaluation will be done using:

$$w^1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad w^2 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad w^3 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \quad (10.90)$$

and the Restricted Master Problem coefficient matrix transformations will be done using

$$A^1 = \begin{pmatrix} 3 & 2 & 1 & 6 & 5 & 4 \\ 1 & 8 & 3 & 7 & 1 & 4 \end{pmatrix} \quad (10.91)$$

$$A^2 = \begin{pmatrix} 8 & 5 & 7 & 3 & 4 & 1 \\ 5 & 2 & 5 & 3 & 2 & 6 \end{pmatrix} \quad (10.92)$$

$$A^3 = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \quad (10.93)$$

This results in

$$G^{11} = A^1 u^{11} = \begin{pmatrix} 24 \\ 26 \end{pmatrix}$$

$$G^{21} = A^2 u^{21} = \begin{pmatrix} 43 \\ 30 \end{pmatrix}$$

$$G^{31} = A^3 u^{31} = \begin{pmatrix} 1 \\ 3 \end{pmatrix}$$

and

$$g^{11} = (w^1)^T u^{11} = 0$$

$$g^{21} = (w^2)^T u^{21} = 0$$

$$g^{31} = (w^3)^T u^{31} = 0$$

The corresponding associated objective coefficients are set to zero for Phase I. The Phase I Restricted Master Problem is then

$$\begin{array}{rll}
 \text{Minimize} & 0\alpha_{11} + 0\alpha_{21} + 0\alpha_{31} + 1x_{13} + 1x_{14} & = w \\
 \text{subject to} & 28\alpha_{11} + 43\alpha_{21} + \alpha_{31} - x_{13} & = 64 \\
 & 26\alpha_{11} + 30\alpha_{21} + 3\alpha_{31} + x_{14} & = 63 \\
 & \alpha_{11} & = 1 \\
 & & \alpha_{21} & = 1 \\
 & & & \alpha_{31} & = 1 \\
 & \alpha_{ki} \geq 0, x_j \geq 0. & & &
 \end{array} \tag{10.94}$$

On solving the Phase I restricted Linear Program (10.94), all five variables are basic and we obtain the following multipliers:

$$\bar{\pi} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad \text{and} \quad \bar{\gamma} = \begin{pmatrix} -2 \\ 13 \\ -2 \end{pmatrix}. \tag{10.95}$$

The multipliers are used to obtain the adjusted costs for the subproblem objectives as follows:

$$\begin{aligned}
 \bar{\rho}^1 &= w^1 - (A^1)^T \bar{\pi} = (2 \quad -6 \quad -2 \quad -1 \quad 4 \quad 16)^T \\
 \bar{\rho}^2 &= w^2 - (A^2)^T \bar{\pi} = (3 \quad 3 \quad 2 \quad 0 \quad 2 \quad -5)^T \\
 \bar{\rho}^3 &= w^3 - (A^3)^T \bar{\pi} = (-2 \quad -2)^T
 \end{aligned}$$

After solving the three subproblems with the above objective coefficients respectively, we obtain the following solutions:

$$\begin{aligned}
 \text{Sub 1 : } & u_2^{12} = 1, u_3^{12} = 2, u_4^{12} = 2, u_6^{12} = 2, z_1 = -12. \\
 \text{Sub 2 : } & u_1^{22} = 1, u_2^{22} = 3, u_4^{22} = 2, u_6^{22} = 3, z_2 = -3. \\
 \text{Sub 3 : } & v_1^{32} = 1, v_2^{32} = 1. \quad \text{Homogeneous Solution.}
 \end{aligned} \tag{10.96}$$

Note that

$$z_1 < \bar{\gamma}_1, \quad z_2 < \bar{\gamma}_2, \quad z_3 < \bar{\gamma}_3. \tag{10.97}$$

Hence we transform and insert each of the solutions into the Restricted Master Problem. The transformations are

$$\begin{aligned}
 G^{12} &= A^1 u^{12} = \begin{pmatrix} 24 \\ 36 \end{pmatrix} \\
 G^{22} &= A^2 u^{22} = \begin{pmatrix} 32 \\ 35 \end{pmatrix} \\
 H^{32} &= A^3 v^{32} = \begin{pmatrix} 3 \\ 7 \end{pmatrix}
 \end{aligned}$$

and

$$\begin{aligned} g^{12} &= (w^1)^T u^{12} = 0 \\ g^{22} &= (w^2)^T u^{22} = 0 \\ h^{32} &= (w^3)^T v^{32} = 0 \end{aligned}$$

The new Phase I Restricted Master Problem is then

$$\begin{aligned} \text{Minimize} \quad & 0\alpha_{11} + 0\alpha_{12} + 0\alpha_{21} + 0\alpha_{22} + 0\alpha_{31} + 0\alpha_{32} + 1x_{13} + 1x_{14} = w \\ \text{subject to} \quad & 28\alpha_{11} + 24\alpha_{12} + 43\alpha_{21} + 32\alpha_{22} + \alpha_{31} + 3\beta_{31} - x_{13} = 64 \\ & 26\alpha_{11} + 36\alpha_{12} + 30\alpha_{21} + 35\alpha_{22} + 3\alpha_{31} + 7\beta_{31} + x_{14} = 63 \\ & \alpha_{11} + \alpha_{12} = 1 \\ & \alpha_{21} + \alpha_{22} = 1 \\ & \alpha_{31} = 1 \\ & \alpha_{ki} \geq 0, \beta_{31} \geq 0, \text{artificial } x_{13} \geq 0, x_{14} \geq 0. \end{aligned} \tag{10.98}$$

After solving this, we obtain a basic feasible solution with all artificials out of the basis. We start Phase II with replacing the objective coefficients by the transformed coefficients:

$$\begin{aligned} g^{11} &= (c^1)^T u^{11} = 28 \\ g^{21} &= (c^2)^T u^{21} = 28 \\ g^{31} &= (c^3)^T u^{31} = 33 \\ g^{12} &= (c^1)^T u^{12} = 33 \\ g^{22} &= (c^2)^T u^{22} = 7 \\ h^{32} &= (c^3)^T v^{32} = -3 \end{aligned}$$

On optimizing the modified Phase II Restricted Master, the multipliers are:

$$\bar{\pi} = \begin{pmatrix} -0.1630 \\ -0.3587 \end{pmatrix} \quad \text{and} \quad \bar{\gamma} = \begin{pmatrix} 41.2391 \\ 50.7717 \\ 8.2391 \end{pmatrix}. \tag{10.99}$$

The new multipliers are used to obtain the adjusted costs for the subproblem objectives as follows:

$$\begin{aligned} \bar{\rho}^1 &= c^1 - (A^1)^T \bar{\pi} = (1.8477 \quad 5.1956 \quad 4.2391 \quad 7.4889 \quad 6.1737 \quad 8.0868)^T \\ \bar{\rho}^2 &= c^2 - (A^2)^T \bar{\pi} = (4.0975 \quad 3.5324 \quad 5.9345 \quad 5.5651 \quad 6.3694 \quad 8.3152)^T \\ \bar{\rho}^3 &= c^3 - (A^3)^T \bar{\pi} = (8.2391 \quad -8.2391)^T \end{aligned}$$

After solving the three subproblems with the above objective coefficients repectively, we obtain the following solutions:

$$\begin{aligned} \text{Sub 1 : } & u_1^{13} = 2, \quad u_3^{13} = 1, \quad u_5^{13} = 1, \quad u_6^{13} = 3, \quad z_1 = 38.3686 \\ \text{Sub 2 : } & u_2^{23} = 3, \quad u_3^{23} = 1, \quad u_4^{23} = 3, \quad u_6^{23} = 2, \quad z_2 = -3 \\ \text{Sub 3 : } & u_1^{33} = 1 \quad z_3 = 8.3291 \end{aligned} \tag{10.100}$$

Note that

$$z_1 < \bar{\gamma}_1, \quad z_2 < \bar{\gamma}_2, \quad z_3 = \bar{\gamma}_3. \tag{10.101}$$

Hence we transform and insert the solutions of subproblems 1 and 2 into the Restricted Master Problem. The transformations are

$$G^{13} = A^1 u^{13} = \begin{pmatrix} 24 \\ 18 \end{pmatrix}$$

$$G^{23} = A^2 u^{23} = \begin{pmatrix} 33 \\ 32 \end{pmatrix}$$

and

$$g^{13} = (c^1)^T u^{13} = 28$$

$$g^{23} = (c^2)^T u^{23} = 33$$

The new Restricted Master Problem is

$$\begin{array}{rll} \text{Minimize} & 28\alpha_{11} + 28\alpha_{12} + 28\alpha_{13} + 33\alpha_{21} + 33\alpha_{22} + 33\alpha_{23} + 7\alpha_{31} + -3\alpha_{32} = & z \\ \text{subject to} & 28\alpha_{11} + 24\alpha_{12} + 24\alpha_{13} + 43\alpha_{21} + 32\alpha_{22} + 33\alpha_{23} + \alpha_{31} + 3\beta_{31} = & 64 \\ & 26\alpha_{11} + 36\alpha_{12} + 18\alpha_{13} + 30\alpha_{21} + 35\alpha_{22} + 32\alpha_{23} + 3\alpha_{31} + 7\beta_{31} = & 63 \\ & \alpha_{11} + \alpha_{12} + \alpha_{13} & = 1 \\ & & \alpha_{21} + \alpha_{22} + \alpha_{23} & = 1 \\ & & & \alpha_{31} & = 1 \\ & \alpha_{ki} \geq 0, \beta_{31} \geq 0, x_j \geq 0. & & & \end{array} \tag{10.102}$$

On optimizing the Restricted Master, the multipliers are:

$$\bar{\pi} = \begin{pmatrix} -0.0789 \\ -0.3948 \end{pmatrix} \quad \text{and} \quad \bar{\gamma} = \begin{pmatrix} 37.0000 \\ 48.2367 \\ 8.2632 \end{pmatrix} \tag{10.103}$$

Once again, the new multipliers are used to obtain the adjusted costs for the subproblem objectives as follows:

$$\bar{\rho}^1 = c^1 - (A^1)^T \bar{\pi} = (1.6315 \quad 5.3164 \quad 4.2633 \quad 7.2370 \quad 5.7893 \quad 7.8940)^T$$

$$\bar{\rho}^2 = c^2 - (A^2)^T \bar{\pi} = (3.6052 \quad 3.1841 \quad 5.5263 \quad 5.4211 \quad 7.8940 \quad 8.4477)^T$$

$$\bar{\rho}^3 = c^3 - (A^3)^T \bar{\pi} = (8.2632 \quad 8.2632)^T$$

Upon solving the three subproblems with the above objective coefficients respectively, we obtain the following solutions:

$$\begin{array}{l} \text{Sub 1 : } u_1^{14} = 2, \quad u_3^{14} = 1, \quad u_5^{14} = 1, \quad u_6^{14} = 3, \quad z_1 = 37.0000 \\ \text{Sub 2 : } u_2^{24} = 1, \quad u_3^{24} = 3, \quad u_4^{24} = 3, \quad u_5^{24} = 2, \quad z_2 = 48.6327 \\ \text{Sub 3 : } u_1^{33} = 1 \quad z_3 = 8.2632 \end{array} \tag{10.104}$$

Since

$$z_1 = \bar{\gamma}_1, \quad z_2 = \bar{\gamma}_2, \quad z_3 = \bar{\gamma}_3, \tag{10.105}$$

the solution is optimal.

10.3 BENDERS DECOMPOSITION

Benders decomposition is Dantzig-Wolfe decomposition applied to the dual. Under this approach, the number of variables is reduced at the expense of usually adding many new inequalities. Analogous to generating columns of the D-W master only when needed, the inequalities of the Benders master are generated only when needed. Thus, it is a *Delayed Row-Generation Procedure*. Benders decomposition plays a central role in the solution of multistage stochastic linear programs (see Chapter 12). In this section we develop the theory of Benders decomposition as applied to solve a linear program of the following form.

$$\begin{aligned} \text{Maximize} \quad & (b^1)^T \pi^1 + (b^2)^T \pi^2 = \psi \\ \text{subject to} \quad & (A^1)^T \pi^1 + (A^2)^T \pi^2 \leq c \end{aligned} \tag{10.106}$$

where A^1 is $m_1 \times n$ and A^2 is $m_2 \times n$. Although developed independently of the D-W algorithm, its arithmetic steps turn out to be identical to solving by applying the D-W algorithm to its dual

$$\begin{aligned} \text{Minimize} \quad & c^T x = z \\ \text{subject to} \quad & A^1 x = b^1, \quad A^1 : m_1 \times n, \\ & A^2 x = b^2, \quad A^2 : m_2 \times n, \\ & x \geq 0, \end{aligned} \tag{10.107}$$

and interpreting its optimal conditions as the optimal solution of (10.106).

The D-W decomposition transforms a system (10.107) of $(m_1 + m_2)$ equations in n variables x into a system of $m_1 + 1$ equations and usually many more variables. Benders decomposition transforms a system of n inequalities in $m_1 + m_2$ variables (π^1, π^2) into a system in $m_1 + 1$ variables and usually many more inequalities. We sometimes refer to Benders decomposition as a way to *eliminate variables*. We shall describe the steps and the justification of the steps of Benders decomposition from these two perspectives.

10.3.1 DUAL OF D-W DECOMPOSITION

The *first way* of deriving the algorithm is a straightforward implementation of the following lemma:

LEMMA 10.14 (Benders Decomposition is Dual of D-W Decomposition)
Solving the original problem by applying the D-W Decomposition procedure to the dual of the original linear program (10.106) results in a procedure that is identical to the Benders Decomposition Procedure described in Section 10.3.2.

Thus, the elimination of the m_2 components of π^2 in (10.106) can be done by eliminating the last m_2 equations of its dual (10.107) using the D-W decomposition algorithm.

▷ **Exercise 10.30** Prove Lemma 10.14.

10.3.2 DERIVATION OF BENDERS DECOMPOSITION

We will now present *another way* of deriving the Benders Decomposition Algorithm. It is the one usually found in the literature. For ease of exposition, we *assume that* (10.106) *is feasible and has a finite optimum.*

By moving the terms in (10.106) corresponding to π^1 to the right-hand side of the inequality (see Exercise 10.31), we get:

$$\begin{aligned} & \text{Maximize}_{\pi^1} && (b^1)^T \pi^1 + \max_{\pi^2 | \pi^1} (b^2)^T \pi^2 \\ & \text{subject to} && (A^2)^T \pi^2 \leq c - (A^1)^T \pi^1. \end{aligned} \quad (10.108)$$

Holding π^1 fixed, we wish to:

$$\begin{aligned} & \text{Maximize}_{\pi^2 | \pi^1} && (b^2)^T \pi^2 = \psi_2(\pi^1) \\ & \text{subject to} && (A^2)^T \pi^2 \leq c - (A^1)^T \pi^1. \end{aligned} \quad (10.109)$$

To simplify the discussion, we *assume that there always exists some π^2 such that given π^1 , (π^1, π^2) is feasible for the original problem.* We shall now show how to obtain a solution to (10.109) under this assumption. Letting x be the dual variables corresponding to (10.109), we obtain:

$$\begin{aligned} & \text{Minimize}_{x | \pi^1} && (c - (A^1)^T \pi^1)^T x = \psi_2(\pi^1) \\ & \text{subject to} && A^2 x = b^2 \\ & && x \geq 0. \end{aligned} \quad (10.110)$$

The *Full Benders Master Program* is found by expressing every feasible solution of (10.110) as a convex linear combination of the extreme-point solutions $\{u^i\}$, for $i = 1, \dots, L$ plus a nonnegative linear combination of normalized extreme homogeneous solutions $\{v^j\}$, for $j = 1, \dots, M$ of (10.110). The Resolution Theorem 10.5 tells us that any feasible solution x of (10.110) can be written as:

$$x = \sum_{i=1}^L \alpha_i u^i + \sum_{j=1}^M \beta_j v^j, \quad (10.111)$$

for some choice of $\alpha_i \geq 0$, $i = 1, \dots, L$, where $\sum_{i=1}^L \alpha_i = 1$, and $\beta_j \geq 0$, $j = 1, \dots, M$.

Substituting (10.111) into (10.110) and noting that $A^2 u^i = b^2$, $A^2 v^j = 0$ for all i and j whatever be $\alpha_i \geq 0$, $\sum \alpha_i = 1$, $\beta_j \geq 0$, we obtain the following linear program.

Given π^1 , find minimum $\psi_2(\pi^1)$, and $\alpha_i \geq 0$, $\beta_j \geq 0$, such that

$$\begin{aligned} \sum_{i=1}^L \left[(c - (A^1)^T \pi^1)^T u^i \right] \alpha_i + \sum_{j=1}^M \left[(c - (A^1)^T \pi^1)^T v^j \right] \beta_j &= \psi_2(\pi^1) \\ \sum_{i=1}^L \alpha_i &= 1. \end{aligned} \quad (10.112)$$

Next, for (10.112) we note the following:

- Because of our feasibility assumptions of the original linear program, the optimal solution of (10.112) must be bounded for our choice of π^1 ; otherwise (10.109) would be infeasible. Thus, feasibility of (10.109) implies that π^1 satisfies

$$(c - (A^1)^T \pi^1)^T v^j \geq 0 \quad \text{for all } j = 1, \dots, M. \quad (10.113)$$

- Given that π^1 satisfies (10.113), an optimal solution must then occur at an extreme point of (10.110), that is

$$\psi_2^*(\pi^1) = \min_{1 \leq i \leq L} (c - (A^1)^T \pi^1)^T u^i. \quad (10.114)$$

Therefore the original linear program (10.106), which is the same as (10.109), is equivalent to

$$\begin{aligned} & \text{Maximize}_{\pi^1} \quad \left[(b^1)^T \pi^1 + \min_{1 \leq i \leq L} (c - (A^1)^T \pi^1)^T u^i \right] \\ & \text{subject to} \quad (c - (A^1)^T \pi^1)^T v^j \geq 0 \end{aligned} \quad (10.115)$$

It is easy to see that this then reduces to:

$$\begin{aligned} & \text{Maximize}_{\pi^1, \gamma} \quad (b^1)^T \pi^1 + \gamma \\ & \text{subject to} \quad (c - (A^1)^T \pi^1)^T u^i - \gamma \geq 0 \quad i = 1, \dots, L, \\ & \quad (c - (A^1)^T \pi^1)^T v^j \geq 0 \quad j = 1, \dots, M, \end{aligned} \quad (10.116)$$

where $(c - (A^1)^T \pi^1)^T v^j \geq 0$, $j = 1, \dots, M$, are the additional constraints to ensure that the choice of π^1 has the property, which we assumed, that there exists some π^2 such that (π^1, π^2) is feasible for the original problem. Substituting the definitions

$$\begin{aligned} G^i &= A^1 u^i, & H^j &= A^1 v^j, \\ g_i &= c^T u^i, & h_j &= c^T v^j, \end{aligned} \quad (10.117)$$

into (10.116) we obtain:

THE FULL BENDERS MASTER PROGRAM

Find maximum ψ , π^1 , γ unrestricted in sign, such that

$$\begin{aligned} & (b^1)^T \pi^1 + \gamma = \psi \\ & (G^i)^T \pi^1 + \gamma \leq g_i, \quad i = 1, \dots, L, \\ & (H^j)^T \pi^1 \leq h_j, \quad j = 1, \dots, M. \end{aligned} \quad (10.118)$$

THEOREM 10.15 (Benders Transformation Into an Equivalent LP) *Benders Full Master Program (10.118) transforms the original $n \times (m_1 + m_2)$ linear program into an equivalent linear program with fewer columns and possibly many more inequalities, namely, an $(L + M) \times (m_1 + 1)$, where m_1 is the number of rows of A^1 , L is the number of extreme solutions, and M is the number of normalized homogeneous solutions of $\{x \mid A^2 x = b^2, x \geq 0\}$, that is, linear program (10.118).*

▷ **Exercise 10.31** Prove that (10.106) is equivalent to (10.108).

▷ **Exercise 10.32** Prove that the assumption that the original problem (10.106) has a finite optimal feasible solution implies that (10.110) has feasible solutions and has an optimal feasible solution when π^1 has the property that there exists some π^2 such that (π^1, π^2) is feasible for the original problem (10.106).

Just as in the D-W decomposition, it is usually impractical to express explicitly the full set of basic feasible solutions and normalized extreme homogeneous solutions, so in Benders it is usually impractical to express explicitly the full set of inequalities for (10.118). To initiate an iterative step, assume that we have already inherited from earlier iterations a set of inequalities that is a subset \bar{L} of the first set of L inequalities and a subset \bar{M} of the second set of M inequalities (10.118). The linear program with $\bar{L} + \bar{M}$ inequalities is called the *Benders Restricted Master Program*. The inequalities themselves are called *cuts* (a term derived from Integer Programming, where each inequality generated “cuts off” a region in the feasible space where no integer solutions of interest lie).

Assuming that the $\bar{L} + \bar{M}$ inequality Benders Restricted Master has an optimal solution $\pi^1 = \bar{\pi}^1$ and $\gamma = \bar{\gamma}$, we generate a new inequality by letting the Adjusted Costs be

$$\bar{\rho} = c - (A^1)^T \bar{\pi}^1 \quad (10.119)$$

and solving:

$$\begin{aligned} \text{Minimize} \quad & \bar{\rho}^T x = z_1 \\ \text{subject to} \quad & A^2 x = b^2 \\ & x \geq 0. \end{aligned} \quad (10.120)$$

The solution of (10.120) gives rise to one of two cases:

1. *Optimality Cut*. If a finite optimal solution $x = u^*$ is obtained for (10.120), then it generates a new inequality

$$(G^{\bar{L}+1})^T \pi^1 + \gamma \leq g_{\bar{L}+1} \quad (10.121)$$

where $G^{\bar{L}+1} = A^1 u^*$ and $g_{\bar{L}+1} = c^T u^*$. This inequality is called an *optimality cut*.

2. *Feasibility Cut*. On the other hand, if an extreme homogeneous solution $x^h = v^*$ is obtained for (10.120), then it generates a new inequality

$$(H^{\bar{M}+1})^T \pi^1 \leq h_{\bar{M}+1}. \quad (10.122)$$

where $H^{\bar{M}+1} = A^1 v^*$ and $h_{\bar{M}+1} = c^T v^*$. This inequality is called a *feasibility cut*.

After augmenting the Benders Restricted Master Program by either the new optimality cut or feasibility cut indexed by $i = \bar{L} + 1$ or $j = \bar{M} + 1$, it is re-solved.

LEMMA 10.16 (Original Problem Infeasible) *If, after augmenting the Benders Restricted Master Program by a new feasibility cut and re-solving, we find that it is infeasible, then the original linear program is infeasible.*

▷ **Exercise 10.33** Prove Lemma 10.16.

THEOREM 10.17 (Optimality) *If the new optimality cut is feasible for $\pi = \bar{\pi}^1$ and $\gamma = \bar{\gamma}$, then $(\pi^1, \pi^2) = (\bar{\pi}^1, \bar{\pi}^2)$ is an optimal solution to the original linear program, where $\pi^2 = \bar{\pi}^2$ is an optimal dual solution to (10.120).*

▷ **Exercise 10.34** Prove Theorem 10.17.

Example 10.7 (Illustration of Benders Decomposition) Consider the following linear program

x_1	x_2	x_3	x_4	x_5	x_6		
1	1	1	3	2	1	=	z (min)
1	2	3				=	6
3	2	1				=	6
1	1	1	4	-1	1	=	9
3	2	1	3	2	1	=	15
4	-1	1	1	1	1	=	9

(10.123)

where $x_j \geq 0$ for $j = 1, \dots, 6$. The problem is in the following form

$$\begin{aligned} c^T x + f^T y &= z \text{ (min)} \\ Ax &= b \\ Bx + Dy &= d \\ x \geq 0, y &\geq 0, \end{aligned}$$

where $x^T = (x_1, x_2, x_3)$, $y^T = (y_1, y_2, y_3) = (x_4, x_5, x_6)$, $c^T = (1, 1, 1)$, $f^T = (3, 2, 1)$, $d^T = (9, 15, 9)$,

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 1 & 1 \\ 3 & 2 & 1 \\ 4 & -1 & 1 \end{pmatrix}, \quad \text{and} \quad D = \begin{pmatrix} 4 & -1 & 1 \\ 3 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

To solve the problem by Benders decomposition, we start by creating the Initial Restricted Master Problem

$$\begin{aligned} c^T x + \delta \theta &= z \text{ (min)} \\ Ax &= b \\ x &\geq 0, \end{aligned} \tag{10.124}$$

where $\delta = 0$ if there are no optimality cuts and $\delta = 1$ if there is at least one optimality cut. In this case $\delta = 0$, because there are no optimality cuts so far. Given a solution $x = x^0$, to the Master Problem, we solve the subproblem

$$\begin{aligned} f^T y &= w \text{ (min)} \\ Dy &= d - Bx^0 \\ y &\geq 0. \end{aligned} \tag{10.125}$$

The solution of this is then used to define cuts for the Master Problem.

The solution to

$$\begin{aligned} & \text{Minimize} && x_1 + x_2 + x_3 = z \\ & \text{subject to} && x_1 + 2x_2 + 3x_3 = 6 \\ & && 3x_1 + 2x_2 + x_3 = 6 \\ & && x \geq 0 \end{aligned}$$

is $x = x^0 = (1.5, 0, 1.5)^T$. Using this we first compute the right-hand side to the subproblem (10.125) as

$$d - Bx^0 = \begin{pmatrix} 9 \\ 15 \\ 9 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 1 \\ 3 & 2 & 1 \\ 4 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1.5 \\ 0.0 \\ 1.5 \end{pmatrix} = \begin{pmatrix} 6.0 \\ 9.0 \\ 1.5 \end{pmatrix},$$

and then solve the sub problem:

$$\begin{aligned} & \text{Minimize} && 3y_1 + 2y_2 + y_3 = w \\ & \text{subject to} && 4y_1 - y_2 + y_3 = 6.0 \\ & && 3y_1 + 2y_2 + y_3 = 9.0 \\ & && y_1 + y_2 + y_3 = 1.5 \\ & && y \geq 0. \end{aligned}$$

This problem is infeasible and so we use its infeasibility multipliers to create an infeasibility cut. The infeasibility multipliers are:

$$\pi^1 = \begin{pmatrix} -0.2 \\ 1.0 \\ -2.2 \end{pmatrix}$$

Next we compute the infeasibility cut $G^1x \geq g^1$ by computing

$$G^1 = (\pi^1)^T B = (-6 \quad 4 \quad -1.4)^T$$

and

$$g^1 = (\pi^1)^T d = -6.6$$

The new Benders Restricted Master is:

$$\begin{aligned} & \text{Minimize} && x_1 + x_2 + x_3 + \delta\theta = z \\ & \text{subject to} && x_1 + 2x_2 + 3x_3 = 6 \\ & && 3x_1 + 2x_2 + x_3 = 6 \\ & && -6x_1 + 4x_2 - 1.4x_3 \geq -6.6 \\ & && x \geq 0, \end{aligned}$$

where $\delta = 1$ because there still is no optimality cut. The optimal solution to this is $z^1 = 3.0$, $x^1 = (1.207792 \quad 0.584416 \quad 1.207792)$. Using this we first compute the right-hand side to the new subproblem as

$$d - Bx^1 = \begin{pmatrix} 9 \\ 15 \\ 9 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 1 \\ 3 & 2 & 1 \\ 4 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1.207792 \\ 0.584416 \\ 1.207792 \end{pmatrix} = \begin{pmatrix} 6.0 \\ 9.0 \\ 3.545456 \end{pmatrix}$$

and then solve the subproblem:

$$\begin{aligned} & \text{Minimize} && 3y_1 + 2y_2 + y_3 = w \\ & \text{subject to} && 4y_1 - y_2 + y_3 = 6.0 \\ & && 3y_1 + 2y_2 + y_3 = 9.0 \\ & && y_1 + y_2 + y_3 = 3.545456 \\ & && y \geq 0. \end{aligned}$$

This subproblem solves to optimality: $w = w^* = 9.0$, $y = y^* = (1.9091 \quad 1.6364 \quad 0.0)^T$ with optimal multipliers

$$\pi^2 = \begin{pmatrix} 0.0 \\ 1.0 \\ 0.0 \end{pmatrix}.$$

Next we compute the optimality cut $G^2x + \theta \geq g^2$ by computing

$$G^2 = (\pi^2)^T B = (3.0 \quad 2.0 \quad 1.0)$$

and

$$g^2 = (\pi^2)^T d = 15.$$

The new Benders Restricted Master is:

$$\begin{array}{llll} \text{Minimize} & x_1 + x_2 + x_3 + \theta = z \\ \text{subject to} & x_1 + 2x_2 + 3x_3 & = & 6 \\ & 3x_1 + 2x_2 + x_3 & = & 6 \\ & -6x_1 + 4x_2 - 1.4x_3 & \geq & -6.6 \\ & 3x_1 + 2x_2 + x_3 + \theta & \geq & 15 \\ & x & \geq & 0. \end{array}$$

The optimal solution to this is $z^2 = 12.0$, $x^2 = (1.207792 \quad 0.584416 \quad 1.207792)$, $\theta^2 = 9$. Since $\theta^2 = w^* = 9$ we are optimal and we terminate.

- ▷ **Exercise 10.35** Solve (10.123) to verify that the solution obtained by Benders decomposition is correct.
- ▷ **Exercise 10.36** Write down an outline of an algorithm to solve the linear program (10.106) by Benders decomposition.
- ▷ **Exercise 10.37** If at iteration k of the Restricted Master the optimal value of θ^k is equal to the sum of the optimal objective values of the subproblems, prove that we have found an optimal solution to the linear program.
- ▷ **Exercise 10.38** Show that the Benders Decomposition algorithm terminates after a finite number of steps with one of the following: a feasible optimal solution, an indication that there exists no feasible solution, or an unbounded solution consisting of a feasible solution plus a feasible homogeneous solution.

As we keep adding cuts and solving the Benders Restricted Master augmented by the new cuts, some of the inequalities may no longer remain tight at an optimal solution of the restricted master. Then to reduce the computational effort or not exceed computer memory, some or all the inactive inequalities may be discarded because they can always be regenerated as and when needed. However, practical implementations have shown that discarding all such inactive inequalities typically results in many more iterations than if some inactive inequalities were retained, as a part of the restricted master. See Section 10.2.2.2 for some ideas of criteria for deciding which inactive inequalities to retain.

10.4 BLOCK-ANGULAR SYSTEM

In this section we shall show how to use the D-W decomposition principle to solve the *block-angular* problem (10.126), namely:

$$\begin{aligned}
 \text{Minimize} \quad & (c^0)^T x^0 + (c^1)^T x^1 + \cdots + (c^K)^T x^K = z \\
 \text{subject to} \quad & A^0 x^0 + A^1 x^1 + \cdots + A^K x^K = b \\
 & F^1 x^1 = f^1 \\
 & \quad \quad \quad \ddots \\
 & \quad \quad \quad F^K x^K = f^K \\
 & x^0 \geq 0, x^1 \geq 0, \dots, x^K \geq 0.
 \end{aligned} \tag{10.126}$$

For our discussion it will be convenient to think of problem (10.126) as solving

$$\begin{aligned}
 \text{Minimize} \quad & (c^0)^T x^0 + (c^1)^T x^1 + \cdots + (c^K)^T x^K = z \\
 \text{subject to} \quad & A^0 x^0 + A^1 x^1 + \cdots + A^K x^K = b \\
 & x^0 \geq 0
 \end{aligned} \tag{10.127}$$

subject to the additional constraints:

$$(\mathcal{S}_k): \quad F^k x^k = f^k, \quad x^k \geq 0, \quad \text{for } k = 1, \dots, K. \tag{10.128}$$

Assume for the moment that all the basic feasible solutions and all the extreme homogeneous solutions for \mathcal{S}_1 to \mathcal{S}_K for (10.128) are available. In practice, of course, these are usually too numerous to be all at hand. When this is the case, our goal will be to show how to generate just those solutions among them that are needed.

From the Resolution Theorem 10.5 any solution $x^k \geq 0$ to (\mathcal{S}_k) for $k = 1, \dots, K$ can be written in the form

$$x^k = \sum_{i=1}^{L_k} \alpha_{ki} u^{ki} + \sum_{j=1}^{M_k} \beta_{kj} v^{kj}, \tag{10.129}$$

where

$$\sum_{i=1}^{L_k} \alpha_{ki} = 1, \quad \alpha_{ki} \geq 0, \quad i = 1, \dots, L_k, \tag{10.130}$$

$$\beta_{kj} \geq 0, \quad j = 1, \dots, M_k, \tag{10.131}$$

and where u^{ki} for $i = 1, \dots, L_k$ are the full finite set of basic feasible solutions for \mathcal{S}_k and v^{kj} for $j = 1, \dots, M_k$ are the full finite set of normalized extreme homogeneous solutions for \mathcal{S}_k . Conversely any solution represented by (10.129) is feasible for \mathcal{S}_k .

▷ **Exercise 10.39** Rewrite problem (10.126) in terms of α_{ki} and β_{kj} .

Denote the linear transforms of G^{ki} of u^{ki} and H^{kj} of v^{kj} by:

$$\begin{aligned} G^{ki} &= A^k u^{ki}, \\ H^{kj} &= A^k v^{kj}, \end{aligned} \quad (10.132)$$

and denote the associated scalar costs by:

$$\begin{aligned} g_{ki} &= (c^k)^T u^{ki}, \\ h_{kj} &= (c^k)^T v^{kj}. \end{aligned} \quad (10.133)$$

- ▷ **Exercise 10.40** Write down the original linear program in terms of the transforms defined by (10.132) and (10.133). *Note:* The resulting linear program is called the Full Master Program.

Assume for the following exercises that we have an initial basic feasible solution to a Restricted Master Program and let the simplex multipliers associated with the $m + K$ rows of the restricted master program be $\pi, \gamma_1, \gamma_2, \dots, \gamma_K$, where π is the m -vector of multipliers associated with the first m constraints of the Restricted Master Program and $\gamma_k, k = 1, \dots, K$, are the scalar multipliers associated with the K convexity constraints (10.130) of the Restricted Master Program.

- ▷ **Exercise 10.41** Show how to compute the simplex multipliers and how to determine the reduced costs.
- ▷ **Exercise 10.42** Show that in order to obtain the lowest reduced cost, we must solve the subproblems:

$$\begin{aligned} \text{Minimize} \quad & (\bar{\rho}^k)^T x^k = z_k \\ \text{subject to} \quad & F^k x^k = f^k \\ & x^k \geq 0, \end{aligned} \quad (10.134)$$

where $\bar{\rho}^k$, the Adjusted Costs, satisfy

$$\bar{\rho}^k = c^k - (A^k)^T \pi \quad (10.135)$$

for $k = 1, \dots, K$.

If, at some iteration, basic feasible solutions are obtained to the subprograms such that all the reduced costs for the full master program are nonnegative, we are at an optimal solution of the full master program, and (10.129) can be used to compute an optimal solution to (10.126). Otherwise we bring a new column into the basis for the master program.

- ▷ **Exercise 10.43** Specify the conditions under which we bring in G^{k*} or H^{k*} for $k = 1, \dots, K$ into the basis of the restricted master problem.
- ▷ **Exercise 10.44** Analogous to Algorithm 10.1, write down the steps of the Dantzig-Wolfe algorithm for solving a linear program in the Block-Angular form.

- ▷ **Exercise 10.45** State and prove a Theorem analogous to Theorem 10.9 for the Block Angular system. Also state and prove an analogous Corollary 10.10 for the Block Angular system.
- ▷ **Exercise 10.46** State and prove a Lemma analogous to Lemma 10.12.
- ▷ **Exercise 10.47** Consider the following two stage block-angular problem which for convenience has been written in a form suitable for decomposing into one master problem, consisting of one equation and one convexity constraint, and two subprograms.

$$\begin{aligned}
 \text{Minimize} \quad & x_1^1 - x_1^1 - 3x_3^1 + 3x_4^1 + 20x_1^2 + 30x_2^2 + 7x_3^2 + x_4^2 + x_1^1 = z \\
 \text{subject to} \quad & 3x_1^1 - x_2^1 - 3x_3^1 + 2x_4^1 + x_1^2 + 2x_2^2 + 0x_3^2 - x_4^2 + x_5^2 = 1 \\
 & x_1^1 + x_2^1 - x_3^1 + x_4^1 = 3 \\
 & 0x_1^1 + x_2^1 - x_3^1 - x_4^1 = 4 \\
 & x_1^2 + x_2^2 + x_3^2 + x_4^2 + 0x_5^2 = 1 \\
 & 2x_1^2 + x_2^2 - 2x_3^2 + 0x_4^2 + x_5^2 = 2 \\
 \text{and } x_i^1 \geq 0, & \quad i = 1, \dots, 4 \text{ and } x_j^2 \geq 0, j = 1, \dots, 5.
 \end{aligned}$$

Solve this problem using the D-W decomposition principle.

- ▷ **Exercise 10.48** For the Block-Angular system show how to obtain an initial feasible solution by a Phase I procedure.
- ▷ **Exercise 10.49** State and prove a theorem analogous to Theorem 10.13 for the Block Angular system.
- ▷ **Exercise 10.50** Can the coefficients of the objective be modified in such a way that a class of feasible solutions exist such that z is unbounded below? If yes, apply the Benders Decomposition Algorithm to the modified problem.

10.5 STAIRCASE STRUCTURED PROBLEMS

A staircase linear program has a square partitioned structure consisting of $K \times K$ submatrices all of whose elements are zero except possibly the elements of the submatrices on and just below the main diagonal; for example

$$\begin{aligned}
 \text{Minimize} \quad & (c^1)^T x^1 + (c^2)^T x^2 + (c^3)^T x^3 + (c^4)^T x^4 = z \\
 \text{subject to} \quad & A^{11} x^1 = b^1 \\
 & A^{21} x^1 + A^{22} x^2 = b^2 \\
 & A^{32} x^2 + A^{33} x^3 = b^3 \\
 & A^{43} x^3 + A^{44} x^4 = b^4 \\
 & x^k \geq 0, \quad k = 1, \dots, K, \text{ where } K = 4.
 \end{aligned} \tag{10.136}$$

We will call the successive *steps* from the top of the *stairs* down as the time periods $t = 1, t = 2, t = 3$, and $t = 4$, although in certain applications the steps may be stages of a production process or partitions of a physical structure. The methods to be described are quite general and can be used to solve any K -step problem. In this section we sketch three ways that either the D-W or Benders Decomposition Principle can be applied recursively using a *nested decomposition* approach.

10.5.1 USING BENDERS DECOMPOSITION

One way to nest the partitions of x is *forward-in-time* starting with the variables x into $\{x^1\}$, $\{x^2, x^3, x^4\}$ which results in a Benders subproblem to be solved for some fixed x^1 :

$$\begin{aligned} \text{Minimize} \quad & (c^2)^T x^2 + (c^3)^T x^3 + (c^4)^T x^4 = \gamma_1 \\ \text{subject to} \quad & A^{22} x^2 = b^2 - A^{21} x^1 \\ & A^{32} x^2 + A^{33} x^3 = b^3 \\ & A^{43} x^3 + A^{44} x^4 = b^4 \\ & x^k \geq 0, \quad k = 2, 3, 4. \end{aligned} \tag{10.137}$$

This subproblem is solved by nesting the partition corresponding $\{x^2, x^3, x^4\}$ into $\{x^2\}$, $\{x^3, x^4\}$ and so forth. Each nesting recursively decreases the number of steps until the remaining subproblem has only one step.

The Benders Restricted Master corresponding to the first partition has a form similar to (10.118) that, in vector notation, is

$$\begin{aligned} (c^1)^T x^1 + \gamma_1 &= z \text{ (Min)} \\ G^T x^1 + e \gamma_1 &\geq g, \quad e = (1, 1, \dots, 1)^T, \\ H^T x^1 &\geq h, \\ x^1 &\geq 0. \end{aligned} \tag{10.138}$$

where $G = (G^1, G^2, \dots)$, $g^T = (g_1, g_2, \dots)$, $H = (H^1, H^2, \dots)$, and $h^T = (h_1, h_2, \dots)$. The Restricted Masters for the successive subproblems are defined in an analogous way.

▷ **Exercise 10.51** Complete the description of the Benders Decomposition approach for $K = 4$. Specify all the Restricted Master and Sub Problems. Write down all the steps to solve the entire problem.

▷ **Exercise 10.52** Show (using Benders Decomposition in a forward direction in time as just described) that the optimal first-period activity levels are determined by prices on items produced in the first period for use in the second and subsequent periods. Show, however, that these prices are not unique and therefore multiple cuts will be needed to determine the optimal first-period activity levels.

▷ **Exercise 10.53** Consider the following staircase system:

$$\begin{aligned}
 (\bar{c}^1)^T \bar{x}^1 + (c^1)^T x^1 + (\bar{c}^2)^T \bar{x}^2 + (c^2)^T x^2 + (\bar{c}^3)^T \bar{x}^3 + (c^3)^T x^3 + (c^4)^T x^4 &= z \text{ (Min)} \\
 \bar{A}^{11} \bar{x}^1 + A^{11} x^1 &= b^1 \\
 A^{21} x^1 + \bar{A}^{22} \bar{x}^2 + A^{22} x^2 &= b^2 \\
 A^{32} x^2 + \bar{A}^{33} \bar{x}^3 + A^{33} x^3 &= b^3 \\
 A^{43} x^3 + A^{44} x^4 &= b^4
 \end{aligned}$$

$$x^k \geq 0, \quad k = 1, \dots, 4, \quad \bar{x}^j \geq 0, \quad j = 1, \dots, 3.$$

1. Show that the *structure* when viewed forward-in-time is identical to the structure when viewed backward in time, i.e., relabeling the indices (1, 2, 3, 4) to (4, 3, 2, 1). The variables x^1 link the first and second periods, the variables x^2 link the second and third periods, and the variables x^3 link the third and fourth periods. Assume that the linking variables have very few components (for example, 1 or 2), while \bar{x}^1 , \bar{x}^2 , \bar{x}^3 have many components. Show how this information can be used to develop a more efficient algorithm.
2. Suppose that $(\bar{x}^1, \bar{x}^2, \bar{x}^3)$ is each a scalar variable. Apply a backward-in-time dynamic-programming recursion approach to effectively solve the problem.

▷ **Exercise 10.54** Apply Benders Decomposition to (10.136) by the backward-in-time partitioning x into $\{x^1, x^2, x^3\}$, $\{x^4\}$ to form the first nested subproblem

$$\begin{aligned}
 \text{Minimize} \quad & (c^4)^T x^4 = \gamma \\
 \text{subject to} \quad & A^{44} x^4 = b^4 - A^{43} x^3 \\
 & x^4 \geq 0.
 \end{aligned} \tag{10.139}$$

Then partition $\{x^1, x^2, x^3\}$ into $\{x^1, x^2\}$, $\{x^3\}$ and finally partition $\{x^1, x^2\}$ into $\{x^1\}$, $\{x^2\}$ to recursively nest. Compare this way to doing the nesting with the forward-in-time nesting way discussed at the start of this section. Why is the forward-in-time way to be preferred?

▷ **Exercise 10.55** Apply Benders Decomposition to (10.136) by partitioning x into the sets $\{x^1, x^2\}$, $\{x^3, x^4\}$ to form a Restricted Master corresponding to $\{x^3, x^4\}$ and sub corresponding to $\{x^1, x^2\}$. Next partition $\{x^1, x^2\}$ into $\{x^1\}$, $\{x^2\}$ to form a Restricted Master with $\{x^2\}$ and a sub with $\{x^1\}$. Compare this with the approach in Exercise 10.54.

10.5.2 USING D-W DECOMPOSITION

Using D-W Decomposition, we partition the rows into time steps $\{t = 1\}$, $\{t=2, t=3, t=4\}$, then we solve

$$\begin{aligned}
 \text{Minimize} \quad & (c^1)^T x^1 + (c^2)^T x^2 + (c^3)^T x^3 + (c^4)^T x^4 = z \\
 \text{subject to} \quad & A^{11} x^1 = b^1 \\
 & x^k \geq 0, \quad k = 1, \dots, 4.
 \end{aligned} \tag{10.140}$$

subject to the additional constraints

$$\begin{aligned} A^{21}x^1 + A^{22}x^2 &= b^2 \\ A^{32}x^2 + A^{33}x^3 &= b^3 \\ A^{43}x^3 + A^{44}x^4 &= b^4 \\ x^k &\geq 0, \quad k = 1, \dots, 4. \end{aligned} \quad (10.141)$$

Then using the Resolution Theorem 10.5, any solution x of (10.141) can be written as:

$$x = \begin{pmatrix} x^1 \\ x^2 \\ x^3 \\ x^4 \end{pmatrix} = \sum_{i=1}^L \alpha_i \begin{pmatrix} u^{i1} \\ u^{i2} \\ u^{i3} \\ u^{i4} \end{pmatrix} + \sum_{j=1}^M \beta_j \begin{pmatrix} v^{j1} \\ v^{j2} \\ v^{j3} \\ v^{j4} \end{pmatrix} \quad (10.142)$$

where

$$\sum_{i=1}^{L_1} \alpha_i = 1, \quad \alpha_i \geq 0, \quad i = 1, \dots, L, \quad \beta_j \geq 0, \quad j = 1, \dots, M, \quad (10.143)$$

and where $u^i = (u^{i1}, u^{i2}, u^{i3}, u^{i4})$ for $i = 1, \dots, L$ are the full finite set of basic feasible solutions for (10.141) and $v^j = (v^{j1}, v^{j2}, v^{j3}, v^{j4})$ for $j = 1, \dots, M$ are the full finite set of normalized extreme homogeneous solutions for (10.141). Then the Full Master Program is:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i=1}^L [(c^1)^T u^{i1} + (c^2)^T u^{i2} + (c^3)^T u^{i3} + (c^4)^T u^{i4}] \alpha_i \\ & + \sum_{j=1}^M [(c^1)^T v^{j1} + (c^2)^T v^{j2} + (c^3)^T v^{j3} + (c^4)^T v^{j4}] \beta_j = z \\ \text{subject to} \quad & \sum_{i=1}^L A^{11} u^{i1} \alpha_i + \sum_{j=1}^M A^{11} v^{j1} \beta_j = b^1 \\ & \sum_{i=1}^L \alpha_i = 1 \\ & \alpha_i \geq 0, \quad i = 1, \dots, L, \quad \beta_j \geq 0, \quad j = 1, \dots, M. \end{aligned} \quad (10.144)$$

Because it is impractical in general to generate the Full Master Program, we instead use the optimal prices π^1 of the Restricted Master Program to determine the objective for subproblem (10.141). The subproblem with the adjusted objective is then solved to determine the next incoming column:

$$\begin{aligned} \text{Minimize} \quad & (c^1 - (\pi^1)^T A^{11})^T x^1 + (c^2)^T x^2 + (c^3)^T x^3 + (c^4)^T x^4 = z \\ \text{subject to} \quad & A^{21}x^1 + A^{22}x^2 = b^2 \\ & A^{32}x^2 + A^{33}x^3 = b^3 \\ & A^{43}x^3 + A^{44}x^4 = b^4 \\ & x^k \geq 0, \quad k = 1, \dots, 4. \end{aligned} \quad (10.145)$$

This subproblem is now in a form similar to the original problem that can be decomposed by partitioning the rows into time steps $\{t = 2\}$, $\{t = 3, t = 4\}$, and so forth.

▷ **Exercise 10.56** Complete the description of the D-W Decomposition approach. Specify all the Restricted Master Problems and corresponding subproblems. Write down all the steps to solve the entire problem.

▷ **Exercise 10.57** Apply D-W Decomposition to (10.136) by partitioning the rows into time steps $\{t = 1, t = 2, t = 3\}$, $\{t = 4\}$ with $\{t = 4\}$ used to form the subproblem. Then partition $\{t = 1, t = 2, t = 3\}$ into $\{t = 1, t = 2\}$, $\{t = 3\}$ with $\{t = 3\}$ used to form the subproblem, and finally partition $\{t = 1, t = 2\}$ into $\{t = 1\}$, $\{t = 2\}$ with $\{t = 2\}$ used to form the subproblem. Compare the two ways to do the partitioning and discuss why the first way is preferred.

10.5.3 USING D-W DECOMPOSITION WITH ALTERNATE STAGES FORMING THE SUBPROBLEMS

We start the decomposition process by making a subprogram of every other stage. We arbitrarily let the master problem correspond to the second and fourth stages of (10.136) to form the master program and let the first and third stages form the subprograms. That is we wish to solve

$$\begin{aligned} \text{Minimize} \quad & (c^1)^T x^1 + (c^2)^T x^2 + (c^3)^T x^3 + (c^4)^T x^4 = z \\ \text{subject to} \quad & A^{21} x^1 + A^{22} x^2 = b^2 \\ & A^{43} x^3 + A^{44} x^4 = b^4 \\ & x^4 \geq 0. \end{aligned} \tag{10.146}$$

subject to the additional two *independent* sets of constraints

$$(\mathcal{S}_1) : \quad A^{11} x^1 = b^1, \quad x^1 \geq 0 \tag{10.147}$$

and

$$(\mathcal{S}_2) : \quad A^{32} x^2 + A^{33} x^3 = b^3, \quad x^2 \geq 0, \quad x^3 \geq 0 \tag{10.148}$$

Assuming that there are no homogeneous solutions, substituting the convex combination of extreme points x^{1i} , $i = 1, \dots, M_1$ of (10.147) and x^{2j} , $j = 1, \dots, M_2$

(10.148) into (10.146), and letting $(c^j)^T = d^j$, we generate the Master problem

$$\begin{aligned}
 &\text{Find Min } z, \lambda_i \geq 0, \mu_j \geq 0, \text{ such that} \\
 &\sum_i (d^1 x^{1i}) \lambda_i + \sum_j (d^2 x^{2j} + d^3 x^{3j}) \mu_j + d^4 x^4 = z(\text{Min}) \\
 &\sum_i (A^{21} x^{1i}) \lambda_i + \sum_j (A^{22} x^{2j}) \mu_j = b^1 \\
 &\sum_i \lambda_i = 1 \\
 &\sum_j (A^{23} x^{2j}) \mu_j + A^{44} x^4 = b^2 \\
 &\sum_j \mu_j = 1
 \end{aligned} \tag{10.149}$$

Note that the Master Problem is again a staircase problem with half the number of steps. In general, we can partition the steps of a staircase problem into two sets of equations, making one set the Sub and the equations of the resulting Master corresponding to the other set plus one convexity constraint. If the Sub consists of, for example, the subset of even steps, then equations of each even step will be independent of those of any other even step, and each will give rise to an independent convexity constraint in the Master; see (10.149).

▷ **Exercise 10.58** Extend the theory to include homogeneous solutions.

THEOREM 10.18 (Decomposition of a $2K$ -Stage Problem) *Given a $2K$ -step staircase problem, each step consisting of m equations. Making the set of K even steps, the subproblem will give rise to a Master Problem equivalent to the original problem that is a staircase problem of K steps, each step consisting of $m + 1$ equations, one of which is a convexity constraint.*

▷ **Exercise 10.59** Prove Theorem 10.18. Restate and prove Theorem 10.18 for the case when the staircase problem has an odd number of steps.

▷ **Exercise 10.60** Provide details for the nested decomposition approach using alternate stages to form the subproblems.

10.6 DECOMPOSITION USED IN CENTRAL PLANNING

The theory developed for decomposition makes it possible to plan the overall operation of an organization without the central office staff having any detailed knowledge

With these it can set up a Restricted Master Program:

$$\begin{aligned}
 (c^o)^T x^o + z_{11}\alpha_{11} + z_{12}\alpha_{12} + \cdots + z_{K1}\alpha_{K1} + z_{K2}\alpha_{K2} &= z \quad (\text{Min}) \\
 A^o x^o + G^{11}\alpha_{11} + G^{12}\alpha_{12} + \cdots + G^{K1}\alpha_{K1} + G^{K2}\alpha_{K2} &= b \\
 \alpha_{11} + \alpha_{12} &= 1 \\
 &\vdots \\
 \alpha_{K1} + \alpha_{K2} &= 1 \\
 \alpha_{k1} \geq 0, \alpha_{k2} \geq 0, k = 1, \dots, K.
 \end{aligned} \tag{10.154}$$

We assume, to simplify the discussion, this Restricted Master problem is feasible. Its optimal solution $\alpha_{k1} = \bar{\alpha}_{k1}, \alpha_{k2} = \bar{\alpha}_{k2}$ provides us with allocation vectors $\bar{G}^k = \bar{\alpha}_{k1}G^{k1} + \bar{\alpha}_{k2}G^{k2}$ for $k = 1, \dots, K$. Let $\pi = \bar{\pi}, \gamma_k = \bar{\gamma}_k$, for $k = 1, \dots, K$ be the optimal prices associated with the Restricted Master Problem. We, the Central Office, now wish to use these prices to determine if these \bar{G}^k are the optimal allocation vectors, and, if not, how they can be improved. To this end we instruct the plants k to solve their own detailed program by tentatively assuming that $\pi = \bar{\pi}$ are the prices for the scarce resources.

$$\begin{aligned}
 \text{Minimize} \quad & (c^k - (A^k)^T \bar{\pi})^T x^k = \theta_k \\
 \text{subject to} \quad & F^k x^k = f^k \\
 & x^k \geq 0.
 \end{aligned} \tag{10.155}$$

A sufficient test for our tentative allocation to be optimal is, according to Theorem 10.9 on Page 290,

$$\theta_k = \bar{\gamma}_k \quad \text{for } k = 1, \dots, K, \tag{10.156}$$

where $\bar{\gamma}_k$ are the optimal multipliers on the convexity constraints of (10.154).

For those k that fail the test, the Central Office requires the plant to determine the costs z_k of their tentative plan $x^k = \bar{x}^k$ and its corresponding use G^k of scarce resources:

$$\begin{aligned}
 z_{k3} &= (c^k)^T \bar{x}^k \\
 G^{k3} &= A^k \bar{x}^k
 \end{aligned} \tag{10.157}$$

which the Central Office uses as additional columns in the Restricted Master Program with weights α_{k3} . The iterative process is repeated until the optimality test is passed.

10.7 NOTES & SELECTED BIBLIOGRAPHY

The generalized linear program of Section 10.1 was first developed in the joint work of Philip Wolfe and George Dantzig on a decomposition principle for large-scale block-angular programs (discussed in Section 10.2), the origin of which is discussed in the next paragraph. Wolfe suggested that the procedure there could be viewed as a special case of the generalized linear program discussed in Section 10.1.

Kuhn & Tucker [1950] considered a broad class of nonlinear programming problems whose objective function is a general convex function and the constraints are of the form

$f_i(x) \leq 0$, where the $f_i(x)$ were convex functions; Dantzig [1963] proved that this general class of nonlinear problems is a special case of Wolfe's Generalized Program.

For details on the Resolution Theorem 10.5 and properties of convex polyhedral sets, upon which the Decomposition Principle and Generalized Programming are based, see Goldman [1956], Goldman & Tucker [1956a, 1956b], and Hoffman & Hirsch [1961].

The decomposition principle for linear programs was first developed by Dantzig & Wolfe [1960, 1961]; the two papers present two different ways of looking at the decomposition principle. Historically, the special case (10.2) gave rise to the more general concept of decomposition as applied to a generalized linear program (Dantzig & Wolfe [1960, 1961]). According to Dantzig [1963], the decomposition approach was inspired by the proposals of Ford & Fulkerson [1958b] for solving multistage commodity network problems. Jewell [1958] also used similar approaches to that of Ford & Fulkerson. Later Benders [1962] developed an approach that when applied to the dual was the same as the Dantzig-Wolfe approach applied to the primal problem. Benders (dual) decomposition has been used extensively to solve stochastic programs; it is the method of choice for solving linear programs under uncertainty (see Chapter 12).

The decomposition principle stirred up a lot of interest at first, but interest waned when it was observed in practice, that, while the method generated good approximations in a reasonable amount of iterates, it was slow to converge to a very close approximation to the optimal. The convergence rate increased dramatically once the initial software was replaced by software written by skilled numerical analysts. For a discussion of the behavior of decomposition-based algorithms see Adler & Ülkücü [1973], Beale, Hughes, & Small [1965], Bradley, Hax, & Magnanti [1977], and Ho [1984].

Modern implementations of the Simplex Algorithm take advantage of sparsity to efficiently solve large-scale problems. For even larger systems, refinements of a decomposition algorithm together with a very good sparse representation are promising. For an advanced implementation of the Dantzig-Wolfe decomposition approach, see, for example, see Ho & Loute [1981]. See also Entriken [1989] for decomposition of linear programs using parallel computing.

For staircase structured problems it is most common to use a decomposition approach recursively as a nested-decomposition approach. This approach was first suggested by Dantzig [1963]. Since then there has been work by a considerable number of authors. For discussions of the nested decomposition approach see, for example, Dantzig [1963], Dantzig, Dempster, & Kallio [1981], Glassey [1971], Ho [1974], and Ho & Manne [1974]. Such a primal nested decomposition approach has been applied to large-scale modeling problems in the European Common Market by Ho & Loute [1981]. See also Bisschop & Meeraus [1981], Dantzig & Perold [1978], Fourer [1982, 1983a, 1984], and Nishiya [1983].

A linear program solved by decomposition has very different numerical characteristics than if solved without decomposition. The numerical properties of the decomposition process are not yet fully understood. Some characteristics that have been observed are:

1. A linear program can be well-scaled as initially formulated but can become very badly scaled after the decomposition principle is applied.
2. In spite of possible ill-conditioning of a decomposed problem, it usually turns out that its "optimal solution" is "close" to the true optimal solution.
3. In most practical implementations the basic feasible solutions and extreme homogeneous solutions of the subprograms are not stored; instead, G^i and H^j , the products of a matrix times these solutions, are used. Thus, at the end of the algorithm, the so-

lutions to the original linear program have to be reconstructed by solving additional systems of equations. This can lead to numerical errors.

For further details and examples on the numerical behavior of decomposition algorithms see Nazareth [1984, 1987]. For convergence of decomposition algorithms see Ho [1984].

10.8 PROBLEMS

- 10.1 *Dantzig [1963]*. The coordinator, “Staff,” of the Central Agency must procure tankers to assist his distributor, “Sub,” in the shipping of their product from two plants to four terminals. Sub has the following transportation cost/availability/requirement array

	3	6	6	5	9
	8	1	3	6	8
2	7	3	5		

where the right most column contains the availability at each plant, the last row represents the requirements at each terminal, and the remaining entries are the costs of shipping from each plant to each terminal. The shipments from plant 1 to terminal 3 and from plant 2 to terminal 2 are made via tankers, with each unit of product requiring two tankers. All other shipments are made via pipeline. Staff is not interested in the Subs details but does know that there are nine tankers available for use.

- Solve Sub’s transportation problem.
 - Re-solve Sub’s transportation problem after putting in an arbitrarily high cost for the use of tankers.
 - Set up Staff’s problem (restricted master) from the two solutions obtained from Sub.
 - Solve the entire problem using the D-W decomposition algorithm. At each iteration obtain an estimate of a lower bound on the objective function.
- 10.2 Show that the feasible solutions generated by the Dantzig-Wolfe decomposition algorithm can lie in the interior of the original linear program.
- 10.3 The Dantzig-Wolfe method yields the optimal multipliers which are then used to generate the primal variables. Show how to apply the Dantzig-Wolfe algorithm to the dual of a problem in order to generate the primal variables directly and the multipliers indirectly.
- 10.4 Consider a network with M source-terminal pairs (s_i, t_i) and let the flow value between s_i and t_i be F_i for $i = 1, \dots, M$; that is, the flow between each pair (s_i, t_i) can be thought of as the flow of a different commodity. Suppose that each arc (i, j) of the network has arc capacity h_{ij} which is an upper bound on the total flow of all commodities on the directed arc (i, j) . Let c_{ij} be the cost

per unit total flow on arc (i, j) . Assuming that all flows can be positive on directed arcs, the goal is to find a minimum-cost feasible flow.

1. Formulate this problem.
2. If this Minimum-Cost Multi-Commodity Flow is to be solved by the D-W Decomposition procedure, what are the subproblems?

10.5 *Ph.D. Comprehensive Exam, March 30, 1970, at Stanford.* Let x , y , and z be unknown vectors; A^1 , A^2 , B^1 , B^2 , B^3 known matrices; b^1 , b^2 , b^3 , and c known column vectors. Suppose that a Dantzig-Wolfe decomposition model has two subproblems.

$$\begin{aligned} \mathcal{L}_1 &= \{x \mid A^1x = b^1; x \geq 0\}, \\ \mathcal{L}_2 &= \{y \mid A^2y = b^2; y \geq 0\}. \end{aligned} \tag{10.158}$$

Suppose further that the master problem is written in the form:

$$\begin{aligned} &\text{Maximize} && c^Tz \\ &\text{subject to} && B^1x + B^2y + B^3z = b^3 \\ & && x \in \mathcal{L}_1 \\ & && y \in \mathcal{L}_2 \\ & && z \geq 0. \end{aligned} \tag{10.159}$$

- (a) Formulate both the master and the subproblems as linear programs.
- (b) Are \mathcal{L}_1 and \mathcal{L}_2 defined by (10.158) convex sets?
- (c) How does your formulation of the Master Problem handle the case in which either \mathcal{L}_1 or \mathcal{L}_2 is an unbounded set?
- (d) Suppose that both \mathcal{L}_1 and \mathcal{L}_2 are bounded, and that we have one basic feasible solution to the master problem. Applying just one simplex solution to each of the subproblems, how may we calculate both a lower and an upper bound on the maximand of (10.159)?
- (e) Suppose that both \mathcal{L}_1 and \mathcal{L}_2 are bounded and that we wish to solve the following problem:

$$\begin{aligned} &\text{Maximize} && c^Tz \\ &\text{subject to} && B^1x + B^2y + B^3z = b^3 \\ & && x \text{ is an extreme point of } \mathcal{L}_1 \\ & && y \text{ is an extreme point of } \mathcal{L}_2 \\ & && z \geq 0 \end{aligned} \tag{10.160}$$

Now suppose that we have specified arbitrarily one extreme point in \mathcal{L}_1 , another in \mathcal{L}_2 , and that we have then solved the linear program for z . How may the information from this simplex solution be used to calculate both a lower and an upper bound on the maximand of (10.160)?

- (f) Show that (10.160) is a mixed-integer programming problem; i.e., where some of the variables are forced to be integers.

10.6 *Ph.D. Comprehensive Exam, September 24, 1977, at Stanford.* Assume that a linear program of the form

$$\begin{aligned} & c^1x^1 + c^2x^2 + c^3x^3 = z \text{ (min)} \\ \text{I : } & A^1x^1 + A^2x^2 = b^1 \\ \text{II : } & \bar{A}^2x^2 + A^3x^3 = b^2 \end{aligned} \quad (x^1, x^2, x^3) \geq 0$$

is solved by the decomposition principle. Equations II constitute the subproblem and the “master” equations correspond to Equations I. There are three questions to be answered: (a), (b1), (b2).

- (a) Show that the optimal prices to the master problem are also the optimal prices $\pi^1 = \hat{\pi}^1, \pi^2 = \hat{\pi}^2$ associated with the original problem I and II and also the optimal values $x^1 = \hat{x}^1$ but that optimal values for x^2 and x^3 are not available nor can they be reconstructed from the solution of the master problem but can be reconstructed if complete records are kept of the extreme solution to the subproblem of the t th cycle, i.e., the vectors $(x^2, x^3) = (x^{2,t}, x^{3,t})$ for $t = 1, 2, \dots$, and not just their linear transforms used to form the master.
- (b) Using the known optimal value of $x^1 = \hat{x}^1$ from the solution of the master problem, suppose we now solve

$$\begin{aligned}
 \text{I}' : \quad & A^2 x^2 & = & b^1 - A^1 \hat{x}^1 \\
 & & & (x^2, x^3) \geq 0 \\
 \text{II}' : \quad & \bar{A}^2 x^2 + A^3 x^3 & = & b^2 \\
 & c^2 x^2 + c^3 x^3 & = & \min
 \end{aligned}$$

as a linear program to find optimal x^2 and x^3 .

- (b1) Show that the optimal prices associated with I' and II' are, in general, not unique even when $\hat{\pi}^1$ and $\hat{\pi}^2$ are unique for I and II.
- (b2) Describe computational difficulties that might arise in solving I' and II' due to small round-off errors in the forming of $b^1 - A^1 \hat{x}^1$.

10.7 Show that solving

$$\begin{aligned}
 (c^1)^T x^1 + \theta & = z \text{ (min)} \\
 A^1 x^1 & = b^1 \\
 \theta - (c^2)^T x^2 & \geq 0 \\
 B^1 x^1 + A^2 x^2 & = b^2
 \end{aligned} \tag{10.161}$$

is equivalent to solving

$$\begin{aligned}
 (c^1)^T x^1 + (c^2)^T x^2 & = z \text{ (min)} \\
 A^1 x^1 & = b^1 \\
 B^1 x^1 + A^2 x^2 & = b^2.
 \end{aligned} \tag{10.162}$$

Next show that adding the constraint

$$(G^1)^T x^1 + \theta \geq g_1 \tag{10.163}$$

to (10.161), where $G^1 = (B^1)^T \pi^2$ and $g_1 = (b^2)^T \pi^2$ for any given π^2 such that $(A^2)^T \pi^2 \leq c^2$ is also equivalent to solving (10.162).

10.8 *Ph.D. Comprehensive Exam, September 24, 1983, at Stanford.* Consider the following linear program to find $x^1 \geq 0, \theta \geq 0, x^2 \geq 0$, and $\min z$:

$(c^1)^T x^1 + \theta$	$= z \text{ (min)}$	(10.164)
$A^1 x^1$	$= b^1$	
$(G^1)^T x^1 + \theta$	$\geq g_1$	
$B^1 x^1$	$+ A^2 x^2 = b^2$	
$+ \theta$	$- (c^2)^T x^2 \geq 0$	

where $G^1 = (B^1)^T \pi^2$ and $g_1 = (b^2)^T \pi^2$ for any given π^2 such that $(A^2)^T \pi^2 \leq c^2$. Assume that $c^2 \geq 0$ and that this system is feasible and optimal solutions exist and for the latter, $\min z = z_{\min}$.

Also assume that an optimal solution $\hat{x}^1, \hat{\theta}, w_{\min}$ exists for the system (10.165):

$$\begin{array}{l|l} (c^1)^T x^1 + \theta & = w \text{ (min)} \\ \hline A^1 x^1 & = b^1 \\ (G^1)^T x^1 + \theta & \geq g_1 \\ \hline x^1 \geq 0, \theta \geq 0. & \end{array} \quad (10.165)$$

For the optimal solution to (10.165), assume that

$$A^2 x^2 = b^2 - B^1 \hat{x}^1, \quad x^2 \geq 0 \quad (10.166)$$

is feasible.

All parts of this question should be easy for you to prove.

(a) Prove there exists an $x^2 = \hat{x}^2 \geq 0$ and $\pi^2 = \hat{\pi}^2$ such that

$$\begin{aligned} A^2 \hat{x}^2 &= b^2 - B^1 \hat{x}^1, \\ (A^2)^T \hat{\pi}^2 &\leq c^2, \quad (\hat{\pi}^2)^T A^2 \hat{x}^2 = (c^2)^T \hat{x}^2. \end{aligned}$$

(b) Prove for all feasible solutions to (10.164): $z \geq z_{\min}$.

(c) Prove for all optimal feasible solution to (10.165): $w_{\min} \leq z_{\min}$.

(d) Define $\theta^* = (c^2)^T \hat{x}^2$. Prove that $\hat{x}^1, \theta^*, \hat{x}^2$ is a feasible solution to (10.164).

(e) Prove that $(c^1)^T \hat{x}^1 + \theta^* \geq z_{\min} \geq (c^1)^T \hat{x}^1 + \hat{\theta}$.

(f) Prove that $\theta^* = \hat{\theta}$ implies $(\hat{x}^1, \theta^*, \hat{x}^2)$ is optimal.

(g) Prove that if $\hat{\theta} < \theta^*$, then $\hat{\theta} < (c^2)^T \hat{x}^2$.

(h) Show that $(\hat{\pi}^2)^T B^1 \hat{x}^1 + \theta \geq (\hat{\pi}^2)^T b^2$ for all feasible (x^1, θ, x^2) .

(i) Show that $(\hat{\pi}^2)^T B^1 \hat{x}^1 + \theta^* = (\hat{\pi}^2)^T b^2$.

(j) Show that a necessary condition for (x^1, θ) to be part of an optimal feasible solution to (10.164) is

$$G^2 x^1 + \theta \geq g_2 \quad \text{where} \quad G^2 = (\hat{\pi}^2)^T B^1 \quad \text{and} \quad g_2 = (\hat{\pi}^2)^T b^2$$

and this inequality is not satisfied for $(\hat{x}^1, \hat{\theta})$ except if $\hat{\theta} = \theta^*$; that is,

$$(\hat{\pi}^2)^T B^1 \hat{x}^1 + \hat{\theta} < (\hat{\pi}^2)^T b^2 \text{ if } \hat{\theta} < \theta^*.$$

(k) Discuss how adjoining inequalities to (10.165) of the type above leads to a finite iterative process for solving a partitioned system providing the π^2 of part 1 are extreme dual solutions.

10.9 *Ph.D. Comprehensive Exam, September 22, 1990, at Stanford.* Given a linear program (10.167) with the following structure

$$\begin{array}{ll} \text{Minimize} & (c^1)^T x^1 + (c^2)^T x^2 = z \\ \text{subject to} & A^1 x^1 = b^1 \\ & -B^1 x^1 + A^2 x^2 = b^2 \\ & x^1, x^2 \geq 0 \end{array} \quad (10.167)$$

where A^1 is $m_1 \times n_1$ and A^2 is $m_2 \times n_2$. It happens that m_1 and m_2 are so large that it is not feasible to solve the problem with your software.

Someone tells you that $(x^{1,0}, x^{2,0})$ is an optimal solution but you wish to verify it. Your software can solve linear programs with m_1 equations and m_2 equations but not with $m_1 + m_2$ equations.

You derive the following procedure:

- (a) You test whether or not $x^1 = x^{1,0}$ is feasible for $A^1 x^1 = b^1$, $x^1 \geq 0$ and discover $x^{1,0}$ is feasible but not a basic solution to $A^1 x^1 = b^1$, $x^1 \geq 0$. How did you discover that it was not a basic solution?

Why, in general, given $(x^{1,*}, x^{2,*})$, an optimal feasible solution to (10.167), would you expect $x^{1,*}$ to be not basic for $A^1 x^1 = b^1$, $x^1 \geq 0$?

- (b) You optimized

$$\begin{aligned} \min (c^2)^T x^2 &= z - (c^1)^T x^{1,0} \\ A^2 x^2 &= b^2 + B^1 x^{1,0} \\ x^2 &\geq 0 \end{aligned} \quad (10.168)$$

and discovered that $x^2 = x^{2,0}$ is indeed an optimal solution and the optimal dual prices are $\pi^2 = \pi^{2,0}$. Keeping in mind that an optimal solution need not be unique, how did you discover that $x^{2,0}$ was an optimal feasible solution?

Why, in general, would you expect an optimal solution to be degenerate and the corresponding basic prices not unique, when $(x^{1,0}, x^{2,0})$ is optimal for (10.167)?

- (c) You next optimized

$$\begin{aligned} \min (c^1 + (\pi^{2,0})^T B^1)^T x^1 &= z - (\pi^{2,0})^T b^2 \\ A^1 x^1 &= b^1 \\ x^1 &\geq 0 \end{aligned} \quad (10.169)$$

and discovered that $x^{1,0}$ is an optimal feasible solution even though not a basic solution. How were you able to ascertain it was optimal for (10.169)?

- (d) Having ascertained that $(x^{1,0}, x^{2,0})$ is feasible for (10.167), and $(x^{2,0}, \pi^{2,0})$ are optimal primal and dual solutions for (10.168) given $x^1 = x^{1,0}$, and having ascertained that $x^1 = x^{1,0}$ is optimal for (10.169), prove that $(x^{1,0}, x^{2,0})$ is indeed an optimal feasible solution to (10.167).

This page intentionally left blank

C H A P T E R 11

STOCHASTIC
PROGRAMMING:
INTRODUCTION

Stochastic Programming, is the Art and Science of deciding on the best plan of action (in some expected-value sense) while hedging against the myriad of possible ways the best laid plans can go awry. Stochastic Mathematical Programming belongs to the general field of Planning Under Uncertainty which includes such topics as: Dynamic Programming, Decision Trees, Simulation, Stochastic Processes, and Chance Constrained Systems. It differs from deterministic mathematical programs only in that some of the parameters (coefficients and right-hand sides) may not be known at the time the decision is made.

Most important real-world models have some degree of uncertainty in the values of some of its model parameters and can often make a significant difference when these uncertainties are properly taken into account. Although deterministic mathematical programs are routinely solved by industry and government, which often involve thousands of variables with a linear or nonlinear objective and many thousands of inequality constraints, typically these are formulated as if the values of the coefficient matrix and the values of the constant terms are known with certainty. The solutions obtained are often ignored by those doing planning because these results do not properly hedge against future contingencies that might arise.

Each decade from the 1950s on has witnessed the development of more and more powerful techniques that properly incorporate uncertainty in the values of coefficients and right-hand side directly as part of the model formulation. The particular form of *stochastic programs* that we will consider is that of finding an

“optimal” solution to a linear program whose coefficients and constant terms are not known at the time the decision is made but whose probability distributions are known or have been estimated based on historical experience. In this and the next chapter, we will discuss some old ideas and some exciting new ideas.

11.1 OVERVIEW

Historically, planners have used various devices such as *sensitivity analysis* as a way to determine how robust proposed solutions are to changes. For example, sometimes planners *hedge* against running out of stock by overstating what they need, i.e., by incorporating lots of *fat* in the system. If plans are made by overstating actual needs and understating actual availabilities, then, should the disastrous happen, it is highly unlikely for the planned optimal set of activities to turn out to be infeasible. Consumption rates, production rates, and the like are all estimated on the high side so that whatever the extreme values of the demand turn out to be, the planners’ solution will still remain feasible. The effect of an unfavorable future event (should it happen) can also be further reduced by deliberately understating the amount of exogenous scarce resources available to the system. It turns out these ways often lead to an infeasible program. Historically, a new program would have to be developed based on a slower time schedule. This slowed down the demand until supply caught up. Often this ran the risk of entering the battle with too little too late. In World War II proper planning could have ended the war at least one year earlier, with the saving of millions of lives.

Adding fat, understating resource availability, and allowing delays are ways to hedge against uncertainty but at a price; typically the resulting plans are *very very inefficient* because they are too costly or too late to do any good. This chapter is concerned with finding solutions that have low expected costs, while hedging against contingencies that may arise, and also taking advantage of favorable events that may arise.

Research started in 1955 with George Dantzig’s paper and, independently, one by Martin Beale, and an application paper by Ferguson and Dantzig that was published a year later. This was about the same time that computers became reliable. Earlier in 1952 and then in 1959 Harry Markowitz began in earnest developing Portfolio Analysis, the first successful application of Stochastic Programming for which, years later, he received the Nobel Prize. Most of the field’s early pioneers did their research in isolation, unappreciated and undersupported. The stochastic programming field grew at a snail’s pace until the late 1980s when parallel computers, powerful workstations and PCs became a reality. This availability spurred researchers to try their hand at solving practical uncertainty problems. To everyone’s surprise, using a combination of techniques such as large-scale methods, D-W and Benders Decomposition, Importance Sampling, and Sampling-Space Partitioning, it turned out to be possible to solve many important practical cases. By “solve”, we mean in the practical sense of determining strategic decisions that are demonstrably superior to those obtained by ways that fail to properly take uncer-

tainty fully into account—shortcuts, such as replacing the technological structure of a model by one easier to optimize; or by making up ground rules that allow one to optimize over a reduced set of possible alternatives.

Plans that properly hedge against future risks can make a significant improvement. On some real and some realistic test problems these improvements are manifested in many ways, such as increases in safety, reliability, saving of lives, profitability, or control of risk.

On the other hand, failure to properly plan under uncertainty can be disastrous. Here are some examples:

1. A single car breaks down on the freeway and hundreds are caught in a horrific traffic jam. This happens all the time and keeps getting worse.
2. A single circuit breaker tripped in a storm in upstate New York blacked out for days the whole Eastern Seaboard. There was a significant peak in the birth rate nine months later.
3. A power failure in Idaho cascaded into a series of power failures from Canada to Mexico.
4. A single satellite went out of sync, blacking out communications over a large area of North America.
5. Our homes and offices depend on electric power from a single source, yet few of us have emergency generators in our homes.
6. Vivid in our memory are scenes of death and destruction due to earthquakes, floods, hurricanes, and sabotage of buildings and subways.

All of these disasters could have been significantly mitigated by better system design and recovery strategies, particularly those generated by models that properly hedge against the myriad of possible contingencies that might arise.

Often it is not some single event but the simultaneous occurrence of two or more rare events that start a cascade of events that becomes a major disaster. Two examples:

1. A weak transformer, hot weather, and an improper shift of load in a power system.
2. Doors locked blocking escape, poorly trained emergency personnel, panic, and the late arrival of fire trucks due to a traffic jam.

A key reason why plans have failed in the past is that planners did not know how to develop and implement strategies that properly adapt to a spate of unusual emergencies that might arise in the future. At best, they knew how to hedge against one or two of the myriad of possible ways that the best-laid plans of mice and men (to paraphrase the poet Robert Burns) can go awry.

Mitigating World Crisis: With the methodology developed so far, it is possible to develop adaptive strategic plans that, if implemented, would go a long way toward mitigating some of the world's chronic crises such as *over-population, resource depletion, floods, starvation, plague (AIDS), drought, and worldwide economic malaise.*

So far, all of these truly global disasters are still out of control. Perhaps we humans are incapable of ever getting our act together, and the only way that we humans will ever come to a sustainable equilibrium with nature will be (as in the past) by war, famine, and plague.

While we may not be able to stop an earthquake or a flood from happening, it is possible, by applying our methodology, to mitigate its disastrous effects *before, during, and after* disaster strikes by developing and implementing adaptive strategic plans that hedge against the many possible contingencies that can arise in the future. These chronic crises can be mitigated by developing flexible strategic plans that hedge against the myriad of contingencies that might arise and by adaptively reoptimizing future plans as events unfold in the future. While it may be possible to mitigate chronic crises by adaptive strategic plans, we will never be able to achieve this goal unless we find a way to bring about close cooperation between planners charged with finding a solution and those researchers who know how to apply the techniques.

11.2 UNCERTAIN COSTS

11.2.1 MINIMUM EXPECTED COSTS

We will illustrate the basic concepts, using as our example the nutrition problem. A housewife wishes to buy a diet for her family. The vector $A_{\bullet j}$ is the assumed vector of calories, proteins, fats, carbohydrates, vitamins, and minerals per unit of purchase; c_j is the cost per unit of purchase; and b is the vector of material requirements.

Deterministic Case: If the values of all the parameters are known in advance of the purchase, then the housewife obtains the matrix $\begin{pmatrix} c^T \\ A \end{pmatrix}$ from, say the Internet, and inputs the vector of nutritional requirements b of her family and asks the computer to find the diet $x = (x_1, x_2, \dots, x_n)$ that minimizes the cost $z = c^T x$; that is

$$\begin{aligned} &\text{Minimize} && c^T x = z \\ &\text{subject to} && Ax = b, && A: m \times n, \\ &&& x \geq 0. \end{aligned} \tag{11.1}$$

Stochastic Case: Assume that parameters A and b are known in advance of making a decision x , but the *costs c_j for each item j are not known with certainty.* For example, a housewife must decide what diet to buy for her family *before* she knows what the latest prices are. However, she does have some idea what the

expected price $E[c_j]$ for each item j is likely to be. Having chosen an x satisfying $Ax = b$, $x \geq 0$, the total cost z is $\sum_{j=1}^n c_j x_j$, a weighted sum of random variables c_j . Then, since her purchase of x_j of items j will be too small to effect the market price c_j , then the expected cost is given by (11.4) where ϕ is the probability density distribution on c_j :

$$E[z] = \int \cdots \int \phi(c_1, c_2, \dots, c_n) \sum_{j=1}^n c_j x_j dc_1 dc_2 \cdots dc_n \quad (11.2)$$

$$= \sum_{j=1}^n c_j \left[\int \cdots \int \phi(c_1, c_2, \dots, c_n) dc_1 dc_2 \cdots dc_n \right] x_j \quad (11.3)$$

$$= \sum_{j=1}^n E[c_j] x_j, \quad (11.4)$$

We have thus proved the following lemma.

LEMMA 11.1 *If the distribution $\phi(c_1, c_2, \dots, c_n)$ of the costs c_j to buy a unit amount of j is independent of the amount x_j , then the minimum expected total cost of purchases $\sum_{j=1}^n c_j x_j$ is obtained by finding $x \geq 0$ satisfying $Ax = b$ and minimizing $\sum_{j=1}^n E[c_j] x_j$.*

Suppose next that the costs c_j do depend on x_j but are independent of x_k for $k \neq j$. We then write $c_j x_j = \phi_j(x_j)$. In this case, the expected cost is

$$E[z] = \sum_{j=1}^n E[\phi_j(x_j) x_j] = \sum_{j=1}^n f_j(x_j) \quad (11.5)$$

where $f_j(x_j)$ is not necessarily linear in x_j . In this case special separable nonlinear optimization methods will be needed to solve the resulting problem. When $f_j(x_j)$ are convex functions, the method discussed in *Linear Programming 1* can be applied to solve the problem.

- ▷ **Exercise 11.1** Discuss situations where unit cost c_j goes up with increasing x_j and other situations where c_j goes down with increasing x_j . Show that $\phi_j(x_j) x_j$ is a convex function of x_j if $\phi_j(x_j)$ is increasing with increasing x_j but is no longer convex if $\phi_j(x_j)$ is decreasing with increasing x_j .

11.2.2 MINIMUM VARIANCE

In a number of applications it is desirable to minimize the risk, i.e., variance of the expected costs. For example, a stockbroker might advise a client to buy a portfolio of stocks j , some of which historically have had low return r_j with low variability

and other stocks that have high return r_j with high variability. A typical objective would be to buy a portfolio of stocks that gives at least a desired level r_o of expected return while minimizing the overall variability. In order to help us determine such a portfolio of stocks we would set up a portfolio optimization model as follows.

Let $E[r_j] = \bar{r}_j$, $E[(r_j - \bar{r}_j)^2] = \sigma_j^2$, and $E[(r_j - \bar{r}_j)(r_k - \bar{r}_k)] = \sigma_j \sigma_k \rho_{jk} = \sigma_{jk}$. Assuming that the costs are independent of the x_j , the variance of is $x_j^2 \sigma_{jj}$ and the covariance between $r_j x_j$ and $r_k x_k$ is $x_j x_k \sigma_j \sigma_k \rho_{jk}$. Thus, it follows that the variance of the objective function is the *quadratic*:

$$Q = E \left[\left(\sum_{j=1}^n (r_j - \bar{r}_j) x_j \right)^2 \right] = \sum_{j=1}^n \sum_{k=1}^n x_j x_k \sigma_{jk} = x^T M x, \quad (11.6)$$

where $M_{jk} = M_{kj} = \sigma_{jk}$. Then we solve the problem

$$\begin{aligned} \text{Minimize} \quad & \sum_{j=1}^n \sum_{k=1}^n x_j x_k \sigma_{jk} = Q \\ \text{subject to} \quad & Ax = b, \quad A: m \times n, \\ & \sum_{j=1}^n \bar{r}_j x_j \geq r_o, \\ & x \geq 0. \end{aligned} \quad (11.7)$$

▷ **Exercise 11.2** Prove that Q is a positive semi-definite quadratic form in x_j .

There are three cases to consider.

Case 1: $Q^{1/2}$ Is Linear. If the cost coefficients are so highly correlated that the correlation coefficient $\rho_{jk} \approx 1$ for all $j \neq k$ then $\sigma_{jk} \approx \sigma_j \sigma_k$, and

$$Q^{1/2} \approx x_1 \sigma_1 + x_2 \sigma_2 + \cdots + x_n \sigma_n. \quad (11.8)$$

We solve problem (11.7) by minimizing the linear function $Q^{1/2}$ subject to $Ax = b$, $\sum_{j=1}^n E[r_j] x_j \geq r_o$, $x \geq 0$. The effect on Q of varying r_o can then be studied by solving a standard parametric programming problem.

Case 2: Q Is a Sum of Squares. If, on the other hand, the correlation between cost coefficients are $\rho_{jk} = 0$ for all $j \neq k$, then

$$Q = x_1^2 \sigma_1^2 + x_2^2 \sigma_2^2 + \cdots + x_n^2 \sigma_n^2. \quad (11.9)$$

In this case, Q is convex separable and the convex functions x_j^2 may be approximated by piecewise linear functions or by applying a convex quadratic programming algorithm. This again reduces to a standard parametric programming problem if we wish to study the effects of changing r_o on Q .

Case 3: Q is general. In this case $Q = x^T M x$ where M is positive semi-definite. Then Q can be reduced to Case 2 by a suitable transformation. For example, by factoring $M = DD$. Then (11.7) can be written as $\min y^T y$ such that $Ax = b$, $\sum_{j=1}^n \bar{r}_j x_j \geq r_o$, $Dx - y = 0$, $x \geq 0$. We can then solve this by the piecewise linear approximation method discussed in *Linear Programming 1*.

Comment: Typically the variances and covariances σ_{jk} are not available but can be estimated from historical returns. Let r_j^i be the returns in historical period i for $i = 1, \dots, t$ for stocks j for $j = 1, \dots, n$, and let $\bar{r}_j = (1/t) \sum_{i=1}^t r_j^i$ be the mean return. We can compute the variance-covariance matrix $M = [\sigma_{jk}]$ by forming $M = (1/t) \bar{R}^T \bar{R}$, where

$$\bar{R} = \begin{pmatrix} r_1^1 - \bar{r}_1 & r_2^1 - \bar{r}_2 & \cdots & r_n^1 - \bar{r}_n \\ r_1^2 - \bar{r}_1 & r_2^2 - \bar{r}_2 & \cdots & r_n^2 - \bar{r}_n \\ \vdots & \vdots & \cdots & \vdots \\ r_1^t - \bar{r}_1 & r_2^t - \bar{r}_2 & \cdots & r_n^t - \bar{r}_n \end{pmatrix}. \quad (11.10)$$

11.3 UNCERTAIN DEMANDS

Scheduling to meet an uncertain right-hand side, such as demand, is a special case of a more general two-stage problem to be discussed later in this chapter. In the first stage a decision is made regarding how much, for example, to ship *prior* to knowing what the demands in the second stage will be. We assume instead that we know what the distribution of the demand for each of the various items will be.

Example 11.1 (Uncertain Demand) Suppose that a factory has an inventory of 100 units of some kind, of which $x \leq 100$ must be shipped to an outlet at a shipping cost of \$1 per unit to meet an uncertain demand of d units, where the distribution of d is known. In this oversimplified example, the revenues from selling the item are the same in any scenario. It is also assumed that the value of any leftover supply is written as zero. The shipping is done before the demand for the item is known, and hence it is possible that the demand will be less than the amount shipped; if so, let t denote the amount oversupplied. In the event, however, that the demand exceeds supply, it is required, in order not to lose the customer to the competition, that s items be purchased on the open market to meet the shortage at a cost of \$2 per unit. The equations that must be satisfied are then:

$$\begin{aligned} x &+ 2s &= C \\ x + y &&= 100 \\ x &+ s - t &= d \\ (x, y, s, t) &\geq 0 \end{aligned} \quad (11.11)$$

where x is the number of units shipped from the factory, y is the number stored at the factory, s (shortage) is the number purchased on the open market, t (too much) is the excess supply over demand, d is the unknown demand with a known probability of demand distribution, and C is the total cost. The problem is to determine how much to ship in order to minimize expected cost. This example belongs to a more general class of two-stage problems, which we will discuss in Example 11.2.

▷ **Exercise 11.3** For Example 11.1, suppose that the costs are given by

$$C = x + 2 \text{Max}(0, d - x)$$

where d is uniformly distributed between 70 and 80. Determine how much to ship in order to minimize expected cost. *Hint:* Determine the expected cost $E[C]$ explicitly as a function of x and then determine the value of x that minimizes this function.

Example 11.2 (Two-Stage Problem) In the first stage, $x_j \geq 0$ and $u_k \geq 0$ are determined such that

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, \dots, m, \quad (11.12)$$

$$\sum_{j=1}^n \bar{a}_{kj} x_j = u_k, \quad k = 1, \dots, l, \quad (11.13)$$

where b_i , the initial inventories of raw materials, known in advance, are transformed by the technology matrices $[a_{ij}]$, $[\bar{a}_{kj}]$ and activity levels x_j into finished products u_k to meet an uncertain demand d_k in the second stage. The quantities x_j are decisions in the first stage that result in the quantities u_k , $k = 1, \dots, l$ being available in the second stage. If the amount supplied u_k is less than the demand d_k then let $s_k \geq 0$ be the shortage; on the other hand, if u_k is greater than the demand d_k , let t_k be the excess. Then

$$d_k = u_k + s_k - t_k, \quad (11.14)$$

where either $s_k = 0$ or $t_k = 0$ or both $s_k = 0$, $t_k = 0$; d_k is the uncertain demand with a known probability distribution (where (d_1, d_2, \dots, d_n) may be independent or dependent random variables); s_k is the shortage of supply of k ; and t_k is the excess of supply of k over demand of k .

To simplify the discussion, in this example we assume that *it is not possible to make purchases* on the open market in the case of shortages s_k . It clearly pays to sell as much of the supplied amount, u_k , as possible, that is, $\min(u_k, d_k) = d_k - s_k$. Therefore the total cost is

$$C = \sum_{j=1}^n c_j x_j - \sum_{k=1}^l f_k \min(u_k, d_k), \quad (11.15)$$

where c_j is the cost of one unit of activity j and f_k is the revenue obtained by satisfying one unit of demand of item k .

For every fixed value of x_k , and hence by (11.13), for every fixed value of u_k , the expected cost is:

$$E[C] = \sum_{j=1}^n c_j x_j - \sum_{k=1}^l f_k E[\min(u_k, d_k)]. \quad (11.16)$$

Since the expectation $E[\min(u_k, d_k)]$ is taken with respect to the distribution of demand, it is some function

$$\phi_k(u_k) = E[\min(u_k, d_k)]. \quad (11.17)$$

When, for fixed k , the values of $d = d_k$ are discrete and can take on R discrete values $g_1 \leq g_2 \leq \dots \leq g_R$ with probabilities p_1, p_2, \dots, p_R where $\sum_{i=1}^R p_i = 1$, the computation of $\phi(u) = E[\min(u, d)]$, for any $u = u_k$, $d = d_k$, is straightforward. To see this, note that

$$\min(u, d) = \begin{cases} u & \text{if } d > u; \\ d & \text{if } d \leq u. \end{cases} \quad (11.18)$$

For a given u let r be such that $g_{r-1} \leq u < g_r$ holds for some $r = 1, \dots, R$ where by definition $g_0 = 0$. Then:

$$\begin{aligned} \phi(u) &= E[\min(u, d)] = uP[d > u] + \sum_{j=1}^{r-1} g_j P[d = g_j] \\ &= u \sum_{i=r}^R p_i + \sum_{j=1}^{r-1} g_j p_j \\ &= u \left(1 - \sum_{i=1}^{r-1} p_i \right) + \sum_{j=1}^{r-1} g_j p_j. \end{aligned} \quad (11.19)$$

▷ **Exercise 11.4** Suppose that d_k takes the values 1 and 2 with probabilities 1/4 and 3/4, respectively. Compute the expected value $E[\phi_k(u_k)]$.

▷ **Exercise 11.5** Plot the $\phi(u)$, the expected revenue, as a function of the amount supplied u .

We see that $\phi(u)$ is a broken line function starting at $u = 0$ with initial slope $\beta_1 = 1$; at $u = g_1$ the slope decreases by p_1 to $\beta_2 = 1 - p_1$; at $u = g_2$ the slope decreases by p_2 to $\beta_3 = 1 - p_1 - p_2$; etc. Thus, $-\phi(u)$ is a convex function because the slopes are increasing. We have thus shown the following.

THEOREM 11.2 (Convexity of Total Expected Costs) *Under uncertain demand, the total expected cost is a convex separable function*

$$E[C] = \sum_{j=1}^n c_j x_j - \sum_{k=1}^l f_k \phi_k(u_k) \quad (11.20)$$

where $\phi_k(u_k)$ is a piecewise linear function whose slope between two successive demands $d_k = g_{r-1,k}$ and $d_k = g_{r,k}$ is equal to the probability $1 - \sum_{i=1}^{r-1} p_{ik}$ of exceeding the demand $g_{r-1,k}$.

Then to minimize expected costs we minimize the convex separable functions (11.20) subject to the constraints (11.12) and (11.13); see *Linear Programming 1*.

11.4 NOTES & SELECTED BIBLIOGRAPHY

For an introduction to probability theory, see Feller [1957, 1969]. A minimum variance portfolio selection problem was first considered by Markowitz [1952]. The incorporation of uncertainty into linear programs was proposed independently by Dantzig [1955a] and Beale [1955a]. From then on, various individuals have tried to extend the methods of linear programming to handle the problem of optimizing an objective function whose constants are subject to random variations. Early references are Ferguson & Dantzig [1956], Madansky [1959], and Dantzig & Madansky [1961]. Over the years, different approaches have been used to attack this problem. See, for example, Birge [1985a, 1985b], Birge & Wallace [1988], Birge & Wets [1986], Ermoliev [1983], Frauendorfer [1988], Frauendorfer & Kall [1988], Higle & Sen [1991], Kall [1979], Pereira, Pinto, Oliveira, & Cunha [1989], Rockafellar & Wets [1989], Ruszczyński [1986], Van Slyke & Wets [1969], and Wets [1984]. A survey of different ways to solve stochastic linear programs can be found in Ermoliev & Wets [1988], and an introduction to stochastic programming can be found in Birge & Louveaux [1997].

The *two-stage* case was first studied by Dantzig [1955a, 1963] and subsequently developed by Van Slyke & Wets [1966, 1969] and Wets [1984].

Example 11.1 is adapted from Dantzig [1963]. Theorem 11.2 was verbally communicated to Dantzig by H. Scarf. An important pioneering application of Stochastic Programming is Alan Manne's [1974] paper, "Waiting for the Breeder."

11.5 PROBLEMS

- 11.1 *Dantzig [1963]*. Solve the problem of Example 11.1 using the discrete distribution $d = 70, 71, \dots, 80$ with probability $1/11$ each.
- 11.2 *Dantzig [1963]*. Consider the following transportation problem:

				3
	2	3	4	1
	7	2	5	1
	4	3	2	2
d_1	d_2	d_3	d_4	

Solve the transportation problem

- (a) When the demands $d_1 = 3$, $d_2 = 3$, $d_3 = 2$, and $d_4 = 2$ are certain; i.e., occur with probability 1.
- (b) When the demands have the following probability distribution

$d_1 = 2, 3, 4$ with equal probabilities $1/3$.

$d_2 = 2, 3, 4$ with equal probabilities $1/3$.

$d_3 = 1, 2, 3$ with equal probabilities $1/3$.

$d_4 = 1, 2, 3$ with equal probabilities $1/3$.

- 11.3 *Dantzig [1963]*. Consider a linear program in which all the coefficients are uncertain. Find $x_j \geq 0$, for $j = 1, \dots, n$:

$$\begin{aligned}\epsilon_o(x) &= \sum_{j=1}^n a_{oj}x_j = z(\text{Min}), \\ \epsilon_i(x) &= \sum_{j=1}^n a_{ij}x_j + a_o \leq 0. \quad \text{for } i = 1, \dots, m\end{aligned}$$

The minimum z is desired, but unfortunately all the x_j must be selected *prior* to a random choice of the coefficients a_{ij} whose distributions are, however, known.

- (a) Denote by $\sigma_i(x)$ the standard error of $\epsilon_i(x)$. Show that

$$\sigma_i(x) = \left(\sum_{j=1}^n \sum_{k=1}^n x_j x_k E[(a_{ij} - \bar{a}_{ij})(a_{ik} - \bar{a}_{ik})] \right)^{1/2}.$$

- (b) Suppose we solve the program

$$\begin{aligned}\epsilon_o(x) + t_o \sigma_o(x) &= z(\text{Min}), \\ \epsilon_i(x) + t_i \sigma_i(x) &\leq 0 \quad \text{for } i = 1, \dots, m, \\ x &\geq 0,\end{aligned}$$

where $t_i = 3$, say, means that we have built in a safety factor so that $\bar{\epsilon}_i(x)$, the expected value of $\epsilon_i(x)$, is three standard errors below zero. Prove that this is a *convex program*.

- (c) Show by Tchebycheff's inequality that

$$\text{Prob}[\epsilon_i(x) > 0] < \frac{1}{t_i^2}$$

What is the probability if $\epsilon_i(x)$ is approximately normally distributed?

- (d) Show that if a_{ij} are independent and normally distributed, then $\epsilon_i(x)$ is normally distributed.

This page intentionally left blank

TWO-STAGE STOCHASTIC PROGRAMS

An important class of optimization problems arise in dynamic systems that describe activities initiated at various times $t_0, t_0 + 1, \dots, t, \dots, T$. Those initiated at time t have coefficients at time t and $t + 1$. Such problems, called *dynamic linear programs*, have a *staircase structure*. In the **deterministic case**, the coefficient matrices, constants, and cost coefficients for each stage are known with certainty. Often in practice, initial decisions must be made *prior to random events* that might occur in the future, such as the possible failure of equipment or the possible introduction of new technologies. This is the **stochastic case** in which the coefficient matrices, constants, and cost coefficients for stage $t + 1$ become known only *after* stage t has taken place.

The simplest dynamic linear program has only two stages, which we will now consider; the techniques discussed here can be extended to the multistage problem. To simplify the notation for two-stage stochastic linear programs, we shall depart from treating all vectors as column vectors and subscripts as meaning a component of a vector.

12.1 THE DETERMINISTIC TWO-STAGE LP PROBLEM

Find minimum z , $x \geq 0$, $y \geq 0$, such that

$$\begin{array}{l} \text{1st Stage: } cx + fy = z \\ \text{2nd Stage: } Ax = b \\ \quad \quad \quad Bx + Dy = d, \end{array} \quad (12.1)$$

where, in the deterministic case, the components of the vectors and matrices c , A , b , B , D , d , f are all known with certainty.

12.2 THE ANALOGOUS STOCHASTIC TWO-STAGE LP PROBLEM

Find minimum z , $x \geq 0$, $y = (y_1, y_2, \dots, y_W) \geq 0$, such that:

$$\begin{aligned} cx + E_\omega[f_\omega y_\omega] &= z \\ Ax &= b \\ B_\omega x + D_\omega y_\omega &= d_\omega, \quad \omega = 1, \dots, W, \end{aligned} \quad (12.2)$$

where $p_\omega > 0$, $\sum_{\omega=1}^W p_\omega = 1$, the probability p_ω of the random event $\omega \in \Omega = \{1, \dots, W\}$ occurring is given, and

$$\theta = E_\omega[f_\omega y_\omega] = \sum_{\omega=1}^W p_\omega (f_\omega y_\omega) \quad (12.3)$$

measures the *expected* second-period cost. Assuming some iterative algorithm on its final iteration k has arrived at a final first-stage decision $x = x^k$, we are interested in measuring how good a solution $x = x^k$ is compared with the optimal solution $x = x^*$.

At the time t_o , when the first-stage decision x is made, all the components c , A , b are known with certainty. It is also assumed that there exists x that satisfies $Ax = b$, $x \geq 0$. For each possible future event $\omega = 1, \dots, W$, the values of $B = B_\omega$, $D = D_\omega$, $d = d_\omega$, $f = f_\omega$ are also assumed known at time t_o ; only the event ω is unknown. Assuming some first-stage decision $x = x^k$ is made, one of the possible events (scenarios, contingencies) $\omega \in \Omega$ happens at time t_1 with probability $p_\omega > 0$, so that by the time of the second-stage decision $t_1 > t_o$, the scenario ω with parameter values of $B = B_\omega$, $D = D_\omega$, $d = d_\omega$, $f = f_\omega$ have all become certain. *We will assume, in order to simplify the presentation, that whatever be $x^k \geq 0$ satisfying $Ax = b$, optimal $y_\omega = y_\omega^k$ exist.* Therefore, given ω and $x = x^k$, the optimal second-stage decision $y_\omega = y_\omega^k$, can be found by solving:

$$\begin{aligned} & \text{SUBPROBLEM } \omega \text{ given } x = x^k: \\ & \text{Find } y_\omega = y_\omega^k \geq 0 \text{ that} \\ & \text{minimizes } f_\omega y_\omega \\ & \text{subject to } D_\omega y_\omega = d_\omega - B_\omega x^k, \end{aligned} \quad (12.4)$$

where each second-stage subproblem ω is solved with $x = x^k$. The dual multipliers, given $x = x^k$, are denoted by π_ω^k . The optimal primal solution to the subproblem is denoted by $y_\omega = y_\omega^k$ and its *optimal dual multipliers* $\pi_\omega = \pi_\omega^k$. Then by the Duality Theorem:

$$\pi_\omega^k D_\omega \leq f_\omega, \quad (f_\omega - \pi_\omega^k D_\omega) y_\omega^k = 0, \quad y_\omega^k \geq 0. \quad (12.5)$$

The first-stage decision $x = x^k$ must be made *before* knowing what the outcome of the random event ω will be; however, *what we assume is known* is the probability distribution $p_\omega > 0$, $\sum_{\omega=1}^W p_\omega = 1$ of the random events ω occurring. Therefore the *MINIMUM EXPECTED COST* $\theta = E[f_\omega y_\omega]$ of the second-stage decision at time t_o can be calculated given any decision x :

$$\theta = E_\omega[f_\omega y_\omega] = \sum_{\omega=1}^W p_\omega (f_\omega y_\omega). \quad (12.6)$$

In particular, if $x = x^k$ and $y_\omega = y_\omega^k$ from (12.4), then the expected minimum cost

$$\theta^k = \theta[x^k] = \sum_{\omega=1}^W p_\omega (f_\omega y_\omega^k), \quad (12.7)$$

can be calculated for any proposed $x = x^k$ where we use the notation $\theta[x^k]$ to emphasize that θ is a function of x^k . Our goal is to analyze and develop techniques to solve the *STOCHASTIC PROBLEM*: Find $x = x^*$ that minimizes

$$z = cx + \theta[x], \quad (12.8)$$

i.e., minimizes the first-stage cost plus minimum expected second-stage cost given the first-stage decision x .

▷ **Exercise 12.1** Show that for the two-stage problem under uncertainty

$$\begin{aligned} \sum_{\omega=1}^W p_\omega y_\omega &= z(\text{Min}) \\ Ax &= b, \quad x \geq 0 \\ B_\omega x + y_\omega &= d_\omega, \quad \omega = 1, \dots, W, \end{aligned}$$

where $p_\omega > 0$, $\sum_{\omega=1}^W p_\omega = 1$, the probability p_ω of the random event $\omega \in \Omega = \{1, \dots, W\}$ occurring in the second stage is given, the optimal x can be determined by optimizing

$$\begin{aligned} - \left(\sum_{\omega=1}^W p_\omega B_\omega \right) x &= z(\text{Min}) \\ Ax &= b, \quad x \geq 0. \end{aligned}$$

Is the statement still true in general if the conditions $y_\omega \geq 0$ are imposed.

12.3 LP EQUIVALENT OF THE STOCHASTIC PROBLEM (EQ-LP)

12.3.1 LP EQUIVALENT FORMULATION

THEOREM 12.1 (Equivalent Formulation) *The linear program equivalent*

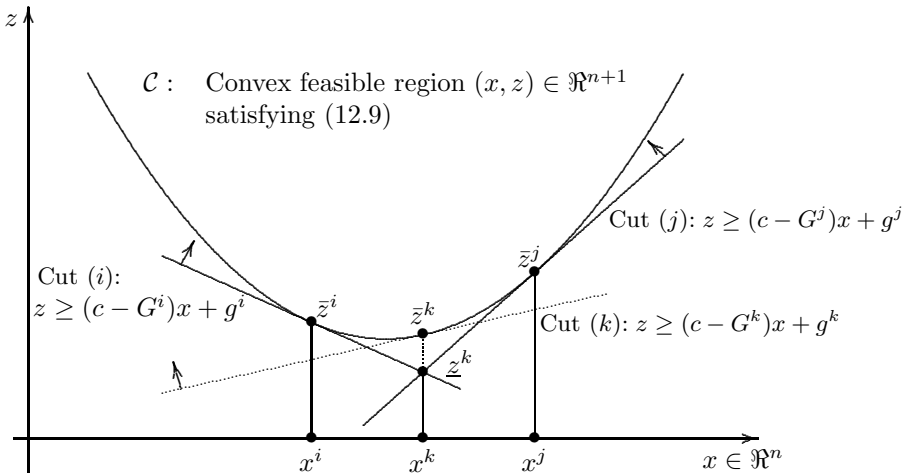


Figure 12-1: Benders Decomposition Applied to EQ-LP

in Figure 12-1 but is a collection of broken line segments, in the two-dimensional case, that are the envelope of the inequalities). At the start of iteration k , iterations $i = 1, \dots, k - 1$ have generated $x = x^i \geq 0$ satisfying $Ax^i = b$ and a point (x^i, \bar{z}^i) in \mathcal{C} with the property that of all the points (x^i, z) in \mathcal{C} , $\bar{z}^i = \min z$. At (x^i, \bar{z}^i) , a tangent (in general, a hyperplane) $z = (c - G^i)x + g^i$ is found that passes through (x^i, \bar{z}^i) .

Because of the convexity of \mathcal{C} , the *tangent hyperplanes* have the property that the corresponding half-spaces $z \geq (c - G^i)x + g^i$ contain \mathcal{C} and their boundaries $z = (c - G^i)x + g^i$ each have at least one point in common with \mathcal{C} . Such hyperplanes are called *tight supporting hyperplanes* or *tight supports*. Their associated half-spaces are called *tight cuts*, because each iteration generates a linear inequality that *cuts away* a portion of the (x, z) -space that does not contain \mathcal{C} . We denote by \mathcal{C}^{k-1} the set of points common to the first $k - 1$ cuts. Note that $\mathcal{C}^{k-1} \subset \mathcal{C}^{k-2} \subset \dots \subset \mathcal{C}^2 \subset \mathcal{C}^1$. (The symbol $\mathcal{C}^t \subset \mathcal{C}^s$ means that the set \mathcal{C}^t is strictly contained in the set \mathcal{C}^s .) The next iterate determines (x^k, \bar{z}^k) and the supporting hyperplane $z \geq (c - G^k)x + g^k$, which is tight at (x^k, \bar{z}^k) . For the algebraic formulas for computing G^k and g^k , see (12.19).

Geometrically, Cut (k) is generated in three steps: The *first step* views the convex set of points \mathcal{C}^{k-1} satisfying the $(k - 1)$ cuts as an *approximation* to \mathcal{C} and determines $(x, z) = (x^k, \underline{z}^k)$ the minimum z in \mathcal{C}^{k-1} . This is done by solving a linear program, called *Benders Master Program (k)*.

The *second step* determines the point (x^k, \bar{z}^k) in \mathcal{C} by fixing the first-stage decision at $x = x^k$ and finds $\bar{z}^k = cx^k + \sum_{\omega=1}^W p_{\omega} f_{\omega} y_{\omega}$ where each $y_{\omega} = y_{\omega}^k$ minimizes the second-stage subproblems ω given $x = x^k$; see (12.4).

The *third step* generates Cut (k) , the supporting half-space that is tight at

(x^k, \bar{z}^k) . This is done by *first multiplying* equation ω ,

$$B_\omega x + D_\omega y_\omega = d_\omega, \quad (12.12)$$

by $p_\omega \pi_\omega^k$, to obtain

$$p_\omega(\pi_\omega^k B_\omega)x + p_\omega(\pi_\omega^k D_\omega)y_\omega = p_\omega(\pi_\omega^k d_\omega). \quad (12.13)$$

where π_ω^k are the optimal dual-feasible solutions to subproblem (12.4) given $x = x^k$.

Next, adding to it the identity

$$-p_\omega(f_\omega y_\omega) + p_\omega(f_\omega y_\omega) = 0 \quad (12.14)$$

and rearranging terms, we obtain

$$p_\omega(\pi_\omega^k B_\omega)x + p_\omega(f_\omega y_\omega) - p_\omega(f_\omega - \pi_\omega^k D_\omega)y_\omega = p_\omega(\pi_\omega^k d_\omega). \quad (12.15)$$

Dropping the term $p_\omega(f_\omega - \pi_\omega^k D_\omega)y_\omega \geq 0$ (because $f_\omega - \pi_\omega^k D_\omega \geq 0$ by dual-feasibility, and $p_\omega > 0$, $y_\omega \geq 0$), we obtain the cut associated with ω and the optimal dual-feasible solution $\pi_\omega = \pi_\omega^k$, given $x = x^k$:

$$p_\omega(\pi_\omega^k B_\omega)x + p_\omega(f_\omega y_\omega) \geq p_\omega(\pi_\omega^k d_\omega). \quad (12.16)$$

To generate Cut (k) , in the space of x and θ , we sum (12.16) for all $\omega \in \Omega$ to obtain

$$\sum_{\omega=1}^W p_\omega(\pi_\omega^k B_\omega)x + \sum_{\omega=1}^W p_\omega(f_\omega y_\omega) \geq \sum_{\omega=1}^W p_\omega(\pi_\omega^k d_\omega), \quad (12.17)$$

or

$$G^k x + \theta \geq g^k, \quad (12.18)$$

where we denote

$$G^k = \sum_{\omega=1}^W p_\omega(\pi_\omega^k B_\omega), \quad g^k = \sum_{\omega=1}^W p_\omega(\pi_\omega^k d_\omega), \quad \theta = \sum_{\omega=1}^W p_\omega(f_\omega y_\omega). \quad (12.19)$$

To generate tight Cut (k) in the space of x and z , we eliminate θ from (12.18) by subtracting it from $cx + \theta \leq z$, see (12.9). Rearranging terms, we obtain the cut in the form displayed in Figure 12-1 and (12.20) below:

$$\text{Cut } (k): \quad z \geq (c - G^i)x + g^i. \quad (12.20)$$

THEOREM 12.2 (Cut (k) Is a Tight Support) *If the cut is generated using $\pi_\omega = \pi_\omega^k$, the optimal dual-feasible solutions to the subproblem ω given $x = x^k$, then Cut (k) defined by inequality (12.20) is a tight lower bound to \mathcal{C} at (x^k, \bar{z}^k) .*

Proof. The inequality Cut (k) (12.20) contains \mathcal{C} because it is generated by a nonnegative combination of inequalities and equations, each of which is satisfied by every feasible point (x, y) in \mathcal{C} , in particular $(x^k, y^k) \in \mathcal{C}$ where y_ω^k are the optimal primal solutions to subproblems ω given $x = x^k$.

If we substitute $y_\omega = y_\omega^k$, $\pi_\omega = \pi_\omega^k$ the *optimal primal and dual solutions* to subproblem ω given $x = x^k$ in (12.15), then, by the Duality Theorem, $(f_\omega - \pi_\omega^k D_\omega)y_\omega^k = 0$, see (12.5). Summing in this case (12.15) for all $\omega \in \Omega$ we obtain

$$\sum_{\omega=1}^W p_\omega(\pi_\omega^k B_\omega)x^k + \sum_{\omega=1}^W p_\omega(f_\omega y_\omega^k) = \sum_{\omega=1}^W p_\omega(\pi_\omega^k d_\omega), \quad (12.21)$$

which, see (12.19), we denote by

$$G^k x^k + \theta^k = g^k. \quad (12.22)$$

Because $\theta^k = \text{Min } \theta$ given $x = x^k$, we also have by the definition of \bar{z}^k

$$c x^k + \theta^k = \bar{z}^k. \quad (12.23)$$

Subtracting (12.23) from (12.22) to eliminate θ^k results in

$$\bar{z}^k = (c - G^i)x^k + g^i \quad (12.24)$$

which proves that the lower boundary point (x^k, \bar{z}^k) of \mathcal{C} lies on the hyperplane boundary of the Cut (k). ■

12.3.3 DECOMPOSITION ALGORITHM

Algorithm 12.1 (Benders Algorithm for Solving EQ-LP)

1. *Initialization.*

- (a) Set iteration count $k = 1$.
- (b) Set tolerance level = TOL .
- (c) Optimize

$$\begin{array}{l} \text{BENDERS MASTER to generate } x^k: \\ \text{Minimize} \quad cx = z \\ \text{subject to} \quad Ax = b, x \geq 0 \end{array}$$

If no feasible solution exists, *terminate* else find optimal $x = x^k$.

2. *Begin Iterative Loop.* For each $\omega \in \Omega$, optimize

SUBPROBLEM ω given $x = x^k$:

$$\begin{array}{l} \text{Find extreme points } y_\omega = y_\omega^k \geq 0 \text{ and corresponding optimal duals } \pi_\omega^k \text{ that} \\ \text{minimizes} \quad f_\omega y_\omega \\ \text{subject to} \quad D_\omega y_\omega = d_\omega - B_\omega x^k. \end{array}$$

Note: To simplify the discussion, we assume that all the subproblems are feasible. In practice, if a subproblem is not feasible either “feasibility cuts” are adjoined to the master or “penalty terms” are adjoined to the subs (see Exercise 12.3).

3. Calculate the expected minimum second-stage costs given $x = x^k$:

$$\theta^k = \sum_{\omega=1}^W p_{\omega}(f_{\omega}y_{\omega}^k)$$

and expected first-stage plus second-stage costs given $x = x^k$

$$\bar{z}^k = cx^k + \theta^k.$$

4. Create Cut (k):

$$G^k x + \theta \geq g^k$$

where

$$G^k = \sum_{\omega=1}^W p_{\omega}(\pi_{\omega}^k B_{\omega}); \quad g^k = \sum_{\omega=1}^W p_{\omega}(\pi_{\omega}^k d_{\omega}).$$

5. Adjoin Cut (k) to Benders Master Program and reoptimize:

BENDERS MASTER (k) to generate x^{k+1} :
 Find $\min z = \underline{z}^{k+1}$, $x = x^{k+1} \geq 0$, $\theta = \underline{\theta}^{k+1}$, such that
 $cx + \theta = z$
 $Ax = b$
 $G^i x + \theta \geq g^i$, for $i = 1, \dots, k$.

6. Set $L = \operatorname{argmin}_{i \leq k} \bar{z}^i$, $\bar{z}^L = \operatorname{Min}_{i \leq k} \bar{z}^i$.

7. If $\underline{z}^{k+1} + TOL < \bar{z}^L$ set $k \leftarrow k + 1$ and LOOP BACK to *Begin Iterative Loop* at Step 2.

8. If $\underline{z}^{k+1} + TOL \geq \bar{z}^L$, declare $z = \bar{z}^L$ as “close enough” to the minimum objective value and declare x^L as a first stage decision whose objective value \bar{z}^L is “close enough” to the minimum objective value; and STOP.

▷ **Exercise 12.3** Show how to generate cuts for the Master problem when one or more subproblems are infeasible. Show that these *infeasibility cuts* take the form $G^i x \geq g^i$. Show how this affects Step 5. Show how to incorporate penalty terms instead into such subproblems so that even an infeasible problem can be replaced by a feasible one having a high cost (penalty).

▷ **Exercise 12.4** Interpret geometrically, in Figure 12-1, the steps of the algorithm for iterations $i < k$, $j < k$, and iteration k .

▷ **Exercise 12.5** Change the position of \bar{z}^k in Figure 12-1 so that $\bar{z}^{k+1} > \bar{z}^k$, illustrating that the upper bounds do not always monotonically decrease with increasing k .

Example 12.1 (Benders Algorithm for Solving EQ-LP Illustrated) Consider the following stochastic linear program:

Minimize

$$x_1 + x_2 + x_3 + 0.5 * (3y_{11} + 2y_{12} + y_{13}) + 0.5 * (2y_{11} + 4y_{12} + 0y_{13}) = z$$

subject to

$$x_1 + 2x_2 + 3x_3 = 6$$

$$3x_1 + 2x_2 + x_3 = 6$$

$$x_1 + x_2 + x_3 + 4y_{11} - y_{12} + y_{13} = 9$$

$$3x_1 + 2x_2 + x_3 + 3y_{11} + 2y_{12} + y_{13} = 15$$

$$4x_1 - x_2 + x_3 + y_{11} + y_{12} + y_{13} = 9$$

$$2x_1 - 2x_2 + x_3 + 2y_{21} - 2y_{22} + y_{23} = 3$$

$$4x_2 + x_3 + 6y_{21} + y_{23} = 15$$

$$6x_1 + x_3 + 4y_{22} + y_{23} = 15$$

(12.25)

where $x_j \geq 0$ for $j = 1, 2, 3$, $y_{\omega k} \geq 0$ for $\omega = 1, 2$, $k = 1, 2, 3$. The problem is clearly in the following standard EQ-LP form:

$$\begin{aligned} c^T x + p_1 f_1^T y_1 + p_2 f_2^T y_2 &= z \text{ (min)} \\ Ax &= b \\ B_1 x + D_1 y_1 &= d_1, \\ B_2 x + D_2 y_2 &= d_2, \\ x \geq 0, y &\geq 0, \end{aligned}$$

where $x^T = (x_1, x_2, x_3)$, $y_1^T = (y_{11}, y_{12}, y_{13})$, $y_2^T = (y_{21}, y_{22}, y_{23})$, $c^T = (1, 1, 1)$, $f_1^T = (3, 2, 1)$, $f_2^T = (2, 4, 0)$, $d_1^T = (9, 15, 9)$, $d_2^T = (3, 15, 15)$,

$$\begin{aligned} A &= \begin{pmatrix} 1 & 2 & 3 \\ 3 & 2 & 1 \end{pmatrix}, \quad B_1 = \begin{pmatrix} 1 & 1 & 1 \\ 3 & 2 & 1 \\ 4 & -1 & 1 \end{pmatrix}, \quad D_1 = \begin{pmatrix} 4 & -1 & 1 \\ 3 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \\ B_2 &= \begin{pmatrix} 2 & -2 & 1 \\ 0 & 4 & 1 \\ 6 & 0 & 1 \end{pmatrix}, \quad D_2 = \begin{pmatrix} 2 & -2 & 1 \\ 6 & 0 & 1 \\ 0 & 4 & 1 \end{pmatrix}. \end{aligned}$$

Initialize the tolerance $TOL = 10^{-7}$. To apply Benders decomposition, we start by creating the Initial Restricted Master Problem:

$$\begin{aligned} c^T x + \theta &= z \text{ (min)} \\ Ax &= b, \\ x \geq 0, \theta &= 0. \end{aligned} \tag{12.26}$$

Given a solution $x = x^1$ to the Master Problem, we solve the subproblems for $\omega = 1, 2$:

$$\begin{aligned} f_\omega^T x &= v_\omega \text{ (min)} \\ D_\omega y_\omega &= d_\omega - B_\omega x^1, \\ y &\geq 0. \end{aligned} \tag{12.27}$$

The solutions of these subproblems are then used to define cuts for the Master Problem.

The solution to

$$\begin{aligned} &\text{Minimize} && x_1 + x_2 + x_3 + \theta = z \\ &\text{subject to} && x_1 + 2x_2 + 3x_3 = 6 \\ &&& 3x_1 + 2x_2 + x_3 = 6 \\ &&& x \geq 0, \theta = 0, \end{aligned}$$

is $z^1 = 3$, $x = x^1 = (1.5, 0, 1.5)^T$, $\theta = \theta^1 = 0$. Using this, we first compute the right-hand side to the subproblem $\omega = 1$:

$$d_1 - B_1 x^1 = \begin{pmatrix} 9 \\ 15 \\ 9 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 1 \\ 3 & 2 & 1 \\ 4 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1.5 \\ 0.0 \\ 1.5 \end{pmatrix} = \begin{pmatrix} 6.0 \\ 9.0 \\ 1.5 \end{pmatrix}$$

and then solve the first subproblem:

$$\begin{aligned} &\text{Minimize} && 3y_{11} + 2y_{12} + y_{13} = w_1^1 \\ &\text{subject to} && 4y_{11} - y_{12} + y_{13} = 6.0 \\ &&& 3y_{11} + 2y_{12} + y_{13} = 9.0 \\ &&& y_{11} + y_{12} + y_{13} = 1.5 \\ &&& y_1 \geq 0 \end{aligned}$$

This problem is infeasible, so we set $\bar{z}^1 = +\infty$ and determine $L = \operatorname{argmin} \bar{z}^1 = 1$, or $\bar{z}^L = +\infty$.

Next we use the multipliers to create an infeasibility cut. The multipliers are:

$$\pi_1^1 = \begin{pmatrix} -0.2 \\ 1.0 \\ -2.2 \end{pmatrix}$$

Next we compute the infeasibility cut $G^1 x \geq g^1$ by computing

$$G^1 = (\pi_1^1)^T B_1 = (-6 \quad 4 \quad 1.4)$$

and

$$g^1 = (\pi_1^1)^T d_1 = -6.6$$

Next we set $\bar{z}^1 = +\infty$ and determine $L = \operatorname{argmin} \bar{z}^1 = 1$, or $\bar{z}^L = +\infty$.

The new Benders Restricted Master is:

$$\begin{aligned} &\text{Minimize} && x_1 + x_2 + x_3 + \theta = z \\ &\text{subject to} && x_1 + 2x_2 + 3x_3 = 6 \\ &&& 3x_1 + 2x_2 + x_3 = 6 \\ &&& -6x_1 + 4x_2 - 1.4x_3 \geq -6.6 \\ &&& x \geq 0, \theta = 0, \end{aligned}$$

The optimal solution to this is $\underline{z}^2 = 3.0$, $x^2 = (1.207792 \quad 0.584416 \quad 1.207792)$, $\theta^2 = 0$. Clearly $\underline{z}^2 + TOL < \bar{z}^L$ and therefore we continue with creating and solving the modified set of subproblems.

Set the iteration counter $k = 2$ and first compute the right-hand side to the subproblem for $\omega = 1$ as

$$d_1 - B_1 x^2 = \begin{pmatrix} 9 \\ 15 \\ 9 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 1 \\ 3 & 2 & 1 \\ 4 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1.207792 \\ 0.584416 \\ 1.207792 \end{pmatrix} = \begin{pmatrix} 6.0 \\ 9.0 \\ 3.545456 \end{pmatrix}$$

and then solve the subproblem:

$$\begin{aligned} &\text{Minimize} && 3y_{11} + 2y_{12} + y_{13} = w_1 \\ &\text{subject to} && 4y_{11} - y_{12} + y_{13} = 6.0 \\ &&& 3y_{11} + 2y_{12} + y_{13} = 9.0 \\ &&& y_{11} + y_{12} + y_{13} = 3.545456 \\ &&& y_1 \geq 0 \end{aligned}$$

This subproblem solves to optimality: $w_1^2 = 9.0$, $y_1 = (1.9091 \quad 1.6364 \quad 0.0)$ with the multipliers

$$\pi_1^2 = \begin{pmatrix} 0.0 \\ 1.0 \\ 0.0 \end{pmatrix}$$

Next we set up and solve the subproblem for $\omega = 2$. We first compute the right-hand side as

$$d_2 - B_2 x^2 = \begin{pmatrix} 3 \\ 15 \\ 15 \end{pmatrix} - \begin{pmatrix} 2 & -2 & 1 \\ 0 & 4 & 1 \\ 6 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1.207792 \\ 0.584416 \\ 1.207792 \end{pmatrix} = \begin{pmatrix} 0.545456 \\ 11.454544 \\ 6.545456 \end{pmatrix}$$

and then solve the subproblem:

$$\begin{aligned} &\text{Minimize} && 2y_{21} + 4y_{22} + 0y_{23} = w_2^2 \\ &\text{subject to} && 4y_{21} - y_{22} + y_{23} = 0.545456 \\ &&& 3y_{21} + 2y_{22} + y_{23} = 11.454544 \\ &&& y_{21} + y_{22} + y_{23} = 6.545456 \\ &&& y_2 \geq 0 \end{aligned}$$

This subproblem solves to optimality: $w_2^2 = 10.3636$, $y_2 = (1.9091 \quad 1.6364 \quad 0.0)$ with multipliers

$$\pi_1^2 = \begin{pmatrix} -2.0 \\ 1.0 \\ 0.0 \end{pmatrix}$$

The expected first-stage plus second-stage costs are:

$$\bar{z}^2 = c^T x^2 + p_1 w_1^2 + p_2 w_2^2 = 3 + 0.5 * 9 + 0.5 * 10.3636 = 12.6818$$

We compute the new upper bound by determining

$$L = \operatorname{argmin}\{\bar{z}^1, \bar{z}^2\} = \{+\infty, 12.6818\} = 12.6818$$

and therefore $\bar{z}^L = 12.6818$.

Next we compute the optimality cut $G^2 x + \theta \geq g^2$ by computing

$$G^2 = p_1(\pi_1^2)^T B_1 + p_2(\pi_2^2)^T B_2 = (-0.5 \quad 5.0 \quad 0.0)$$

and

$$g^2 = p_1(\pi_1^2)^T d_1 + p_2(\pi_2^2)^T d_2 = 12.0$$

The new Benders Restricted Master at the end of iteration 2 is:

$$\begin{aligned} &\text{Minimize} && x_1 + x_2 + x_3 + \theta = z \\ &\text{subject to} && x_1 + 2x_2 + 3x_3 = 6 \\ &&& 3x_1 + 2x_2 + x_3 = 6 \\ &&& -6x_1 + 4x_2 - 1.4x_3 \geq -6.6 \\ &&& -0.5 + 5x_2 + 0x_3 + \theta \geq 12.0 \\ &&& x \geq 0. \end{aligned}$$

The optimal solution to this is $z = \underline{z}^3 = 0.0$, $x^3 = (0.0 \ 3.0 \ 0.0)$, $\theta = \theta^3 = -3.0$. Clearly $\underline{z}^3 + TOL < \bar{z}^L = 12.6818$, and therefore we continue with creating and solving the modified set of subproblems.

Using this, we first compute the right-hand side to the subproblem for $\omega = 1$ as

$$d_1 - B_1x^3 = \begin{pmatrix} 9 \\ 15 \\ 9 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 1 \\ 3 & 2 & 1 \\ 4 & -1 & 1 \end{pmatrix} \begin{pmatrix} 0.0 \\ 3.0 \\ 0.0 \end{pmatrix} = \begin{pmatrix} 6.0 \\ 9.0 \\ 12.0 \end{pmatrix}$$

and then we solve the subproblem:

$$\begin{aligned} &\text{Minimize} && 3y_{11} + 2y_{12} + y_{13} = w_1^3 \\ &\text{subject to} && 4y_{11} - y_{12} + y_{13} = 6.0 \\ &&& 3y_{11} + 2y_{12} + y_{13} = 9.0 \\ &&& y_{11} + y_{12} + y_{13} = 12.0 \\ &&& y_1 \geq 0 \end{aligned}$$

This problem is infeasible. So we set $\bar{z}^3 = +\infty$ and hence we know that the current upper bound is unchanged with $L = 2$ and $\bar{z}^L = 12.6818$.

Next we use the multipliers to create an infeasibility cut. The multipliers are:

$$\pi_1^3 = \begin{pmatrix} -1/3 \\ 2/3 \\ 1 \end{pmatrix}$$

Next we compute the infeasibility cut $G^3x \geq g^3$ by computing

$$G^3 = (\pi_1^3)^T B_1 = (1.666667 \ -2.666667 \ 0.0)$$

and

$$g^3 = (\pi_1^3)^T d_1 = -4.0$$

The new Benders Restricted Master at the end of iteration 3 is:

$$\begin{aligned} &\text{Minimize} && x_1 + x_2 + x_3 + \theta = z \\ &\text{subject to} && x_1 + 2x_2 + 3x_3 = 6 \\ &&& 3x_1 + 2x_2 + x_3 = 6 \\ &&& -6x_1 + 4x_2 - 1.4x_3 \geq -6.6 \\ &&& -0.5 + 5x_2 + 0x_3 + \theta \geq 12.0 \\ &&& 1.666667x_1 - 2.666667x_2 + 0x_3 \geq -4.0 \\ &&& x \geq 0. \end{aligned}$$

The optimal solution to this is $z = \underline{z}^4 = 6.0$, $x^4 = (0.5714285 \ 1.857143 \ 0.5714285)$, $\theta^4 = 3.0$. Clearly $\underline{z}^4 + TOL < \bar{z}^L = 12.6818$ and therefore we continue with creating and solving the modified set of subproblems.

Using this, we first compute the right-hand side to the subproblem for $\omega = 1$ as

$$d_1 - B_1x^4 = \begin{pmatrix} 9 \\ 15 \\ 9 \end{pmatrix} - \begin{pmatrix} 1 & 1 & 1 \\ 3 & 2 & 1 \\ 4 & -1 & 1 \end{pmatrix} \begin{pmatrix} 0.5714285 \\ 1.857143 \\ 0.5714285 \end{pmatrix} = \begin{pmatrix} 6.0 \\ 9.0 \\ 8.0 \end{pmatrix}$$

and then we solve the subproblem:

$$\begin{aligned} & \text{Minimize} && 3y_{11} + 2y_{12} + y_{13} = w_1^4 \\ & \text{subject to} && 4y_{11} - y_{12} + y_{13} = 6.0 \\ & && 3y_{11} + 2y_{12} + y_{13} = 9.0 \\ & && y_{11} + y_{12} + y_{13} = 8.0 \\ & && y_1 \geq 0 \end{aligned}$$

This subproblem solves to optimality: $w_1^4 = 9.0$, $y_1 = (0.0 \quad 1.0 \quad 7.0)$ with multipliers

$$\pi_1^4 = \begin{pmatrix} 0.0 \\ 1.0 \\ 0.0 \end{pmatrix}$$

Next we set up and solve the subproblem for $\omega = 2$. We first compute the right hand side as

$$d_2 - B_2 x^4 = \begin{pmatrix} 3 \\ 15 \\ 15 \end{pmatrix} - \begin{pmatrix} 2 & -2 & 1 \\ 0 & 4 & 1 \\ 6 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0.5714285 \\ 1.857143 \\ 0.5714285 \end{pmatrix} = \begin{pmatrix} 5.0 \\ 7.0 \\ 11.0 \end{pmatrix}$$

and then solve the subproblem:

$$\begin{aligned} & \text{Minimize} && 2y_{21} + 4y_{22} + 0y_{23} = w_2^4 \\ & \text{subject to} && 4y_{21} - y_{22} + y_{23} = 5.0 \\ & && 3y_{21} + 2y_{22} + y_{23} = 7.0 \\ & && y_{21} + y_{22} + y_{23} = 11.0 \\ & && y_2 \geq 0 \end{aligned}$$

This subproblem solves to optimality: $w_2^4 = 4$, $y_2 = (0.0 \quad 1.0 \quad 7.0)$ with multipliers

$$\pi_1^4 = \begin{pmatrix} -1.1428571 \\ 0.7142857 \\ 0.4285714 \end{pmatrix}$$

The expected first-stage plus second stage costs are:

$$\bar{z}^2 = c^T x^4 + p_1 w_1^4 + p_2 w_2^4 = 3 + 0.5 * 9 + 0.5 * 4 = 9.5$$

We compute the new upper bound by determining

$$L = \operatorname{argmin}\{\bar{z}^1, \bar{z}^2, \bar{z}^3, \bar{z}^4\} = \{+\infty, +12.6818, +\infty, 9.5\} = 4$$

and, therefore, $\bar{z}^L = 9.5$.

Next we compute the optimality cut $G^4 x + \theta \geq g^4$ by computing

$$G^4 = p_1(\pi_1^4)^T B_1 + p_2(\pi_2^4)^T B_2 = (1.6428571 \quad 3.5714285 \quad 0.5)$$

and

$$g^4 = p_1(\pi_2^4)^T d_1 + p_2(\pi_2^4)^T d_2 = 14.3571426$$

The new Benders Restricted Master at the end of iteration 4 is:

$$\begin{aligned} & \text{Minimize} && x_1 + x_2 + x_3 + \theta = z \\ & \text{subject to} && x_1 + 2x_2 + 3x_3 = 6 \\ & && 3x_1 + 2x_2 + x_3 = 6 \\ & && -6x_1 + 4x_2 - 1.4x_3 \geq -6.6 \\ & && -0.5 + 5x_2 + 0x_3 + \theta \geq 12.0 \\ & && 1.666667x_1 - 2.666667x_2 + 0x_3 \geq -4.0 \\ & && 1.6428571x_1 + 3.5714285x_2 + 0.5x_3 + \theta \geq 14.3571426 \\ & && x \geq 0. \end{aligned}$$

The optimal solution to this is $\underline{z}^5 = 9.5$, $x^5 = (0.5714285 \quad 1.857143 \quad 0.5714285)$, $\theta^5 = 6.5$. Now $\underline{z}^5 + TOL > \bar{z}^L = 9.5$ and we stop and report the optimal values for x .

12.3.4 THEORY BEHIND THE ALGORITHM

Given x^k , let (x^k, y^k, \bar{z}^k) be the output of the k th iterate of the algorithm. Let \mathcal{C} be the set of feasible points satisfying the original problem (12.9). Denote by $\mathcal{M}(k)$ the set of all feasible points of the updated Benders Master with k cuts.

LEMMA 12.3 (Lower Bound on \bar{z}^k) *At each iteration k ,*

$$\bar{z}^k \geq \min_{\mathcal{C}} z. \quad (12.28)$$

Proof. Since (x^k, y^k, \bar{z}^k) is a feasible point in \mathcal{C} , then obviously

$$z^* = \min_{\mathcal{C}} z \leq \bar{z}^k \quad (12.29)$$

■

LEMMA 12.4 (Point with Smallest z) *The point in $\mathcal{M}(k-1)$ with smallest z , namely, (x^k, \underline{z}^k) , satisfies*

$$\underline{z}^{k-1} \leq \underline{z}^k = \min_{\mathcal{C}^k | x^k} z \leq \text{Min}_{\mathcal{C}} z \leq \bar{z}^L = \min_{i \leq k} \bar{z}^i. \quad (12.30)$$

Proof. It is clear, as we have noted earlier, that all the points on the lower boundary of \mathcal{C} are contained in the half-space of every cut including the latest

$$\text{Cut } (k): \quad z \geq (c - G^k)x + g^k. \quad (12.31)$$

Let P^k be any point on the lower boundary of \mathcal{C} and let $P^k = (x^k, y^k, \bar{z}^k)$. Given $x = x^k$, then $y_\omega = y_\omega^k \geq 0$ minimizes $f_\omega y_\omega$ subject to

$$D_\omega y_\omega = d_\omega - B_\omega x^k, \quad y_\omega \geq 0, \quad \text{for } \omega = 1, \dots, W \quad (12.32)$$

and therefore

$$\bar{z}^k = \min (z \mid x = x^k) = cx^k + \sum_{\omega=1}^W p_\omega (f_\omega y_\omega^k). \quad (12.33)$$

Then (x^k, \bar{z}^k) satisfies (12.31) with equality; see the proof of Theorem 12.2. ■

Definition (Almost Optimal First-Stage Decision):

$$\text{If } TOL \geq \bar{z}^L - \underline{z}^{k+1} \geq \bar{z}^L - z^*, \quad (12.34)$$

then \bar{z}^L , the first-stage cost plus minimum second-stage cost given $x = x^L$ is deemed as “close enough” to $\min z$ for $x = x^L$ to be declared an “almost optimal” first-stage decision.

LEMMA 12.5 (Optimal First-Stage Decision) Let $L = \operatorname{argmin}_{i \leq k} \bar{z}^i$.

$$\text{If } \underline{z}^{k+1} = \bar{z}^L, \text{ decision } x = x^L \text{ is optimal.} \quad (12.35)$$

Proof. The proof is obvious; see (12.30). ■

THEOREM 12.6 (Cut Chops off Part of the Feasible Region) If $\underline{z}^k < \bar{z}^k$, then Cut (k) “chops off” all points (x^k, \underline{z}^k) of the feasible region of $\mathcal{M}(k-1)$, implying that all the cuts differ from one another.

Proof. That each new cut chops off part of the feasible region is evident from Figure 12-1 if $\underline{z}^k < \bar{z}^k$ because the segment on the line $x = x^k$ between \underline{z}^k and \bar{z}^k lies in the feasible set of $\mathcal{M}(k-1)$ but not in that of $\mathcal{M}(k)$. ■

THEOREM 12.7 (Finite Termination) There are only a finite number of cuts when extreme π_ω^k are used to generate the cuts, implying that the iterative process terminates after a finite number of iterations.

Proof. To prove that the iterative process is finite and terminates in an optimal solution for some finite k with (x^k, \bar{z}^k) , we note this must be because π_ω^k is chosen from the finite class of dual extreme solutions of the subproblems. With only a finite set of π_ω^k to choose from, only a finite number of different cuts can be generated. When this finite number k is reached (if not before) $\underline{z}^{k+1} = \bar{z}^L$ (where $L = \operatorname{argmin}_{i \leq k} \bar{z}^i$), because if not, a new different cut would be generated contrary to the fact that no more cuts can be generated, establishing the theorem. ■

12.4 SOLVING STOCHASTIC TWO-STAGE PROBLEMS USING SAMPLING

12.4.1 OVERVIEW

When W is huge, it is no longer practical, given some $x = x^k$, to solve all the second-stage subproblems in order to determine the expected minimum second-stage costs

$$\theta^k = E[f_\omega y_\omega] = \sum_{\omega=1}^W p_\omega(f_\omega y_\omega^k), \quad (12.36)$$

where y_ω^k denotes the optimal second-stage decision for some ω given $x = x^k$. To see why W can be huge in practice, suppose an electric power system has 20 generators and 10 transmission lines, any one or combination of which could be non-operational. Then altogether there are $2^{20} \times 2^{10} = 2^{30}$ possibilities; i.e., there are over one billion possible states of the system. If we also consider the effects of variable demand patterns at different demand points due to weather conditions, then the possible number of cases to consider becomes truly astronomical!

Even though W typically is huge, we can still make progress using *sampling* to estimate the *expected minimum second-stage cost* given a first-stage decision x^k . We assume that the sampling procedure consists of independent trials with replacement. *Naive Sampling* can be used to estimate $\tilde{\theta}^k$ by *randomly sampling the ω proportional to p_ω* and averaging arithmetically over ω in the sample \mathcal{S} the observed minimum second-stage costs $f_\omega y_\omega^k$.

Instead of sampling ω proportional p_ω , $\sum p_\omega = 1$, $p_\omega > 0$, another way is to sample ω proportional to some $q_\omega \neq p_\omega$, where $\sum q_\omega = 1$, $q_\omega > 0$, and averaging over ω in the sample $(p_\omega/q_\omega)(f_\omega y_\omega^k)$, i.e., the minimum second-stage costs $(f_\omega y_\omega^k)$ weighted by (p_ω/q_ω) . If q_ω is chosen to be roughly proportional to $p_\omega(f_\omega y_\omega^k)$, it will sample more frequently (compared to Naive Sampling) those outcomes ω having extremely low probability p_ω but extremely high cost $(f_\omega y_\omega^k)$. When this is the case, the latter technique, called *Importance Sampling*, can often reduce by many orders of magnitude the size of the sample required to attain a given accuracy of estimate of θ^k .

12.4.2 NAIVE SAMPLING

For the discussion that follows, we assume the Naive Sampling procedure is used. According to statistical theory of sampling, which we review in Appendix A, an *unbiased estimator* of θ^k is:

$$\tilde{\theta}^k = \frac{1}{N} \sum_{\omega \in \mathcal{S}} f_\omega y_\omega^k, \quad (12.37)$$

where \mathcal{S} is a random sample of size N with replacements.

The *variance* of the sampled observations is:

$$\frac{1}{N} \sum_{\omega \in \mathcal{S}} (f_\omega y_\omega^k - \tilde{\theta}^k)^2, \quad (12.38)$$

where N is the sample size. However, if (12.38) is used to estimate the true variance of the population, $(\sigma^k)^2$, it can be shown to underestimate it. On the other hand,

$$(\tilde{\sigma}^k)^2 = \frac{1}{N-1} \sum_{\omega \in \mathcal{S}_o} (f_\omega y_\omega^k - \tilde{\theta}^k)^2 \quad (12.39)$$

can be shown to be an *unbiased estimator* of $(\sigma^k)^2$.

The expected value of the set of all means of size N is also θ^k and an unbiased estimate of the variance of all sample means of size N based on *random samples with replacements* of size N is

$$\frac{1}{N} (\tilde{\sigma}^k)^2 = \frac{1}{N(N-1)} \sum_{\omega \in \mathcal{S}_o} (f_\omega y_\omega^k - \tilde{\theta}^k)^2. \quad (12.40)$$

Notation: When $x = x^k$, we denote y_ω^k as an optimum y_ω given $x = x^k$. We denote the expected minimum second-stage costs by θ^k . When a sample \mathcal{S}_i is used to

estimate θ^k , we denote the estimated expected minimum second-stage costs by $\tilde{\theta}_i^k$. Thus

$$\theta^k = E[f_\omega y_\omega^k] = \sum_{\omega=1}^W p_\omega (f_\omega y_\omega^k) \quad \text{and} \quad \tilde{\theta}_i^k = \frac{1}{N} \sum_{\omega \in \mathcal{S}_i} f_\omega y_\omega^k \quad (12.41)$$

where \mathcal{S}_i is a random sample of size N of the ω s. In particular, on iteration k of the iterative process we are given $x = x^k$ and we generate only one corresponding random sample $\mathcal{S} = \mathcal{S}_k$ on iteration k , then, see (12.41), the estimated second-stage cost is $\tilde{\theta}_k^k$, and the estimated first-stage and expected second-stage costs are denoted $\tilde{z}_k^k = cx^k + \tilde{\theta}_k^k$.

12.4.3 SAMPLING METHODOLOGY

In applications of stochastic two-stage linear programs, as noted earlier, the number of scenarios W is typically huge, for example, $W > 10^6$ or $W > 10^9$. We assume that the Decomposition Algorithm (see Section 12.3.3) has been applied with the following replacements:

$$\begin{aligned} \sum_{\omega=1}^W p_\omega f_\omega y_\omega & \quad \text{by} \quad \frac{1}{N} \sum_{\omega \in \mathcal{S}_k} f_\omega y_\omega \\ \sum_{\omega=1}^W p_\omega \pi_\omega^k d_\omega & \quad \text{by} \quad \frac{1}{N} \sum_{\omega \in \mathcal{S}_k} \pi_\omega^k d_\omega \\ \sum_{\omega=1}^W p_\omega \pi_\omega^k B_\omega & \quad \text{by} \quad \frac{1}{N} \sum_{\omega \in \mathcal{S}_k} \pi_\omega^k B_\omega \end{aligned}$$

for iterations $1, \dots, k$. We assume, at iteration k , that the iterative process has been stopped because of Step 8 of Algorithm 12.1 (as modified for sampling) or because a preassigned maximum number k of iterations has been reached.

To measure how good this final solution x^k is, we determine a 95% *upper-bound estimate* for $z^k = cx^k + \sum_{\omega=1}^W p_\omega \theta_\omega^k$ and a 95% *lower-bound estimate* for $\text{Min } z$, meaning a 95% probability that the upper-bound estimate is higher than the true z^k and a 95% probability that the lower-bound estimate is lower than the true $\text{Min } z$. *The difference between these two bounds will be our measure of the “goodness” of the first-stage solution x^k .*

12.4.4 ESTIMATING UPPER BOUND z_{UB} FOR $\text{MIN } z$

An unbiased estimator of θ^k is

$$\tilde{\theta}^k = \frac{1}{N} \sum_{\omega \in \mathcal{S}_k} \theta_\omega^k, \quad (12.42)$$

where θ_ω^k is the minimum second-stage cost $f_\omega y_\omega$ given $x = x^k$. An unbiased “95%” upper-bound estimate of $cx^k + \theta^k$ is

$$cx^k + \theta^k + \lambda_{0.95} \tilde{\sigma}_{UB}^k, \tag{12.43}$$

where the area from $-\infty$ to $\lambda_{0.95}$ under the normal curve with mean 0 and standard deviation 1 is 0.95 and

$$(\tilde{\sigma}_{UB}^k)^2 = \frac{1}{N(N-1)} \sum_{\omega \in \mathcal{S}_k} (\theta_\omega^k - \tilde{\theta}^k)^2. \tag{12.44}$$

12.4.5 ESTIMATING LOWER BOUND z_{LB} FOR MIN z

To obtain a “95%” lower-bound estimate of Min z , we note that the objective value v of any feasible solution to the Dual of (12.9) provides a true lower bound $v \leq \text{Min } z$. Unfortunately because W is huge, we must estimate v by sampling. To insure that the lower-bound estimate is independent of the upper-bound estimate, we choose another independent random sample \mathcal{S}'_k of size N .

Dual of (12.9): Find max v, ρ, π_ω , such that:

$$\begin{aligned} \rho b + \sum_{\omega=1}^W p_\omega \pi_\omega d_\omega &= v \text{ (Max)} \\ \rho A + \sum_{\omega=1}^W p_\omega \pi_\omega B_\omega &\leq c \\ \pi_\omega D_\omega &\leq f_\omega, \text{ for } \omega = 1, \dots, W. \end{aligned} \tag{12.45}$$

One such feasible solution to (12.45) is to set $\pi_\omega = \pi_\omega^k$ for $\omega = 1, \dots, W$ where π_ω^k are the optimal dual multipliers to the second-stage problems given $x = x^k$. These satisfy the last set of inequalities of (12.45). To obtain $\rho = \rho^k$, substitute $\pi_\omega = \pi_\omega^k$ into the first two sets of inequalities of (12.45) and solve this single-stage linear program to determine ρ and v , a lower bound for Min z , namely,

Given $\pi_\omega = \pi_\omega^k$, find Max $v \leq \text{Min } z$, such that:

$$\begin{aligned} \rho b + \sum_{\omega=1}^W p_\omega \pi_\omega^k d_\omega &= v \text{ (Max)} \\ \rho A + \sum_{\omega=1}^W p_\omega \pi_\omega^k B_\omega &\leq c. \end{aligned} \tag{12.46}$$

Rather than optimizing the dual single-stage linear program (12.46) to obtain Max v , a lower bound for Min z , it turns out to be more convenient to approx-

imately optimize the primal of this dual problem:

$$\begin{aligned} &\text{Find } \xi, \text{ Min } z_{LB}, \text{ such that} \\ &\sum_{\omega=1}^W p_{\omega} \pi_{\omega}^k d_{\omega} + \left(c - \sum_{\omega=1}^W p_{\omega} \pi_{\omega}^k B_{\omega} \right) \xi = z_{LB} \quad (\text{Min}) \\ &A\xi = b \\ &\xi \geq 0. \end{aligned} \quad (12.47)$$

By duality $\text{Min} z_{LB} = \text{Max } v \leq \text{Min } z$.

▷ **Exercise 12.6** State conditions that guarantee the existence of ρ satisfying (12.46). What do these imply about its dual (12.47)? Conversely, will these conditions always be satisfied if the second-stage problems of (12.9) are always feasible with finite minima whatever be x^k satisfying $Ax = b$, $x \geq 0$?

Denoting the constant term and coefficients of the objective of (12.47) by

$$\gamma_o = \sum_{\omega=1}^W p_{\omega} \pi_{\omega}^k d_{\omega}, \quad \gamma_j = \sum_{\omega=1}^W p_{\omega} (c_j - \pi_{\omega}^k B_{\omega j}), \quad \text{for } j = 1, \dots, n, \quad (12.48)$$

where $B_{\omega j}$ denotes the j th column of the matrix B_{ω} , we use an independent random sample \mathcal{S}'_k of size N to infer a distribution of possible values of the constant term γ_o and the coefficients γ_j for $j = 1, \dots, n$ of the objective of (12.47) where the *true means* are (12.48).

The sampled means in (12.49) are unbiased estimates of the true means (12.48) based on a sample \mathcal{S}'_k of size N :

$$\tilde{\gamma}_o = \frac{1}{N} \sum_{\omega \in \mathcal{S}'_k} p_{\omega} \pi_{\omega}^k d_{\omega}, \quad \tilde{\gamma}_j = \frac{1}{N} \sum_{\omega \in \mathcal{S}'_k} (c_j - \pi_{\omega}^k B_{\omega j}), \quad \text{for } j = 1, \dots, n. \quad (12.49)$$

For sample sizes sufficiently large (say $N > 200$), the “likelihood” that the observed vector of sample means $(\tilde{\gamma}_o, \tilde{\gamma}_1, \dots, \tilde{\gamma}_n)$ from their vector of true means $(\gamma_o, \gamma_1, \dots, \gamma_n)$ can be reasonably assumed to follow very closely a *multivariate normal distribution* centered at the origin. Our immediate goal is to calculate an unbiased estimate of the true variance-covariance matrix based on the sampled observations about the sample means. If we denote the vector of deviations of the sampled observations from their sampled means by

$$\delta_{\omega o} = \pi_{\omega}^k d_{\omega} - \tilde{\gamma}_k, \quad \delta_{\omega j} = (c_j - \pi_{\omega}^k B_{\omega j}) - (c_j - \tilde{\gamma}_j) \quad \text{for } j = 1, \dots, n, \quad (12.50)$$

and let M be the matrix whose rows are defined by $B_{\omega \bullet} = [\delta_{\omega o}, \delta_{\omega 1}, \dots, \delta_{\omega n}]$, for $\omega \in \mathcal{S}'_k$, then the estimated *variance-covariance* matrix of the multivariate normal distribution about the sampled means (12.49) is $(1/N)M^T M$. However, this estimate is biased. An unbiased estimate of the variance-covariance matrix about the true means (12.48) is $(1/(N-1))M^T M$. Finally an unbiased estimate of

the variance-covariance matrix of the means of samples of size N about their true means, based on a sample S'_k , is:

$$\frac{1}{N(N-1)}M^TM. \tag{12.51}$$

If we write the terms of the objective (12.47) plus a correction term $\delta_o, \delta_1, \dots, \delta_n$ so that the constant term and coefficients agree with their true values displayed in (12.47)

$$\begin{aligned} \frac{1}{N} \sum_{\omega \in S_k} \pi_\omega^k d_\omega + \delta_o + \sum_{j=1}^n \left(c_j + \sum_{\omega \in S_k} \pi_\omega^k B_{\omega j} + \delta_j \right) \xi_j = z_{LB} \quad (\text{Min}) \\ A\xi = b \\ \xi \geq 0, \end{aligned} \tag{12.52}$$

then we are asserting that the vector $\delta = (\delta_o, \delta_1, \dots, \delta_n)$ is a random vector drawn from the multi-variate normal distribution centered at the origin with estimated variance-covariance matrix $(1/N(N-1))M^TM$. For each random choice of the vector $(\delta_o, \delta_1, \dots, \delta_n)$, the linear program (12.52) is optimized and a minimum value z_{LB} is obtained. Thus z_{LB} is a random variable that depends on the choice of the vector $(\delta_o, \delta_1, \dots, \delta_n)$ drawn from a multivariate normal distribution with variance-covariance matrix based on sample S'_k . It is recommended in practice that at least $S'_k \geq 200$ independent random choices of the vector $(\delta_o, \delta_1, \dots, \delta_n)$ from multivariate normal distribution be made, and for each such choice the one-stage LP (12.52) be optimized. This results in an empirical distribution of S'_k estimates of z_{LB} . These 200 or more z_{LB} s are next ranked from low to high and the lower bound z_{LB} for Min z is chosen as that z_{LB} that is located 5 percentile points from the bottom (which is at the 95% probability point according to the empirical distribution that $z_{LB} \leq \text{Min } z$).

▷ **Exercise 12.7** Find a way to generate a random point of a multivariate distribution.

12.5 USE OF IMPORTANCE SAMPLING

One weakness in using the “naive” sampling procedure is that it may fail to sample any of the rare but high-cost catastrophic events ω having a very low probability p_ω of happening. Let q_ω be *any distribution* of ω , i.e., $\sum_{\omega=1}^W q_\omega = 1, q_\omega \geq 0$. If we rewrite

$$\theta^k = \sum_{\omega=1}^W q_\omega \left(\frac{p_\omega}{q_\omega} f_\omega y_\omega^k \right), \quad \sum_{\omega=1}^W q_\omega = 1, q_\omega \geq 0, \tag{12.53}$$

and then sample ω proportional to q_ω and average $(p_\omega/q_\omega)f_\omega y_\omega^o$ over a randomly drawn sample S_o^g of size N drawn from the distribution q_ω , then

$$\tilde{\theta}^k = \frac{1}{N} \sum_{\omega \in S_o^g} \left(\frac{p_\omega}{q_\omega} \right) f_\omega y_\omega^o \tag{12.54}$$

is also an unbiased estimator of θ^o .

Question: Is there some way to choose q_ω so that sampling ω proportional to q_ω to estimate θ^k is superior to estimating θ^k using p_ω in the sense that for some fixed size N , $\tilde{\theta}^k$ has a smaller variance? Later we will discuss classes of problems where the answer to this question is *yes*.

12.5.1 CRUDE (NAIVE) MONTE CARLO METHODS

The main computational difficulty in solving the two-stage stochastic linear program (12.9) is the evaluation of the expected cost of the second stage when W is huge. If so, it will not be feasible to evaluate all the $\omega = 1, \dots, W$ terms of the expected-value expression. The best, numerically efficient way to approximate expected values (which are, by definition, higher-dimensional multiple integrals or sums), according to expert numerical analysts, is by *Monte Carlo* techniques. These use a random sampling of the ν -dimensional domain of the function being integrated to approximate its expected value. The computational effort is often relatively independent of ν , the dimension of the space, whereas the computational effort of classical techniques that subdivide the ν -dimensional sampling space grows proportional to the number of subdivisions that (in turn) grows with the power of ν . Note that ν denotes the dimension of the space while N is the number of discrete points ω in the sample.

Let us suppose that our stochastic linear program (12.2) is a very complex model of an electric power distribution system. Many of the possibly hundreds or thousands of coefficients and constant terms of the second period are known with certainty, but many others are dependent on a small number, ν , of *independent random variables* $V = (V_1, V_2, \dots, V_\nu)$.

For example V_i for $i = 1, \dots, 30$ measures the repair state of the i th electric power generator. If $V_i = 1$ for generator i , it means it is in service, and if $V_i = 0$ it means it is out of service with known probabilities. In addition, assume there are two other random variables V_{31} that measure the annual rainfall and V_{32} that measure the prices of oil, and that V_{31} and V_{32} each can take on five values with known probabilities. Assuming independence of the random variables, the number of scenarios W that model (12.2) has in this case in the second period is $W = 2^{30} \times 5 \times 5 > 26$ billion, a very large number.

Instead of sampling from the more than 26 billion scenarios as if it were one long sequence, we choose ω by independently random sampling each of the small number $\nu = 32$ distributions V_i and using these to evaluate the $B_\omega(i, j)$, $D_\omega(i, j)$, $d_\omega(i)$.

Suppose that a point ω in a sample space Ω results in the ω outcome of a stochastic vector $V = (V_1, V_2, \dots, V_\nu)^T$. We are interested in estimating the expected value of some function $\Psi(V)$ over the sample space:

$$\bar{\psi} = E[\Psi(V)] = \sum_{\omega \in \Omega} p_\omega \psi_\omega, \quad (12.55)$$

where $p_\omega = p(v_1^\omega, v_2^\omega, \dots, v_n^\omega)$ is the joint probability distribution of the point ω in the sample space. For example, ψ_ω could be the minimum second-stage costs of subproblem ω given some first-stage decision x , and some random outcome of the stochastic vector V , i.e.,

$$\psi_\omega = \text{Min } f_\omega y_\omega \quad \text{subject to} \quad D_\omega y_\omega = d_\omega - B_\omega x, \quad y_\omega \geq 0. \quad (12.56)$$

Suppose we take a sample \mathcal{S} of N points (scenarios), v^ω , such that the likelihood of choosing ω is proportional to their joint probability distribution (or mass) function $p_\omega = p(v_1^\omega, v_2^\omega, \dots, v_n^\omega)$. An unbiased estimator of the mean $\bar{\psi}$ of $\Psi(V)$ is:

$$\tilde{\psi} = \frac{1}{N} \sum_{\omega \in \mathcal{S}} \psi_\omega. \quad (12.57)$$

The variance σ^2 for the distribution p_ω is

$$\sigma^2 = \text{Var}[\Psi(V)] = \sum_{\omega \in \Omega} p_\omega (\psi_\omega - \bar{\psi})^2. \quad (12.58)$$

An unbiased estimator of σ^2 is

$$\tilde{\sigma}^2 = \frac{1}{N-1} \sum_{\omega \in \mathcal{S}} (\psi_\omega - \tilde{\psi})^2. \quad (12.59)$$

The variance of the *means* of samples of size N is σ^2/N . Therefore its unbiased estimator is

$$\frac{1}{N} \tilde{\sigma}^2 = \frac{1}{N(N-1)} \sum_{\omega \in \mathcal{S}} (\psi_\omega - \tilde{\psi})^2. \quad (12.60)$$

By the Central Limit Theorem, the distribution of such means $\tilde{\psi}$, as sample size $N \rightarrow \infty$, tends to the normal distribution about the true mean with variance σ^2 . In practice, the distribution of $\tilde{\psi}$ is approximately normal for moderate size N , say N greater than 200. If it turns out that the unbiased sample variance (estimate of the standard error squared) is too high, the sample size N will need to be increased until the error of the estimate is acceptable.

This approach for approximating multiple integrals and summations is used to estimate the expected minimum second-stage costs $\theta^k = \psi$ and their variances, and to estimate the expected values g^k, G^k of Cut (k) of the iterative algorithm, and to estimate the variances of g^k or G^k .

12.5.2 MONTE CARLO METHODS USING IMPORTANCE SAMPLING

In practice the Crude Monte Carlo approach described earlier often has a very slow rate convergence to the normal distribution as sample size $N \rightarrow \infty$, namely, of the order $N^{-1/2}$. (*Note:* Each evaluation of a sample point, in our case, involves

the optimizing of the subproblem ω .) This makes it very desirable to develop and apply inexpensive *variance reduction* techniques for the estimation of $\bar{\psi}$, the expected value of $\Psi(V)$. We shall describe *importance sampling*, which is a classical variance-reduction technique for increasing the efficiency of Monte Carlo techniques. We then apply the idea to reducing the error of approximation of the expected value of the second-stage cost θ^k given the first-stage decision $x = x^k$, and to estimate how far $cx^k + \theta^k = z^k$ is from $\text{Min } z$ of the stochastic linear program.

Importance sampling changes the sampling procedure so that rare events that have catastrophic costs are sampled with greater frequency. Thus, for example, to evaluate the integral $\int_{-\infty}^{\infty} f(x)dx$, the method of importance sampling chooses a probability density $q(u)$ and then evaluates the equivalent integral

$$\int_{-\infty}^{\infty} f(x)dx = \int_{-\infty}^{\infty} \left[\frac{f(u)}{q(u)} \right] q(u)du = E_q [f(U)/q(U)] \quad (12.61)$$

where U is a random variable with probability density function q . The probability density function q is chosen so that it is approximately proportional to $|f(u)|$, i.e., q is large in regions where $|f(u)|$ is large. Thus, the procedure is to sample from the distribution q of U and to estimate the integral $\int_{-\infty}^{\infty} f(x)dx$ as the sample mean of $f(U)/q(U)$.

- ▷ **Exercise 12.8** Suppose that $f(u) > 0$ for all u and $q(u) = \lambda f(u)$ where $\lambda > 0$ is a fixed constant chosen so that $\int_{-\infty}^{\infty} q(u)du = 1$. Prove that $\int_{-\infty}^{\infty} f(x)dx = \lambda$; i.e., knowing the value of λ is the same as determining the value of the integral $\int_{-\infty}^{\infty} f(x)dx$.

- ▷ **Exercise 12.9** Suppose $q(u) = \lambda f(u)$ is a probability distribution and the expected value of $f(U)/q(U)$ is estimated by a random sample drawn from the distribution $q(u)$. Show that, regardless of what random sample of $f(U)/q(U)$ is taken and then averaged, the sample mean is λ and therefore the sample estimates the $\int_{-\infty}^{\infty} f(x)dx$ with standard error estimate $\sigma = 0$.

- ▷ **Exercise 12.10** Suppose $f(u) > 0$ for all u and $q(u)$ is “roughly proportional” to $f(u)$. Show that the ratio $R(u) = f(u)/q(u)$ satisfies $(1 - \epsilon)\lambda \leq R(u) \leq (1 + \epsilon)\lambda$ for some $0 < \epsilon < 1$, $\lambda > 0$. Prove that the Importance Sampling will estimate the integral divided by λ with an error $< \epsilon$.

- ▷ **Exercise 12.11** Suppose the likelihood of a major outage of an electric power system is 1/1000, i.e., one day in 1000 days, but the social cost is enormous when it does happen. On a typical day, when it does not happen, the social cost is zero. Demonstrate why it is better to use importance sampling instead of crude Monte Carlo sampling.

In general, we are interested in computing

$$\bar{\theta} = \sum_{\omega \in \Omega} p(\omega)F(\omega) \quad (12.62)$$

where $F(\omega) = f_\omega y_\omega$ and $p(\omega)$ is the distribution of the random vector V . By choosing a distribution $q(\omega)$ we obtain

$$\bar{\theta} = \sum_{\omega \in \Omega} \frac{p(\omega)F(\omega)}{q(\omega)} q(\omega) = E_q \left[\frac{p(U)F(U)}{q(U)} \right] \quad (12.63)$$

where $U = (U_1, U_2, \dots, U_\nu)^T$ is a random vector with distribution q . We approximate $\bar{\theta}$ by random sampling from the distribution q to obtain an unbiased estimator of $\bar{\theta}$ by

$$\tilde{\theta} = \frac{1}{N} \sum_{j=1}^N \frac{p(u^j)F(u^j)}{q(u^j)}. \quad (12.64)$$

The unbiased estimator of the variance of θ is now:

$$\begin{aligned} \text{Var}_q[\tilde{\theta}] &= \frac{1}{N-1} \sum_{j=1}^N \left(\frac{p(u^j)F(u^j)}{q(u^j)} - \tilde{\theta} \right)^2 q(u^j) \\ &= \frac{1}{N-1} \left(E_q \left[\frac{p(U)^2 F(U)^2}{q(U)^2} \right] - \tilde{\theta}^2 \right). \end{aligned} \quad (12.65)$$

We now address the question of how to choose q so that $\text{Var}_q[\tilde{\theta}]$ is as small as possible for a given sample size N . Assuming $F(u) \geq 0$ the obvious optimal choice is

$$q(u) = q^*(u) \quad \text{where} \quad q^*(u) = \frac{F(u)p(u)}{\sum_{\omega \in \Omega} F(\omega)p(\omega)}. \quad (12.66)$$

because (see Exercises 12.8 and 12.9) with such a choice of q , the variance $\text{Var}_q[\tilde{\theta}]$ is zero. Using $q(u) = q^*(u)$ from (12.66) in (12.64) generates a $\tilde{\theta}$ that is a perfect estimate of the expected value of θ , because in this case (12.64) reduces to

$$\theta = \sum_{\omega \in \Omega} F(\omega)p(\omega).$$

Moreover, this perfect estimate is one that could have been obtained with exactly one observation $N = 1$! Unfortunately, while this ideal optimal choice is perfect, it is useless because, as we have already seen, the denominator of $q^*(u)$ in equation (12.66) is *the very quantity* that we are trying to estimate in the first place. Nevertheless, this observation suggests a good heuristic for determining q .

1. Choose a $q(\omega)$ that is roughly proportional to $|p(\omega)F(\omega)|$; and
2. at the same time choose a $q(\omega)$ that permits carrying out the calculations efficiently.

Two possible ways to choose a q that have been found to be very efficient for certain classes of important applications and requires only the evaluation of one-dimensional integrals or summations (which in general requires substantially less computational effort than evaluating a general multidimensional one).

Multiplicative Assumption

First, let us assume that $F(\omega) = F(V_1^\omega, V_2^\omega, \dots, V_\nu^\omega)$ is *roughly* multiplicative in its arguments, namely:

$$F(\omega) = F(V_1^\omega, V_2^\omega, \dots, V_\nu^\omega) \approx F_1(v_1^\omega)F_2(v_2^\omega) \cdots F_\nu(v_\nu^\omega). \quad (12.67)$$

Obviously the further away $F(\omega)$ is from being multiplicative, the rougher will be the approximation q and the higher will be the variance of the estimator of θ . Under this assumption, we choose the distribution $q(\omega)$ to also be multiplicative

$$q(\omega) = \left(\frac{F_1(v_1^\omega)p_1(v_1^\omega)}{\bar{F}_1} \right) \left(\frac{F_2(v_2^\omega)p_2(v_2^\omega)}{\bar{F}_2} \right) \cdots \left(\frac{F_\nu(v_\nu^\omega)p_\nu(v_\nu^\omega)}{\bar{F}_\nu} \right) \quad (12.68)$$

where $\bar{F}_i = E[F_i(v_i^\omega)]$. The quantity \bar{F}_i (depending on the application) is estimated by using either calculus to perform ν independent one-dimensional integrations or by Monte Carlo sampling of the one-dimensional distributions V_i . Once q is defined in this way, sample points v^ω are chosen by independently choosing components V_i according to their marginal distributions $F_i(v_i)p_i(v_i)/\bar{F}_i$. An estimate for θ is obtained as the arithmetic mean of $F_j(v_j)p_j(v_j)/F_j$ for $j = 1, \dots, N$.

Additive Assumption

It turns out that in certain applications, such as the calculation of financial portfolios and electric power distribution, a multiplicative approximation is not as good for integrating the function that measures cost as the one that is roughly additive in its arguments. We are assuming here that $F(\omega)$ is *roughly additive* of the form:

$$F(\omega) \approx \sum_{i=1}^{\nu} F_i(v_i^\omega) \quad (12.69)$$

Specifically, we are assuming that $q(\omega)$ takes the additive form:

$$q(\omega) = \frac{p(\omega) \sum_{i=1}^{\nu} F_i(v_i^\omega)}{\sum_{\omega \in \Omega} p(\omega) \sum_{k=1}^d F_k(v_k^\omega)} = p(\omega) \sum_{i=1}^{\nu} \left(\frac{\bar{F}_i}{\sum_k \bar{F}_k} \right) \left(\frac{F_i(v_i^\omega)}{\bar{F}_i} \right). \quad (12.70)$$

where $\bar{F}_i = E[F_i(v_i^\omega)]$. The quantity \bar{F}_i is once again easily estimated by sampling the marginal distribution of V_i . Finally we can write

$$q(\omega) = \sum_{i=1}^{\nu} \left(\frac{\bar{F}_i}{\sum_k \bar{F}_k} \right) \left(\frac{p_i(v_i^\omega)F_i(v_i^\omega)}{\bar{F}_i} \prod_{k \neq i} p_k(v_k^\omega) \right). \quad (12.71)$$

The expected value $E[Z_2(V)] = z_2 = \theta$ then becomes

$$\theta = \sum_{i=1}^{\nu} \left(\frac{\bar{F}_i}{\sum_k \bar{F}_k} \right) E_i \left[\frac{c(Y)p(Y)}{q(Y)} \right], \quad (12.72)$$

where E_i means that the component v_i^j of sample point v^j is to be independently sampled according to the marginal distribution of $q_i(v_i^\omega)$, i.e.,

$$q_i(v_i^\omega) = \frac{p_i(v_i^\omega)F_i(v_i^\omega)}{\bar{F}_i} \quad (12.73)$$

and all other components $j \neq i$ according to the marginal distribution $p_j(v_j^\omega)$. Note that θ is these expectations E_i in (12.72) weighted by $\bar{F}_i / \sum_j \bar{F}_j$. Thus, an estimate $\bar{\theta}$ of θ can be obtained when each E_i is itself estimated by sampling one-dimensional distributions.

The extent of variance reduction clearly depends on how good the true cost surface can be represented by an additive representation; if the fit is poor then the variance estimate of the mean of the sample will be high.

Estimation of $F_i(\omega_i)$

The quantities $F_i(v_i^\omega)$ can be estimated by evaluating the cost function on a relatively small lattice of points, namely, a set of lattice points along ν -coordinate directions:

$$F_i(v_i^\omega) \approx F(\tau_1, \dots, \tau_{i-1}, v_i^\omega, \tau_{i+1}, \dots, \tau_d) - F(\tau_1, \dots, \tau_{i-1}, \tau_i, \tau_{i+1}, \dots, \tau_d) \quad (12.74)$$

where the values $\tau_1, \tau_2, \dots, \tau_\nu$ are arbitrarily chosen at some fixed set of values. This determines the quantities up to an additive constant. We can get rid of this additive constant by writing

$$F(\omega) = F(\tau_1, \tau_2, \dots, \tau_\nu) + \Delta F(\omega).$$

Now the new function $\Delta F(\omega)$ is again of additive form but has the advantage that we know a priori that we may take $\Delta F_i(\tau_i) = 0$ and thus eliminate the additive constant entirely.

12.6 NOTES & SELECTED BIBLIOGRAPHY

The incorporation of uncertainty into linear programs was proposed independently by Dantzig [1955a] and Beale [1955a]. From then on various individuals have tried to extend the methods of linear programming to handle the problem of optimizing, in some defined sense, problems whose parameters are not known with certainty. Early references are Dantzig & Madansky [1961], Ferguson & Dantzig [1956], and Madansky [1959]. Over the years, different approaches have been used to attack such problems. See for example, Birge [1985a,b], Birge & Holmes [1992], Birge & Wallace [1988], Birge & Wets [1986, 1987], Dantzig & Infanger [1992a], Ermoliev [1983], Frauendorfer [1988], Frauendorfer & Kall [1988], Higle & Sen [1991], Kall [1979], Pereira, Pinto, Oliveira, & Cunha [1989], Rockafellar & Wets [1989], Ruszczyński [1986], Van Slyke & Wets [1969], and Wets [1984]. A survey of different ways to solve stochastic linear programs can be found in Ermoliev & Wets [1988] and an introduction to stochastic programming can be found in Birge & Louveaux [1997].

The *two-stage* case was first studied by Dantzig [1955a, 1963] and subsequently developed by Van Slyke & Wets [1966, 1969] and Wets [1984]. The solution method described in this Chapter began with Birge [1980] and Dantzig, [1982a] followed by major studies by Abrahamson [1983], Wittrock [1983], and Scott [1985]. It is based on Benders Decomposition (see Benders [1962] and Geoffrion [1970]); and using Importance Sampling based on Dantzig & Glynn [1990], and Glynn & Iglehart [1989]. The discussion of importance sampling applied to stochastic linear programs presented in this chapter also is based on these references. This approach has turned out to be very powerful in practice; see Infanger [1991] and Dantzig & Infanger [1992a], which report on the remarkable computational results obtained for several large-scale problems with up to 52 stochastic parameters. These stochastic problems, if reexpressed in standard linear programming format, could each have several billion constraints. The justification of the lower-bound estimates of the confidence interval have been criticized on theoretical grounds. However, the theory presented in this chapter provides a theoretical way to validate these earlier lower-bound estimates.

Dantzig & Infanger [1993] show how to apply the concepts of stochastic linear programs to portfolio optimization. In Dantzig & Glynn [1990] an extension of this approach for the multistage problem is proposed using parallel processors. For details on planning under uncertainty, see Infanger [1994]. See also Entriken [1989] for decomposition of linear programs using parallel computing.

Berry-Esséen (in Hall [1985]) gives upper bounds on the rates of convergence based on the Central Limit Theorem. If the algorithm described in this chapter terminates with an approximation far from optimal, the only remedy is to increase the sample size and try again. Morton [1993] develops a theory of augmenting the original sample and trying again.

Experimental results for electric power facilities by Nakayama (reported in Dantzig, Glynn, Avriel, Stone, Entriken, & Nakayama [1989]), based on the additive approach described in this chapter, showed importance sampling to be very effective. It turned out that a sample size $1/20000$ smaller was required to obtain the same-size confidence interval with the same degree of confidence of covering the true minimum value using importance sampling than would have been the case using “crude” sampling.

For Monte Carlo approaches to computing multiple integrals or multiple sums, see Davis & Rabinowitz [1984] and Deák [1988]. For a description of Monte Carlo Sampling Techniques, see Hammersly & Handscomb [1964]. An extensive review of methods of approximation, together with a list of references, can be found in Birge & Wets [1989] and Chapter 1 of Ermoliev & Wets, Eds. [1988].

A technique proposed by other researchers for computing the approximate integrals or sums for the continuous two-stage case provides upper and lower bounds by discretizing the sample space Ω into cells and summing the function values at representative points within the cells over all cells; see, for example, Birge [1985a], Birge & Wets [1986], Frauendorfer & Kall [1988], Huang, Ziemba, & Ben-Tal [1977], Kall & Stoyan [1982], and Kall & Wallace [1994]. In these cases, lower bounds are obtained easily by applying Jensen’s inequality (see, for example, Kall & Wallace [1994]). Upper bounds, however, require an exponential number of function evaluations with respect to the dimension d of the sample space Ω . Birge & Wets [1989] proposed a scheme for obtaining upper bounds that requires solving $O(m^2)$ linear programs instead. Their method replaces partitioning into cells by a method that seeks out an approximation using a small number of “positive” basis representations that span the space of columns associated with the second stage. Another technique for doing the approximations is to sample from Ω randomly and to use sample information to

guide the optimization algorithm. Among these methods are the stochastic quasi-gradient methods of Ermoliev [1988] and Gaivoronski [1988]. Their methods provide asymptotic convergence in the continuous case as the size of the sample $s \rightarrow \infty$, but fall short of providing a practical way to compute the accuracy of the bounds.

12.7 PROBLEMS

- 12.1 Suppose that in tomato season there are three canneries under one management that ship cases of canned tomatoes to five warehouses. The number of cases processed at each cannery during the tomato season is known in advance as shown in the table below together with the cost to ship per case to each of the warehouses.

Canneries	Availability of Cases	Shipping Cost (\$/case) to					Dump
		Warehouse					
		a	b	c	d	e	
1	50,000	0.9	2.0	1.8	1.7	2.5	1.0
2	75,000	0.6	1.6	1.4	1.8	2.5	1.0
3	25,000	2.7	1.8	1.5	1.0	0.9	1.0

The last column is the cost per case of dumping unshipped tomatoes. The seasonal demand, however, at each of the warehouses is *uncertain*. The probability distribution of demand is shown in the table below:

Warehouse	Demand at Probability		
	.15	.55	.30
a	15,000	20,000	30,000
b	16,000	20,000	28,000
c	17,000	20,000	26,000
d	18,000	20,000	24,000
e	19,000	20,000	22,000

Cases left over at the end of the season cannot be stored until next year because the food in the cans will spoil. They must be shipped to the dump at a loss of \$1 per case. Failure to supply all of the warehouses demands is penalized at \$0.25 per case, the discounted estimated loss of all future sales. (Turning a customer away runs the risk that the customer will become the customer of another supplier.) What shipping schedule will optimize the sum of total shipping cost plus expected net revenues? Solve the problem in the following ways:

- Formulate this as an equivalent deterministic linear program (EQ-LP). How many equations does this EQ-LP have? Solve this EQ-LP using any available software.
- Solve EQ-LP using the method of Section 12.3.
- Solve the problem using Benders Decomposition with crude Monte Carlo methods.
- Solve the problem using Benders Decomposition with Monte Carlo methods using importance sampling.

- 12.2 A farm comprises 240 acres of cropland. The acreage to be devoted to corn production and the acreage for oats production are the decision variables. Profit per acre for corn production and oats production under varying climate conditions are shown with probabilities below:

Climate	Prob	Profit (\$/acre)	
		Corn	Oats
Poor	0.20	15	20
Average	0.55	40	30
Good	0.25	55	40

- An additional resource restriction is that the total labor hours available during the production period is 320. Each acre of land in corn production uses 2 hours of labor during the production period, whereas production of oats requires only 1 hour. Formulate and solve an LP that maximizes the expected profit. Compare your solution with the one obtained using Bender's decomposition.
- 12.3 Birge & Louveaux [1997]. Northam Airlines is trying to decide how to partition a new plane into economy-, business-, and first-class seats for its Chicago-Detroit route. The plane can seat 200 economy-class passengers. A section can be partitioned off for first-class seats but each of these seats takes the space of 2 economy-class seats. A business class section can also be included, but each of these seats takes as much space as 1.5 economy-class seats. The profit on a first class ticket is, however, three times the profit of an economy ticket. A business-class ticket has a profit of two times an economy ticket's profit. Once the plane is partitioned into these seating classes, it cannot be changed. Northam knows, however, that the plane will not always be full in each section. They have decided that three scenarios will occur with about the same frequency: (1) weekday morning and evening traffic, (2) weekend traffic, and (3) weekday midday traffic. Under Scenario 1, they think they will have a demand for 20 first-class tickets, 50 business-class tickets, and 209 economy tickets. Under Scenario 2, these figures are 10, 25, and 175. Under Scenario 3, they are 5, 10, and 150. You can assume they cannot sell more tickets than seats in each of the sections. (In practice, airlines allow overbooking and have passengers with reservations who do not appear for the flight (no-shows).) The problem of determining how many passengers to accept under these circumstances is part of the field called *yield management*. For one approach to this problem of yield management, see Brumelle & McGill [1993].
- 12.4 A grape grower has just purchased 1,000 acres of vineyards. Due to the quality of the soil and the excellent climate in the region, he can sell all that he can grow of cabernet sauvignon, chardonnay, and sauvignon blanc grapes. He would like to determine how much of each variety to grow on the 1,000 acres, given various costs, profits, and manpower limitations, as shown in Table 12-1. The probabilities of bad, average, and good weather are 0.3, 0.5, and 0.2, respectively. Suppose he has a budget of \$100,000 and staff available to provide 8,000 man-days.
- Formulate the problem as a linear program.
 - Solve it using the DTZG Simplex Primal (*Linear Programming 1*) software option.

Variety	Man-days per acre	Cost \$/acre	Profit (\$/acre)		
			Bad	Average	Good
Cabernet Sauvignon	20	115	30	70	80
Sauvignon Blanc	10	90	40	50	60
Chardonnay	15	200	70	120	130

Table 12-1: Data for the Grape Grower's Dilemma

Week	Demand			Production Limit	Production cost (\$/set)	Storage cost (\$/set)
	Low	Average	High			
1	2	4	6	10	20	1
2	4	6	7	25	30	1
3	7	10	14	20	25	1
4	5	8	10	4	40	N/A

Table 12-2: Data for Dinner Set Production Schedule

- (c) Being curious about other possibilities in the future, the grape grower would like to know whether he should grow the Merlot variety of grapes, which requires 12 man-days/acre, cost \$80 per acre, and produces a profit of \$45/acre, \$55/acre, and \$65/acre depending on whether the weather is bad, average, or good. Without rerunning the problem, determine whether it makes sense to try to grow Merlot. If it does make sense, re-solve the linear program with the new grape variety included in the formulation.
- (d) In order to obtain some initial cash, the grape grower decides to sell futures (at a lower profit) of the yield from 25 acres of Sauvignon Blanc and 150 acres of Cabernet Sauvignon grapes. How does this change the optimal solution?
- 12.5 Your wife has recently taken a ceramics class and discovered that she has a talent for making elegant dinner sets. A specialty store around the corner from the class has recently sold a couple of sets on her behalf. Besides the fact that these sets have been well received, the store's four other suppliers have moved out of town and the store owner has offered your wife the job of supplying dinner sets for the next four weeks to meet the store's demand. With a new baby, it would be difficult for her to meet the demand on her own. As a result she has arranged to hire help over the four weeks. The hired help have different skills and hence different rates. Your wife, on looking over the required demand schedule, availability of firing time at the ceramics class and the cost of inventory storage at the class has realized that the problem is nontrivial. She decides to approach you to see whether your claimed expertise in operations research can help her. The demand, schedule, and costs are displayed in Table 12-2. You immediately realize that it can be set up as a linear program. However, on closer examination you notice that it can be formulated as a transportation problem that can be solved very efficiently.

- (a) Formulate this problem as a transportation problem. *Hint:* Let x_{ij} be the number of dinner sets produced in week i to satisfy demand in week j .
 (b) Solve it by hand.
 (c) Solve it by the **Transportation** software option to verify your solution.

12.6 Prove that the problem:

$$\begin{aligned} \text{Minimize} \quad & c^T x + \frac{2}{3} f_1 v_1 + \frac{1}{3} f_2 v_2 = z \\ \text{subject to} \quad & Ax = b \\ & -B_1 x + F_1 v_1 = d_1 \\ & -B_2 x + F_2 v_2 = d_3 \\ \text{with } & x \geq 0, v_1 \geq 0, v_2 \geq 0 \end{aligned}$$

is equivalent to

$$\begin{aligned} \text{Minimize} \quad & c^T x + \frac{1}{3} f_1 v'_1 + \frac{1}{3} f_1 v''_1 + \frac{1}{3} f_2 v_2 = z \\ \text{subject to} \quad & Ax = b \\ & -B_1 x + F_1 v'_1 = d_1 \\ & -B_1 x + F_1 v''_1 = d_1 \\ & -B_2 x + F_2 v_2 = d_3 \\ \text{with } & x \geq 0, v'_1 \geq 0, v''_1 \geq 0, v_2 \geq 0. \end{aligned}$$

12.7 Show how to convert the problem:

$$\begin{aligned} \text{Minimize} \quad & c^T x + \sum_{j=1}^J p_j f_j v_j = z \\ \text{subject to} \quad & Ax = b \\ & -B_j x + F_j v_j = d_j \quad j = 1, \dots, J \end{aligned}$$

$$\text{where } \sum_{j=1}^J p_j = 1, p_j \geq 0, j = 1, \dots, J \text{ and}$$

$$x \geq 0, v_j \geq 0 \text{ for } j = 1, \dots, J$$

where $p_j = N_j/N$, $\sum_{j=1}^J N_j = N$, and $N_j > 0$ are integers, to a problem of the form:

$$\begin{aligned} \text{Minimize} \quad & c^T x + \frac{1}{N} \sum_{k=1}^K f_k y_k = z \\ \text{subject to} \quad & Ax = b \\ & -B_k x + F_k y_k = d_k, \quad k = 1, \dots, K \\ \text{with } & x \geq 0, y_k \geq 0 \text{ for } k = 1, \dots, K. \end{aligned}$$

- 12.8 *Ph.D. Comprehensive Exam, June 15, 1967, at Stanford.* An individual is interested in choosing a portfolio of securities (stocks, bonds). Let X_i be the value of the i th security after one year per dollar invested today, $i = 1, \dots, n$. The variables X_1, X_2, \dots, X_n are assumed to be random variables with a known joint distribution. The individual has total current wealth A , to be distributed among the n securities. Let Y be the total value of his portfolio of securities after one year. Assume his aim is to maximize $E[U(Y)]$, the expected value of $U(Y)$, where $U(Y)$ is called the *von Neumann-Morgenstern utility indicator*. (It is the value he places on the outcome Y which measures his risk aversion; assume $U(Y)$ is a concave increasing function).

- (a) Derive in as simple a form as possible the equations and inequalities characterizing the optimal portfolio.
- (b) Suppose that an optimal allocation has been made, and that subsequently an additional security, $j = n + 1$, becomes available. Suppose further that the end-of-year value, X_{n+1} , is a random variable independent of the random variables X_1, X_2, \dots, X_n . State the necessary and sufficient condition that the optimal portfolio will be revised to include some portion of the new security.

PROBABILITY THEORY: OVERVIEW

In this appendix we introduce some basic concepts and notation of probability theory for use in solving stochastic linear programs.

A.1 BASIC CONCEPTS, EXPECTED VALUE, AND VARIANCE

We begin by paraphrasing some basic definitions and concepts found in W. Feller's book *Introduction to Probability Theory and Applications*.

Definition (Event, Sample Point, Sample Space): An *event* is defined to be the outcome of an experiment or an observation about the state of some system. A simple event, which is a single outcome of a single experiment or observation, is called a *sample point* and will be denoted by ω . The aggregate of all possible sample points is referred to as the *sample space* and will be denoted by the symbol Ω .

Associated with points ω in a sample space is a function $p(\omega) \geq 0$ that is referred to as *the probability of a simple event ω happening*. The sum of these $p(\omega)$ is unity. For example, if we have an urn containing balls labeled with either a 0 or a 1, we might assign the probability of "randomly" choosing a 0 ball as equal to the proportion of 0 balls to the total balls in the urn. In this case, the sample space may be thought of as consisting of two points lying on a line such that one point has coordinate 0 and the other point has coordinate 1 with probabilities $p(0) = \alpha$, $p(1) = \beta$, where $\beta = 1 - \alpha$.

Definition: A discrete sample space is one consisting of only a finite number of points, say n .

Another example of a discrete sample space is the toss of a coin, is where the probability of observing a head is $1/2$ and the probability of observing a tail is $1/2$. We ignore the very remote possibility of a coin landing on its edge.

The probability of an event being observed in a subset of the sample space is assumed to be simple additions of probabilities in the subset of the sample points associated with the event.

Definition (Random Variable): A random variable X is not a variable but a function defined over a sample space.

In particular, a discrete random variable X is a function that takes on a discrete set of values with values x_1, x_2, \dots, x_n , where x_i can be a sample point ω itself, an event, or, more generally, it can be a function of a set of sample points. The probability that a discrete random variable X can take on a value x_j is denoted by $P[X = x_j] = p_j$.

Definition (Probability Distribution): The set of these probabilities for all the outcomes of a discrete random variable X is called a *probability distribution* (or *density*) of the random variable. Clearly,

$$p_j \geq 0, \quad \sum_{j=1}^n p_j = 1. \quad (\text{A.1})$$

Definition (Expected Value or Mean): The *expected value* or *mean* of a discrete random variable X is

$$E[X] = \sum_{j=1}^n p_j x_j. \quad (\text{A.2})$$

Often the expected value is denoted by μ_x or simply μ when the association with X is clear. In general, the expected value of a function of X , say $h(X)$, is

$$E[h(X)] = \sum_{j=1}^n p_j h(x_j). \quad (\text{A.3})$$

Definition (Variance and Standard Deviation): The *variance* of a random variable X is denoted by σ_x^2 or simply σ^2 when the association with X is clear. It is defined as the expected squared deviations of an observation x_j from its expected values; i.e.,

$$\sigma_x^2 = \sum_{j=1}^n p_j (x_j - E[X])^2 = E[X^2] - (E[X])^2. \quad (\text{A.4})$$

Its positive square root, denoted by σ_x , is called the *standard deviation*.

Definition (Independent Random Variables): If $P[X = x_i] = p_i$ and $P[Y = y_j] = q_j$, then random variables X, Y are said to be *independent* if their joint probability distribution $P[(X, Y) = (x_i, y_j)] = p_i q_j$.

Definition (Correlation Coefficient): The *correlation coefficient* between the two variables X and Y is defined to be

$$\rho_{XY} = \frac{E[(X - E[X])(Y - E[Y])]}{\sigma_X \sigma_Y} = \frac{E[XY] - E[X]E[Y]}{\sigma_X \sigma_Y}. \quad (\text{A.5})$$

The correlation coefficient satisfies $-1 \leq \rho_{XY} \leq +1$. If the random variables X and Y are independent $\rho_{XY} = 0$. The *covariance* between random variables X and Y is defined to be

$$\sigma_{XY} = \sigma_X \sigma_Y \rho_{XY}. \quad (\text{A.6})$$

In a more general setting, a random variable V can be a vector function consisting of d components $V = (V_1, V_2, \dots, V_d)$ with outcomes $v^\omega = (v_1^\omega, v_2^\omega, \dots, v_d^\omega)$. The probabilities associated with v^ω will be denoted by $p(v^\omega)$ or simply by $p(v)$.

Definition (Independent Components): If the joint density probability distribution $p(v_1, v_2, \dots, v_n)$ satisfies

$$p(v_1, v_2, \dots, v_n) = p_1(v_1)p_2(v_2) \cdots p_n(v_n), \quad (\text{A.7})$$

the components of the random variable V are said to be *independent*.

If the components of the random variable V are *independent*, the sample space Ω is obtained by crossing the sets of outcomes for each component of the vector entry, i.e.,

$$\Omega = \Omega_1 \times \Omega_2 \times \cdots \times \Omega_d. \quad (\text{A.8})$$

Then the expectation of a function $h(V)$ is of the form:

$$E[h(V)] = \sum_{v_1} \sum_{v_2} \cdots \sum_{v_d} h(v) p_1(v_1) p_2(v_2) \cdots p_n(v_n). \quad (\text{A.9})$$

A random variable that is continuous is treated in much the same way except that we now use integrals.

Definition (Probability Density Function): If X is a continuous random variable, we associate with it a *probability density function* $f(x)$ with the property that

$$f(x) \geq 0, \quad \int_{\Omega} f(x) dx = 1. \quad (\text{A.10})$$

We shall often use the shorter form *density function* to mean a probability density function.

In the general case of a continuous random vector V , the expectation of a function $h(V)$ is given by

$$E[h(V)] = \int_{\Omega_1} \int_{\Omega_2} \cdots \int_{\Omega_l} h(v) f(v) dv_1 dv_2 \cdots dv_l. \quad (\text{A.11})$$

Sometimes an expectation is defined in terms of a Stieltjes integral; that is, the expected value of a function $g(x)$ of a random variable X , which is a combination of continuous and discrete elements, can be expressed as

$$E[X] = \int_{-\infty}^{\infty} g(x) dF(x) \quad (\text{A.12})$$

where $F(x) = P[X \leq x]$ is the cumulative density function.

Stochastic linear programs are hard to solve, in part because of the expense and difficulty involved in evaluating multiple integrals or multiple sums. The computation of the expected value typically involves a very large number of function evaluations, and each function evaluation may require solving a very large linear program.

A.2 NORMAL DISTRIBUTION AND THE CENTRAL LIMIT THEOREM

Many techniques in statistics are based on the normal probability distribution.

Definition (Normal Distribution): A random variable X is said to be distributed *normally* if its density function $f(x)$ is a *normal distribution*, i.e., it is of the form:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}, \quad (\text{A.13})$$

where $-\infty \leq \mu \leq \infty$ is the mean and $\sigma > 0$ is the standard deviation of the normal distribution. The normal distribution of (A.13) is often denoted by $\mathcal{N}(\mu, \sigma)$.

Definition (Standardized Normal Random Variable): A normally distributed random variable with mean $\mu = 0$ and standard deviation $\sigma = 1$, is called a *standardized* normal random variable. A standardized normal distribution is denoted by $\mathcal{N}(0, 1)$.

Normal distributions play an important role when sampling techniques are used because, according to the Central Limit Theorem, to be stated soon, the distribution of the mean of a sample of size n approaches a normal distribution as $n \rightarrow \infty$. For this purpose, we need to define what is meant by a random sample, sample mean, and sample variance.

Definition (Random Sample): The random variables X_1, X_2, \dots, X_n are defined to be a *random sample* from a population with probability density $f(x)$ if X_1, X_2, \dots, X_n are independent identically distributed random variables; i.e., if their joint probability density function satisfies

$$g(x_1, x_2, \dots, x_n) = f(x_1)f(x_2) \cdots f(x_n), \quad (\text{A.14})$$

where $f(x_i)$ is the density function of each of the random variables X_i .

Often the form of the density function of a random variable is known; for example, we may know that it is normally distributed. However, the density function may have parameters such as μ and σ , which are not known. The problem is then to estimate these unknown parameters through the use of a *statistic*.

Definition: A *statistic* is a function of a random sample of size n that is used to estimate an unknown parameter of a density function. For example, $\bar{X} = (1/n) \sum_{i=1}^n X_j$ is a statistic used to estimate the mean of a distribution.

LEMMA A.1 (Mean and Variance of a Random Sample) *Consider a population with density function $f(x)$, which has mean μ and variance σ^2 . Let X_1, X_2, \dots, X_n be a random sample from this population and let $\bar{X} = (1/n) \sum_{i=1}^n X_j$. Then*

$$E[\bar{X}] = \mu_{\bar{X}} = \mu \quad \text{and} \quad \text{Var}[\bar{X}] = \sigma_{\bar{X}}^2 = \frac{1}{n} \sigma^2. \quad (\text{A.15})$$

▷ **Exercise A.1** Prove Lemma A.1.

When sampling, we often can guess at the form of the probability density function but do not know its parameters, nor do we know its mean or variance. We can use one of several different ways to estimate the parameters; however, we typically would like the expected value of an estimator of a parameter to be equal to the true value of the parameter. Estimators that satisfy this property are called *unbiased estimators*.

Definition (Unbiased Estimator): An *unbiased estimator* γ of a function $g(\theta)$ of a parameter θ has the property $E[\gamma] = g(\theta)$. For example, the mean μ and variance σ^2 are parameters of a distribution. An unbiased estimator \bar{X} of μ is the mean of a sample of size n :

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i. \quad (\text{A.16})$$

Definition (Sample Variance): The *sample variance* of a random sample of size n is by definition:

$$\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2. \quad (\text{A.17})$$

A biased estimator of σ^2 is the sample variance because

$$E \left[\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2 \right] = \frac{n-1}{n} \sigma^2 \neq \sigma^2. \quad (\text{A.18})$$

Therefore an unbiased estimator of σ^2 is

$$\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2. \quad (\text{A.19})$$

Definition (Unbiased Sample Variance): We shall refer to (A.19) as the *unbiased sample variance* for estimating σ^2 .

▷ **Exercise A.2** Prove (A.18).

THEOREM A.2 (Chebyshev's Weak Law of Large Numbers) *Let \bar{X} be the sample mean of a random sample of size n drawn from a probability density $f(x)$ with mean μ and variance σ^2 . Then in order for*

$$P[-\epsilon < \bar{X} - \mu < \epsilon] \geq 1 - \delta \quad (\text{A.20})$$

for some specified $0 < \delta < 1$ and $\epsilon > 0$, choose sample size $n > \sigma^2/\epsilon^2\delta$.

The *weak law of large numbers* tells us how large we must take the sample size n in order to have a probability greater than $1 - \delta$ for $|\bar{X} - \mu| < \epsilon$, where \bar{X} is the sample mean.

Example A.1 (Sample Size Using Weak Law of Large Numbers) How large must the size n of the sample drawn from a population with known mean $\mu = 0$ and standard deviation $\sigma = 1000$ be in order that the probability is $\geq 1 - \delta = 0.95$ that the observed sample mean will be within $\epsilon = 10$ of the true mean 0? According to the weak law of large numbers, the sample size n should be chosen as $n = 100^2/10^2(0.5) = 200,000$.

THEOREM A.3 (Central Limit Theorem) *Let X_1, X_2, \dots, X_n be a random sample from a probability density function $f(x)$, which has mean μ and variance σ^2 . Define the random variable Z_n as a function of the sample mean \bar{X} as:*

$$Z_n = \frac{\bar{X} - E[\bar{X}]}{\sqrt{\text{Var}[\bar{X}]}} = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}}.$$

Then the distribution of Z_n approaches that of a standardized normal distribution as $n \rightarrow \infty$.

Example A.2 (Sample Size Using the Central Limit Theorem) If in Example A.1,

$$Z_n = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} = \frac{\bar{X}}{1000/\sqrt{n}},$$

then since

$$\int_{-1.96}^{+1.96} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt = 0.95,$$

we have

$$P[-10 \leq \bar{X} \leq 10] = P\left[\frac{-10}{1000/\sqrt{n}} \leq \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} \leq \frac{10}{1000/\sqrt{n}}\right] \geq 0.95.$$

Then $\frac{10}{1000/\sqrt{n}} = 1.96$ or $\sqrt{n} \geq 196$ or sample size should be chosen $n \geq 196^2 = 38,416$. The sample size n obtained using the Central Limit Theorem is much smaller than that obtained using Chebyshev's Weak Law of Large Numbers.

The Central Limit Theorem is one of the most important theorems in probability and statistics. It basically tells us that as the sample size grows the distribution of the sample mean \bar{X} approaches that of a normally distributed random variable.

A.3 CHI-SQUARE DISTRIBUTION, STUDENT'S *t*-DISTRIBUTION, AND CONFIDENCE INTERVALS

The “solution” of a stochastic linear program will be a decision x that is feasible. Its associated objective cost z is a random variable that depends on x and on the outcome of a random event ω . The problem is to choose $x = \hat{x}$ such that the *expected value* of $z_{\hat{x}}$ is minimum. If sample space Ω is too large, so that it is not possible to evaluate all $\omega \in \Omega$, then our goal will be given ϵ and δ to choose $x = \hat{x}$ and a sample size n such that $P[E[z_{\hat{x}}] < E[z_{\hat{x}}] + \epsilon] \geq 1 - \delta$. With this in mind we define the chi-square distribution, Student's *t*-distribution, and confidence intervals.

A.3.1 CHI-SQUARE DISTRIBUTION

A probability distribution, called the *chi-square distribution* plays an important role in determining the distribution of the unbiased sample variance.

Definition (Chi-Square Distribution): A random variable X is said to have a *chi-square distribution with k degrees of freedom* if its density function is of the form

$$f(x | k) = \frac{1}{\Gamma(k/2)} \left(\frac{1}{2}\right)^{k/2} x^{k/2-1} e^{-x/2}, \quad 0 < x < \infty. \tag{A.21}$$

where $\Gamma(u)$ is the gamma function:

$$\Gamma(u) = \int_0^{\infty} x^{u-1} e^{-x} dx \quad \text{for } u > 0 \quad (\text{A.22})$$

Integrating by parts results in the property

$$\Gamma(u + 1) = u\Gamma(u).$$

Hence if u is an integer, say $u = n$, then $\Gamma(n + 1) = n!$. Note that $\Gamma(u)$ is defined by (A.22) for any u , integer or noninteger, for example,

$$\Gamma(n + 1/2) = \frac{1 \cdot 3 \cdot 5 \cdots (2n - 1)}{2^n} \sqrt{\pi}$$

and $\Gamma(1/2) = \sqrt{\pi}$, $\Gamma(3/2) = (1/2)\sqrt{\pi}$. A random variable with a chi-square distribution with k degrees of freedom has $E[X] = k$ and $\text{Var}[X] = 2k$.

▷ **Exercise A.3** Show that $E[X] = k$ and $\text{Var}[X] = 2k$ for a random variable X that has a chi-square distribution with k degrees of freedom.

Next we discuss some properties of random samples.

LEMMA A.4 (Sample Mean from a Normal Distribution) Let \bar{X} be the sample mean of a random sample X_1, X_2, \dots, X_n drawn from a normal distribution $\mathcal{N}(\mu, \sigma^2)$. Then \bar{X} is itself normal with mean μ and variance σ^2/n .

LEMMA A.5 (Chi-Square Distribution) Let U be a random variable defined by:

$$U = \sum_{i=1}^k \left(\frac{X_i - \mu_i}{\sigma_i} \right)^2$$

where the X_i are normally and independently distributed with means μ_i and variance σ_i^2 . Then U has a chi-square distribution with k degrees of freedom.

COROLLARY A.6 Let $S_{\bar{X}}$ be the unbiased sample variance (A.19) of a random sample X_1, X_2, \dots, X_n from a normal distribution with mean μ and variance σ^2 . Then

$$U = \frac{(n-1)S_{\bar{X}}^2}{\sigma^2}$$

has a chi-square distribution with $(n-1)$ degrees of freedom.

A.3.2 STUDENT'S *t*-DISTRIBUTION

The Student's *t*-distribution also plays a very important role in estimation.

Definition (Student's t-Distribution): A random variable X is said to have a *Student's t-distribution* with k degrees of freedom if its probability density function is of the form

$$stu(x | k) = \frac{\Gamma([k + 1]/2)}{\Gamma(k/2)} \frac{1}{\sqrt{k\pi}} \frac{1}{(1 + x^2/k)^{(k+1)/2}}, \tag{A.23}$$

where Γ is the gamma function defined by (A.22).

LEMMA A.7 (Student's *t*-Distribution) *Let Z be a random variable with a standardized normal distribution and let U be an independent random variable with a chi-square distribution with k degrees of freedom. Then the random variable*

$$X = \frac{Z}{\sqrt{U/k}} \tag{A.24}$$

has a Student's t-distribution with k degrees of freedom, $stu(x | k)$.

Let X_1, X_2, \dots, X_n be a random sample from a normal probability density function with mean μ and variance σ^2 . Define

$$Z = \frac{(\bar{X} - \mu)}{\sigma/\sqrt{n}} \tag{A.25}$$

$$U = \left(\frac{1}{\sigma^2}\right) \sum_{i=1}^n (X_i - \bar{X})^2. \tag{A.26}$$

Then, from Lemma A.4, it is easy to see that Z has a standardized normal distribution and, from Corollary A.6, U has a chi-square distribution with $n - 1$ degrees of freedom.

LEMMA A.8 (Independence of Z and U) *The random variables Z defined by (A.25) and U defined by (A.26) are independent.*

Because Z and U are independent by Lemma A.8, from Lemma A.7 it follows that

$$\frac{(\bar{X} - \mu)/(\sigma/\sqrt{n})}{\sqrt{(1/\sigma^2) \sum_{i=1}^n (X_i - \bar{X})^2 / (n - 1)}} = \frac{\sqrt{n(n - 1)}(\bar{X} - \mu)}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2}} \tag{A.27}$$

has a Student's *t*-distribution with $n - 1$ degrees of freedom, $stu(x | n - 1)$.

A.3.3 CONFIDENCE INTERVALS

Before formally defining a confidence interval, we illustrate the subject of confidence intervals through an example.

Example A.3 (Illustration of Confidence Intervals) Suppose that we would like to estimate the mean μ of a normal distribution with known standard deviation $\sigma = 1$. We can sample from the distribution and construct the sample mean \bar{X} as an estimate of the unknown mean μ . However, often we are more interested in constructing an interval that is guaranteed to cover the unknown mean μ with specified probability even though we do not know what the fixed value of μ is. For example, we may be interested in the interval that gives a probability of 0.9554 of covering the unknown mean μ . In order to do this, observe that, for a sample of size n , the random variable

$$Z = \frac{\bar{X} - \mu}{\sigma/\sqrt{n}} = \frac{\bar{X} - \mu}{1/\sqrt{n}} \quad (\text{A.28})$$

has a standardized normal distribution. Now from the probability tables we know that

$$P[-2 < Z < 2] = \int_{-2}^2 \frac{1}{\sqrt{2\pi}} e^{-x^2/2} = 0.9554 \quad (\text{A.29})$$

to four decimal places. Substituting (A.28) into (A.29) we obtain

$$\begin{aligned} 0.9554 &= P[-2 < Z < 2] = P\left[-2 < \frac{\bar{X} - \mu}{1/\sqrt{n}} < 2\right] \\ &= P[\bar{X} - 2/\sqrt{n} < \mu < \bar{X} + 2/\sqrt{n}]. \end{aligned} \quad (\text{A.30})$$

In particular, for $n = 100$,

$$P[\bar{X} - 0.2 < \mu < \bar{X} + 0.2] = 0.9554 \quad (\text{A.31})$$

Then $P[\bar{X} - 0.2 < \mu < \bar{X} + 0.2] = 0.9554$ measures our “confidence” that the unknown true mean lies in the *open interval* $(\bar{X} - 2/\sqrt{n}, \bar{X} + 2/\sqrt{n})$, which is called the 95.54% confidence interval. Notice that the interval $(\bar{X} - 0.2, \bar{X} + 0.2)$ is a random interval that covers the unknown true mean μ with probability 0.9554. That is, if random samples of size 100 were repeatedly drawn from the population and the random intervals $(\bar{X} - 0.2, \bar{X} + 0.2)$ repeatedly computed, then the fraction of times that the interval actually covered the mean would approach 0.9554 as the number of repetitions tends toward infinity. At this point we would also like to point out that often a particular realization of an interval (i.e., for one sample size of 100) is also called a *confidence interval for estimating μ* .

We now formalize the definition of a confidence interval.

Definition (Confidence Interval, Confidence Limit, Confidence Level): Let X_1, X_2, \dots, X_n be a random sample from a probability density function $f(x; \theta)$ parameterized by a constant θ whose unknown value we wish to estimate. Let $L \leq H$, where $L = L(X_1, X_2, \dots, X_n)$, $H = H(X_1, X_2, \dots, X_n)$, be statistics for θ such that $P[L < \theta < H] = \gamma$ where γ does not depend on θ . Then (L, H) is defined to be a 100γ percent confidence interval for θ and γ is called

the *confidence level*. Furthermore, the random variable L is called the *lower confidence limit* and the random variable H is called the *upper confidence limit* for θ . A particular realization (l, h) of (L, H) is also called a 100γ percent *confidence interval* for estimating θ . In a similar manner one can define upper and lower *one-sided confidence intervals*.

In order to derive a confidence interval for estimating the unknown mean μ of a distribution whose variance is also unknown, we note that

$$\frac{(\bar{X} - \mu)/(\sigma/\sqrt{n})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2/(n-1)\sigma^2}} = \frac{\bar{X} - \mu}{S_{\bar{X}}^2/\sqrt{n}}$$

has a Student's t -distribution with $(n-1)$ degrees of freedom, $stu(x | n-1)$. For some choice of q_1 and q_2 , typically $q_1 = q_2$, let

$$P\left[-q_1 < \frac{\bar{X} - \mu}{S_{\bar{X}}^2/\sqrt{n}} < q_2\right] = \gamma.$$

Then $(\bar{X} - q_2(S_{\bar{X}}^2/\sqrt{n}), \bar{X} + q_1(S_{\bar{X}}^2/\sqrt{n}))$ is a 100γ percent confidence interval for μ .

In a similar manner we can derive a confidence interval for estimating the unknown variance σ^2 of a distribution whose mean is also unknown. In order to do this we note that

$$\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{\sigma^2} = \frac{(n-1)S_{\bar{X}}^2}{\sigma^2}$$

has a chi-square distribution with $n-1$ degrees of freedom, $f(x | n-1)$. For some choice of q_1 and q_2 , let

$$P\left[q_1 < \frac{(n-1)S_{\bar{X}}^2}{\sigma^2} < q_2\right] = \gamma.$$

Then $((n-1)S_{\bar{X}}^2/q_2, (n-1)S_{\bar{X}}^2/q_1)$ is a 100γ percent confidence interval for the variance σ^2 .

A.4 NOTES & SELECTED BIBLIOGRAPHY

For an introduction to probability theory, see Feller [1957, 1969].

This page intentionally left blank

REFERENCES

A

- Abadie, J. (1965). Verbal Communication to Dantzig.
- Abrahamson, P.G. (1983). "A Nested Decomposition Approach for Solving Staircase Linear Programs," Technical Report SOL 83-4, Department of Operations Research, Stanford University, Stanford, CA.
- Ackoff, R.L. and Rivett, P. (1963). *A Manager's Guide to Operations Research*, Wiley, New York.
- Adler, I. (1976). "Equivalent Linear Programs," Department of IE & OR, University of California, Berkeley, California.
- Adler, I., Karmarkar, N., Resende, M.G.C. and Veiga, G. (1990). "Data Structures and Programming Techniques for the Implementation of Karmarkar's Algorithm," *ORSA Journal on Computing* **1**, 84–106.
- Adler, I., Resende, M.G.C., Veiga, G., and Karmarkar, N. (1989). "An Implementation of Karmarkar's Algorithm for Linear Programming," *Mathematical Programming* **44**, 297–336.
- Adler, I. and Ülkücü, A. (1973). "On the Number of Iterations in the Dantzig-Wolfe Decomposition," in D. M. Himmelblau (ed.), *Decomposition of Large Scale Problems*, North-Holland, Amsterdam, the Netherlands, 181–187.
- Ahuja, R.K., Magnanti, T.L., and Orlin, J.B. (1991). "Network Flows," in G.L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd (eds.), *Handbooks in Operations Research and Management Science, Vol 1: Optimization*, North-Holland, Amsterdam, the Netherlands, 211–369.
- Ahuja, R.K., Magnanti, T.L., and Orlin, J.B. (1993). *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Ahuja, R.K., Mehlhorn, K., Orlin, J.B., and Tarjan, R.E. (1990). "Faster Algorithms for the Shortest Path Problem," *Journal of the Association for Computing Machinery* **37**, 211–369.
- Ali, A.I., Helgason, R.V., Kennington, J.L., and Lall, H.S. (1978). "Primal Simplex Network Codes: State of the Art Implementation Technology," *Networks*, **8**, 315–339.

- Andersen, E.D. and Ye, Y. (1995). "Combining Interior-Point and Pivoting Algorithms for Linear Programming," working paper, Department Sciences, The University of Iowa, Iowa City, Iowa; to appear in *Management Science*.
- Anstreicher, K. M. (1986). "A Monotonic Projective Algorithm for Fractional Linear Programming," *Algorithmica* **1**, 483–498.
- Anstreicher, K.M. (1989). "A Worst-Case Step in Karmarkar's Algorithm," *Mathematics of Operations Research* **14**, 294–302.
- Anstreicher, K.M. (1990). "On Long Step Path Following and SUMT for Linear and Quadratic Programming," Technical Report, Yale School of Management, Yale University, New Haven, CT.
- Anstreicher, K.M. and Watteyne, P. (1993). "A Family of Search Directions for Karmarkar's Algorithm," *Operations Research* **41**, 759–767.
- Arid, T.J., and Lynch, J.E. (1975). "Computable Accurate Upper and Lower Error Bounds for Approximate Solutions of Linear Algebra Systems," *ACM Transactions on Mathematical Software* **1**, 217–231.
- Arntzen, B.C., Brown, G.G., Harrison, T.P., and Trafton, L.L. (1995). *Interfaces* **25:1**, 69–93.
- Avi-Itzhak, H. (1994). "High Accuracy Correlation Based Pattern Recognition," Ph.D. thesis, Electrical Engineering, Stanford University, Stanford, CA.
- Avi-Itzhak, H.I., Van Mieghen, J.A., and Rub, L. (1995). "Multiple Subclass Pattern Recognition—A Max-Min Approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **17**, 418–431.
- Avriel, M. (1976). *Nonlinear Programming: Analysis and Methods*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Axelsson, O. and Munksgaard, N. (1983). "Analysis of Incomplete Factorizations with Fixed Storage Allocation," in D. Evans (ed.), *Preconditioning Methods Theory and Applications*, Gordon and Breach, New York, 219–241.

B

- Bahn, O., Goffin, J.L., Vial, J. P., and Merle, O.D. (1994). "Implementation and Behavior of an Interior Point Cutting Plane Algorithm for Convex Programming: An Application to Geometric Programming," *Discrete Applied Mathematics* **49**, 3–23; previously published in 1991 as a working paper, University of Geneva, Geneva, Switzerland.
- Bailey, D. (1988). "Extra High Speed Matrix Multiplication on the Cray-2," *SIAM Journal on Scientific and Statistical Computing* **9**, 603–607.
- El-Bakry, A., Tapia, R., and Zhang, Y. (1991). "Numerical Comparisons of Local Convergence Strategies for Interior-Point Methods in Linear Programming," Technical Report TR 91-18, Department of Computational and Applied Mathematics, Rice University, Houston, TX.
- Balanski, M.L. and Gomory, R.E. (1963). "A Mutual Primal-Dual Simplex Method," in R.L. Graves and P. Wolfe (eds.), *Recent Advances in Mathematical Programming*, McGraw-Hill, New York, 17 - 26.

- Barnes, E.R. (1986). "A Variation on Karmarkar's Algorithm for Linear Programming," *Mathematical Programming* **36**, 174–182.
- Bartels, R.H. (1971). "A Stabilization of the Simplex Method," *Numerische Mathematik* **16**, 414–434.
- Bartels, R.H. and Golub, G.H. (1969). "The Simplex Method of Linear Programming Using the LU Decomposition," *Communications of the Association for Computing Machinery* **12**, 266–268.
- Bartels, R.H., Stoer, J. and Zenger, C. (1971). "A Realization of the Simplex Method Based on Triangular Decompositions," in J.H. Wilkinson and C. Reinsch (eds.), *Contributions I/II in Handbook for Automatic Computation, Volume II: Linear Algebra*, Springer-Verlag, Berlin and New York, 152–190.
- Bastian, M. (1984). "Implicit Representation of Generalized Variable Upper Bounds Using the Elimination Form of the Inverse on Secondary Storage," *Mathematical Programming* **30**, 357–361.
- Bazaraa, M.S., Jarvis, J.J., and Sherali, H.D. (1990). *Linear Programming and Network Flows*, John Wiley and Sons, New York.
- Beale, E.M.L. (1954a). "Linear Programming by the Method of Leading Variables," *Report of the Conference on Linear Programming*, (May), arranged by Ferranti Ltd., London.
- Beale, E.M.L. (1954b). "An Alternative Method for Linear Programming," *Proceedings of the Cambridge Philosophical Society* **50**, 513–523.
- Beale, E.M.L. (1955a). "On Minimizing a Convex Function Subject to Linear Inequalities," *Journal of the Royal Statistical Society, Series B*, **17**, 173–184.
- Beale, E.M.L. (1955b). "Cycling in the dual simplex algorithm," *Naval Research Logistics Quarterly*, **2**, 269–275.
- Beale, E.M.L. (1970). "Matrix Generators and Output Analyzers," in H.W. Kuhn (ed.), *Proceedings of the Princeton Symposium on Mathematical Programming*, Princeton University Press, Princeton, New Jersey, 25–36.
- Beale, E.M.L., Hughes, P.A.B., and Small, R. (1965). "Experiences in Using a Decomposition Program," *Computer Journal* **8**, 13–18.
- Bell, C.G. and Newell, A. (1971). *Computer Structures: Readings and Examples*, McGraw-Hill, New York.
- Bellman, R. E. (1957). *Dynamic Programming*, Princeton University Press, Princeton, New Jersey.
- Bellman, R.E. (1958). "On a Routing Problem," *Quarterly of Applied Mathematics* **1958**, 87–90.
- Bellman, R.E. and Dreyfus, S. (1962). *Applied Dynamic Programming*, Princeton University Press, Princeton, New Jersey.
- Ben-Dov, Y., Hayre, L., and Pica, V. (1992). "Mortgage Valuation Models at Prudential Securities," *Interfaces* **22:1**, 55–71.
- Benders, J.F. (1962). "Partitioning Procedures for Solving Mixed-Variables Programming Problems," *Numerische Mathematik* **4**, 238–252.

- Benichou, M., Gauthier, J.M., Hentges, G., and Ribi re, G. (1977). "The Efficient Solution of Large-Scale Linear Programming Problems—Some Algorithmic Techniques and Computational Results," *Mathematical Programming* **13**, 280–322.
- Bennett (1963). "An Approach to Some Structured Linear Programming Problems," Bas-sar Computing Department, School of Physics, University of Sydney, Australia.
- Bertsekas, D.P. (1988). "The Auction Algorithm: A Distributed Relaxation Method for the Assignment Problem," *Annals of Operations Research* **14**, 105–123.
- Berge, C. (1963). *Topological Spaces*, Oliver & Boyd Ltd., Edinburgh.
- Bertsekas, D.P. (1991). *Linear Network Optimization: Algorithms and Codes*, The MIT Press, Cambridge, Massachusetts.
- Birge, J.R. (1980). "Solution Methods for Stochastic Dynamic Linear Programs," Ph.D. thesis. Technical Report SOL 80-29, Department of Operations Research, Stanford University, Stanford, CA.
- Birge, J.R. (1985a). "Aggregation in Stochastic Linear Programming," *Mathematical Programming* **31**, 25–41.
- Birge, J.R. (1985b). "Decomposition and Partitioning Methods for Multi-Stage Stochastic Linear Programming," *Operations Research* **33**, 989–1007.
- Birge, J.R., Dempster, M.A., Gassman, H.I., Gunn, E.A., King, A.J., and Wallace, S.W. (1987). "A Standard Input Format for Multi-Period Stochastic Linear Programs," *Mathematical Programming Society COAL Newsletter* **17**, 1–20.
- Birge, J.R. and Holmes, D.F. (1992). "Efficient Solution of Two-Stage Stochastic Linear Programs Using Interior Point Methods," *Operations Research Letters* **11**, 245–276.
- Birge, J.R. and Louveaux, F. (1997). *Introduction to Stochastic Programming*, Springer-Verlag, Berlin and New York.
- Birge, J.R. and Wallace, S.W. (1988). "A Separable Piecewise Linear Upper Bound for Stochastic Linear Programs," *SIAM Journal on Control and Optimization* **26**, 725–739.
- Birge, J.R. and Wets, R.J. (1986). "Designing Approximate Schemes for Stochastic Optimization Problems, in Particular for Stochastic Programs with Recourse," *Mathematical Programming Study* **27**, 54–102.
- Birge, J.R. and Wets, R.J. (1987). "Computing Bounds for Stochastic Programming Problems by Means of a Generalized Moment Problem," *Mathematics of Operations Research* **12**, 49–162.
- Birge, J.R. and Wets, R.J. (1989). "Sublinear Upper Bounds for Stochastic Programming Problems with Recourse," *Mathematical Programming* **43**, 131–149.
- Birkhoff, G. (1946). "Three Observations on Linear Algebra," *Rev. Univ. Nac. Tucum n, Ser. A.*, Vol. 5, 147–151.
- Bisschop, J. and Meeraus, A. (1981). "Towards Successful Modeling Applications in a Strategic Planning Environment," in G.B. Dantzig, M.A.H. Dempster, and M.J. Kallio. (eds.), *Large-Scale Linear Programming*, Vol. 2, CP-81-51, IIASA Collaborative Proceedings Series, Laxenberg, Austria, 712–745.
- Bisschop, J. and Meeraus, A. (1982). "On the Development of a General Algebraic Modeling System," *Mathematical Programming Study* **20**, 1–29.

- Bixby, R.E. (1981). "Hidden Structure in Linear Programs," in H. Greenberg and J. Maybee (eds.), *Computer Assisted Analysis and Model Simplification*, Academic Press, London and New York, 327–360.
- Bixby, R.E. (1984). "Recent Algorithms for Two Versions of Graph Realization and Remarks on Applications to Linear Programming," in W.R. Pulleybank (ed.), *Progress in Combinatorial Optimization*, Academic Press, Canada, 39–67.
- Bixby, R.E. (2002). "Solving Real-World Linear Programs: A Decade and More of Progress," *Operations Research Letters* **50**, 3–15.
- Bixby, R.E. and Cunningham, W.H. (1980). "Converting Linear Programs to Network Problems," *Mathematics of Operations Research* **5**, 321–357.
- Bixby, R.E. and Fourer, R. (1988). "Finding Embedded Network Rows in Linear Programs I. Extraction Heuristics," *Management Science* **34**, 342–376.
- Bixby, R.E., Gregory, J.W., Lustig, I.J., Marsten, R.E., and Shanno, D.F. (1992). "Very Large Scale Linear Programming: A Case Study in Combining Interior Point and Simplex Methods," *Operations Research* **40**, 885–897.
- Bixby, R.E. and Saltzman, M.J. (1994). "Recovering an Optimal LP Basis from the Interior Point Solution," *Operations Research Letters* **15**, 169–178; previously published in 1992 in Technical Report 607, Department of Mathematical Sciences, Clemson University, Clemson, SC.
- Bixby, R.E. and Wagner, D.K. (1987). "A Note on Detecting Simple Redundancies in Linear Systems," *Operations Research Letters* **6**, 15–17.
- Bixby, R.E. and Wagner, D.K. (1988). "An Almost Linear-Time Algorithm for Graph Realization," *Mathematics of Operations Research* **13**, 99–123.
- Björck, Å. (1967). "Solving Linear Least Squares Problems by Gram-Schmidt Orthogonalization," *BIT* **7**, 1–21.
- Björck, Å. (1987). "Least Square Methods," working paper, Department of Mathematics, Linköping University, S-581 83 Linköping, Sweden.
- Björck, Å., Plemmons, R.J., and Schneider, H. (1981). *Large-Scale Matrix Problems*, North-Holland, Amsterdam, the Netherlands.
- Bland, R.G. (1977). "New Finite Pivoting Methods for the Simplex Method," *Mathematics of Operations Research* **2**, 103–107.
- Blue, J.L. (1975). "Automatic Numerical Quadrature—DQUAD," *Bell Laboratories Report*, Murray Hill, NJ
- Borel, E. (1921). "La Théorie du Jeu et les Équations Intégrales à Noyau Symétrique," *Comptes Rendus de l'Académie des Sciences U.S.S.R.* **173**, 1304–1307; translated by Leonard J. Savage in *Econometrica*, **21**, 97–100, January 1953.
- Borel, E. (1924). "Sur les Jeux où Interviennent le Hasard et l'habileté des Joueurs," *Théorie des Probabilités*, Librairie scientifique, Hermann, Paris, 204–224; translated by Leonard J. Savage in *Econometrica*, **21**, 101–115, January 1953.
- Borel, E. (1927). "Sur les Systèmes de Formes Linéaires à Déterminant Symétrique Gauches et la Théorie Générale du Jeu," from "Algèbre et Calcul des Probabilités," *Comptes Rendus de l'Académie des Sciences U.S.S.R.* **184**, 52–53; translated by Leonard J. Savage in *Econometrica*, **21**, 101–115, January 1953.

- Borgwardt, K.H. (1982a). "The Average Number of Pivot Rules Required by the Simplex Method is Polynomial," *Zeitschrift für Operations Research* **26**, 157–177.
- Borgwardt, K.H. (1982b). "Some Distribution Independent Results About the Asymptotic Order of the Average Number of Pivot Steps in the Simplex Method," *Mathematics of Operations Research* **7**, 441–462.
- Borgwardt, K.H. (1987a). *The Simplex Method: A Probabilistic Analysis*, Springer-Verlag, Berlin and New York.
- Borgwardt, K.H. (1987b). "Probabilistic Analysis of the Simplex Method," in *Operations Research Proceedings 16th Dgor Meeting*, 564–576.
- Bradley, G.H., Brown, G.G., and Graves, G.W. (1977). "Design and Implementation of Large Scale Primal Transshipment Algorithms," *Management Science* **24**, 1–34.
- Bradley, S.P., Hax, A.C., and Magnanti, T.L. (1977). *Applied Mathematical Programming*, Addison-Wesley Publishing Company, Reading, Massachusetts.
- Brearley, A.L., Mitra, G., and Williams, H.P. (1975). "Analysis of Mathematical Programs Prior to Applying the Simplex Algorithm," *Mathematical Programming* **8**, 54–83.
- Brent, R. P. (1973a). "Reducing the Retrieval Time of Scatter Storage Techniques," *Communications of the Association for Computing Machinery* **16:2**, 105–109.
- Brent, R.P. (1973b). *Algorithms for Finding Zeros and Extrema of Functions Without Calculating Derivatives*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Brickman, L. (1989). *Mathematical Introduction to Linear Programming and Game Theory*, Springer-Verlag, Berlin and New York.
- Brooke, A., Kendrick, D., and Meeraus, A. (1988). *GAMS: A User's Guide*, Scientific Press, South San Francisco, California.
- Brown, G.G. and McBride, R.D. (1984). "Solving Generalized Networks," *Management Science* **30**, 1497–1523.
- Brown, G.G. and Thomen, D.S. (1980). "Automatic Identification of Generalized Upper Bounds in Large-Scale Optimization Models," *Management Science* **26**, 1166–1184.
- Brown, G.G. and Wright, W. (1981). "Automatic Identification of Embedded Structure in Large-Scale Optimization Models," in H. Greenberg and J. Maybee (eds.), *Computer Assisted Analysis and Model Simplification*, Academic Press, London and New York, 369–388.
- Brown, G.G. and Wright, W. (1984). "Automatic Identification of Embedded Network Rows in Large-Scale Optimization Models," *Mathematical Programming* **29**, 41–56.
- Brumelle, S.L. and McGill, J.I. (1993). "Airline Seat Allocation with Multiple Nested Fare Classes," *Operations Research* **41**, 127–137.
- Broyden, C.G. (1973). "Some Condition Number Bounds for the Gaussian Elimination Process," *Journal of the Institute of Mathematics and Its Applications* **12**, 273–286.
- Bunch, J.R. and Parlett, B.N. (1971). "Direct Methods for Solving Symmetric Indefinite Systems of Linear Equations," *SIAM Journal on Numerical Analysis* **8**, 639–655.
- Bunch, J.R. and Kaufman, L.C. (1977). "Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Equations," *Linear Algebra and Its Applications* **34**, 341–370.

- Burks, A.W., Goldstine, H.H., and von Neumann, J. (1946). "Planning and Coding of Problems for an Electronic Computing Instrument," Parts 1 and 2, Institute for Advanced Study, Princeton, New Jersey. Reprinted in *Datamation* **8**:9, 24–31, September, 1962 and *Datamation* **8**:10, 36–41, October, 1962.
- Businger, P.A. (1968). "Matrices Which can be Optimally Scaled," *Numerische Mathematik* **12**, 346–348.
- Buzbee, B.L. (1986). "A Strategy for Vectorization," *Parallel Computing* **3**, 187–192.

C

- Cariño, D.R., Kent, T., Myers, D.H., Stacy, C., Sylvanus, M., Turner, A.L., Watanabe, K., and Ziemba, W.T. (1994). "The Russel-Yasuda Kasai Model: An Asset/Liability Model for a Japanese Insurance Company Using Multistage Stochastic Programming," *Interfaces* **14**:1, 29–49.
- Carpenter, T.J., Lustig, I.J., Mulvey, J.M., and Shanno, D.F. (1993). "Higher Order Predictor-Corrector Interior Point Methods with Applications to Quadratic Objectives," *SIAM Journal on Optimization* **3**, 696–725.
- Carpento, G., Martello, S., and Toth, P. (1988). "Algorithms and Codes for the Assignment Problem," in B. Simeone, P. Toth, G. Gallo, F. Maffioli, and S. Pallotino (eds.), *FORTRAN Codes for Network Optimization, Annals of Operations Research* **13**, 193–224.
- Chambers, J.M. (1977). *Computational Methods for Data Analysis*, John Wiley and Sons, New York.
- Chan, T.F. (1985). "On the Existence and Computation of LU Factorizations with Small Pivots," *Mathematics of Computation* **42**, 535–548.
- Charnes, A. (1952). "Optimality and Degeneracy in Linear Programming," *Econometrica*, **20**, 160–170.
- Charnes, A., Cooper, W.W., and Mellon, B. (1952). "Blending Aviation Gasoline—A Study of Programming Interdependent Activities in an Integrated Oil Company," *Econometrica*, **20**, 135–139.
- Charnes, A. and Kortanek, K.O. (1963). "An Opposite Sign Algorithm for Purification to an Extreme Point Solution," Memorandum No. 89, Office of Naval Research.
- Charnes, A. and Lemke, C. (1960). "Multi-Copy Generalized Networks and Multi-Page Programs," R.P.I. Math Report No. 41, Rensselaer Polytechnic Institute; Troy, New York.
- Cheng, M.C. (1985). "Generalized Theorems for Permanent Basic and Nonbasic Variables," *Mathematical Programming* **31**, 229–234.
- Cheng, M.C. (1987). "General Criteria for Redundant and Nonredundant Linear Inequalities," *Journal of Optimization Theory and Applications* **53**, 37–42.
- Cherkassky, B.V., Goldberg, A.V., and Radzik, T. (1996). "Shortest Path Algorithms: Theory and Experimental Evaluation," *Mathematical Programming* **73**, 129–174.
- Choi, I.C., Monma, C.L., and Shanno, D.F. (1990). "Further Developments of a Primal-Dual Interior Point Method," *Operations Research* **40**, 885–897.

- Chvátal, V. (1983). *Linear Programming*, W. H. Freeman and Company, New York.
- Clements, D.W. and Reid, R.A. (1994). "Analytical MS/OR Tools Applied to a Plant Closure," *Interfaces* **24**:2, 1–12.
- Cline, A.K., Conn, A.R., and van Loan, C. (1982). "Generalizing the LINPACK Condition Estimator," in J.P. Hennart (ed.), *Numerical Analysis*, Lecture Notes in Mathematics No. 909, Springer-Verlag, Berlin and New York, 73–83.
- Cline, A.K., Moler, C.B., Stewart, G.W., and Wilkinson, J.H., (1979). "An Estimate for the Condition Number of a Matrix," *SIAM Journal on Numerical Analysis* **16**, 368–375.
- Cline, A.K. and Rew, R.K. (1983). "A Set of Counter Examples to Three Condition Number Estimators," *SIAM Journal on Scientific and Statistical Computing* **4**, 602–611.
- Cody, W.J. (1974). "The Construction of Numerical Subroutine Libraries," *SIAM Review* **16**, 36–46.
- Cohen, A.M. (1974). "A Note on Pivot Size in Gaussian Elimination," *Linear Algebra and Its Applications* **8**, 361–368.
- Coleman, T.F. and More, J.J. (1983). "Estimation of Sparse Jacobian Matrices and Graph Coloring Problems," *SIAM Journal on Numerical Analysis* **20**, 187–209.
- Cope, J.E. and Rust, B.W. (1979). "Bounds on Solutions of Systems with Accurate Data," *SIAM Journal on Numerical Analysis* **16**, 950–963.
- Cornfield, J. (1940). Verbal Communication to G.B. Dantzig.
- Cottle, R.W. (1974). "Manifestations of the Schur Complement," *Linear Algebra and Its Applications* **8**, 189–211.
- Courant, R. (1943). "Variational Methods for the Solution of Problems of Equilibrium and Vibrations," *Bulletin of the American Mathematical Society* **49**, 1–23.
- Cowell, W. R. (ed.), (1983). *Sources and Development of Mathematical Software*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Cox, L.A., Kuehner W., Parrish, S.H., and Qiu, Y. (1993). "Optimal Expansion of Fiber-Optic Telecommunications Networks in Metropolitan Areas," *Interfaces* **23**:2, 35–48.
- Cryer, C.W. (1968). "Pivot Size in Gaussian Elimination," *Numerische Mathematik* **12**, 335–345.
- Cunningham, W.H. (1979). "Theoretical Properties of the Network Simplex Method," *Mathematics of Operations Research* **4**, 196–208.
- Cunningham, W.H. and Klinecicz, J.G. (1983). "On Cycling in the Network Simplex Method," *Mathematical Programming* **26**, 182–189.
- Curtis, A.R., Powell, M.J.D. and Reid, J.K. (1974). "On the Estimation of Sparse Jacobian Matrices," *Journal of the Institute of Mathematics and Its Applications* **13**, 117–119.
- Curtis, A.R. and Reid, J.K. (1971). "The Solution of Large Sparse Unsymmetric Systems of Linear Equations," *Journal of the Institute of Mathematics and Its Applications* **8**, 344–353.

D

- Dahl, O., Dijkstra, E., and Hoare, C.A.R. (1972). *Structured Programming*, Academic Press, London and New York
- Dahlquist, G. and Björk, Å. (1974). *Numerical Methods*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Dantzig, G.B. (1939). "On a Class of Distributions that Approach the Normal Distribution Function," *Annals of Mathematical Statistics* **10**, September, 247–253.
- Dantzig, G.B. (1940). "On the Non-Existence of Tests of Students' Hypothesis Involving Power Functions Independent of Sigma," *Annals of Mathematical Statistics* **11**, June, 186–192.
- Dantzig, G.B. (1948, January 5). "A Theorem on Linear Inequalities," unpublished report.
- Dantzig, G.B. (1949). "Programming in a Linear Structure," Report of the September 9, 1948 meeting in Madison, *Econometrica*, **17**, 73–74.
- Dantzig, G.B. (1951a). "Maximization of a Linear Function of Variables Subject to Linear Inequalities," in T.C. Koopmans (ed.), *Activity Analysis of Production and Allocation*, July–October 1949, Cowles Commission Monograph 13, Proceedings of Linear Programming Conference, June 20–24, 1949, John Wiley and Sons, New York, 339–347.
- Dantzig, G.B. (1951b). "Application of the Simplex Method to the Transportation Problem," in T.C. Koopmans (ed.), *Activity Analysis of Production and Allocation*, July–October 1949, Cowles Commission Monograph 13, Proceedings of Linear Programming Conference, June 20–24, 1949, John Wiley and Sons, New York, 359–373.
- Dantzig, G.B. (1951c). "Programming of Interdependent Activities, II: Mathematical Model," *Econometrica*, **17**, 200–211; also in T.C. Koopmans (ed.), *Activity Analysis of Production and Allocation*, July–October 1949, Cowles Commission Monograph 13, Proceedings of Linear Programming Conference, June 20–24, 1949, John Wiley and Sons, New York, 19–32.
- Dantzig, G.B. (1951d). "A Proof of the Equivalence of the Programming Problem and the Game Problem, in T.C. Koopmans (ed.), *Activity Analysis of Production and Allocation*, July–October 1949, Cowles Commission Monograph 13, Proceedings of Linear Programming Conference, June 20–24, 1949, John Wiley and Sons, New York, 330–335.
- Dantzig, G.B. (1951e). "Linear Programming," in *Problems for the Numerical Analysis of the Future*, Proceedings of Symposium on Modern Calculating Machinery and Numerical Methods, UCLA, July 29–31, 1948, Appl. Math. Series 15, National Bureau of Standards, June 1951, 18–21.
- Dantzig, G.B. (1954a). "The Dual Simplex Algorithm (Notes on Linear Programming: Part VII)," *RM-1270*, The RAND Corporation, July.
- Dantzig, G.B. (1954b). "Upper Bounds, Secondary Constraints, and Block Triangularity in Linear Programming (Notes on Linear Programming: Part VIII, IX, X)," *RM-1367*, The RAND Corporation, October.
- Dantzig, G.B. (1954c). "Notes on Linear Programming: Part XI, Composite Simplex-Dual Simplex Algorithm—I," *RM-1274*, The RAND Corporation, April.

- Dantzig, G.B. (1954d). "A Comment on Eddie's 'Traffic Delays at Toll Booths,'" *Journal of Operations Research Society of America* **2**, 339–341.
- Dantzig, G.B. (1955a). "Linear Programming under Uncertainty," *Management Science* **1**, 197–206. Also in A.F. Veinott (ed.), *Mathematical Studies in Management Science*, Macmillan, New York, 1965, 330–339.
- Dantzig, G.B. (1955b). "Optimal Solution of a Dynamic Leontief Model with Substitution (Notes on Linear Programming: Part XIII)," *RM-1281-1*, The RAND Corporation, April. Also *Econometrica*, **23**, 151–176.
- Dantzig, G.B. (1955c). "Developments in Linear Programming," Proceedings Second Symposium on Linear Programming, National Bureau of Standards and Comptroller, U.S.A.F. Headquarters, January, 667–685.
- Dantzig, G.B. (1956a). "Constructive Proof of the Min-Max Theorem," *Pacific Journal of Mathematics* **6**, 25–33.
- Dantzig, G.B. (1956b). "Recent Advances in Linear Programming," *Management Science* **2**, 131–144.
- Dantzig, G.B. (1956c). "Formulating a Linear Programming Model," *P-893*, The RAND Corporation, July; also in *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963, Chapter 3.
- Dantzig, G.B. (1956d). "Note on Klein's 'Direct Use of Extremal Principles in Solving Certain Problems Involving Inequalities,'" *Operations Research* **4**, 247–249.
- Dantzig, G.B. (1957a). "Thoughts on Linear Programming and Automation," *Management Science* **3**, 131–139; also in *P-824*, The RAND Corporation, March 2, 1956.
- Dantzig, G.B. (1957b). "Concepts, Origins and Uses of Linear Programming," in Davis, Eddison, and Page (eds.), *Proceedings of First International Conference on Operations Research*, Operations Research Society of America, Baltimore, December, 100–108.
- Dantzig, G.B. (1957c). "Discrete Variable Extremum Problems," *Operations Research* **5**, 226–277.
- Dantzig, G.B. (1957d). "On The Status of Multi-Stage Linear Programs," *RM-1028*, The RAND Corporation, February; also in A.F. Veinott (ed.), *Mathematical Studies in Management Science*, Macmillan, New York, 1965, Chapter 6, Section III: Topics in Linear Programming, 303–320; also in *Management Science*, 1959, 71–90.
- Dantzig, G.B. (1958a). "Solving Two-Move Games with Perfect Information," *P-1459*, The RAND Corporation, August.
- Dantzig, G.B. (1958b). "On Integer and Partial Integer Linear Programming Problems," *P-1410*, The RAND Corporation, June.
- Dantzig, G.B. (1958c). "Chemical Equilibrium in Complex Mixtures," *Journal of Chemical Physics* **28**, 751–755.
- Dantzig, G.B. (1959). "Note on Solving Linear Programs in Integers," *Naval Research Logistics Quarterly*, **6**, 75–76.
- Dantzig, G.B. (1960a). "Inductive Proof of the Simplex Method," *IBM Journal of Research and Development* **4**, 505–506; also in *P-1851*, The RAND Corporation, December 28, 1959.

- Dantzig, G.B. (1960b). "On the Shortest Route through a Network," *Management Science* **6**, 187–190. Also in D.R. Fulkerson (ed.), *Some Topics in Graph Theory*, MAA Studies, No. 11, 1975, 89–93.
- Dantzig, G.B. (1960c). "On the Significance of Solving Linear Programming Problems with Some Integer Variables," *Econometrica*, **28**, 30–44.
- Dantzig, G.B. (1960d). "General Convex Objective Forms," in K. Arrow, S. Karlin, and P. Suppes (eds.), *Mathematical Methods in the Social Sciences*, Stanford University Press, Stanford, California, 151–158.
- Dantzig, G.B. (1960e). "A Machine-Job Scheduling Model," *Management Science* **6**, 191–196.
- Dantzig, G.B. (1961a). "Future Developments of Operations Research," Operations Research Center, University of California, Berkeley, Summary Report 1, 1961; also in *Proceedings Thirteenth Annual Industrial Engineering Institute*, University of California, February 1961, 25–28, Special Topic; also in *Decentralized Planning*, Journal of Operations Research Society, Switzerland, 1962, 53–55.
- Dantzig, G.B. (1961b). "Quadratic Programming—A Variant of the Wolfe-Markowitz Algorithms," Technical Report RR-2, Operations Research Center, University of California, Berkeley; also in *Linear Programming and Extensions*, Princeton University Press, Princeton, NJ, 1963, Chapter 24-4.
- Dantzig, G.B. (1962a). "A Proof of a Property of Leontief and Markov Matrices," Technical Report RR-25, Operations Research Center, University of California, Berkeley.
- Dantzig, G.B. (1962a). "Compact Basis Triangularization for the Simplex Method," Technical Report RR-28, Operations Research Center, University of California, Berkeley; also in R.L. Graves and P. Wolfe (eds.), *Recent Advances in Mathematical Programming*, McGraw-Hill, New York, 125–133.
- Dantzig, G.B. (1963). *Linear Programming and Extensions*, Princeton University Press, Princeton, New Jersey, August. Revised edition Fall 1966; fourth printing, 1968, 621 pages. [Japanese translation, Tutte-Mori, Inc., Tokyo, 1983.]
- Dantzig, G.B. (1964a). "New Mathematical Methods in the Life Sciences," *The American Mathematical Monthly* **71**, 4–15. Also in Stacy and Waxman (eds.), *Computers and Biomedical Research 1*, Academic Press, New York, March 1966.
- Dantzig, G.B. (1964b). "Research Problems," *Bulletin of the American Mathematical Society* **70**, 499–501.
- Dantzig, G.B. (1964c). "Linear Control Processes and Mathematical Programming," Technical Report 64-13, Operations Research Center, University of California, Berkeley; also in *SIAM Journal on Control and Optimization* **4**, 1966, 56–60; also in G.B. Dantzig, G.B. and A.F. Veinott, Jr. (eds.), *Mathematics of the Decision Sciences Part 2*, the American Mathematical Society Summer Seminar, Providence, RI, 1968, 31–36; also in J. Abadie (ed.), *Nonlinear Programming*, North-Holland, Amsterdam, the Netherlands, 1967, 283–286; also in A.K. Aziz (ed.), *Lecture Series in Differential Equations 1*, Van Nostrand Reinhold Publishing Co., 1969, 1–7.
- Dantzig, G.B. (1965a). "Operations Research in the World of Today and Tomorrow," Technical Report 65-7, Operations Research Center, University of California, Berkeley; also in Technion Yearbook; also in commemorative volume for Professor Keller, Institut für Ökonometri; presidential address, TIMS 1966, entitled "Management

- Science in the World of Today and Tomorrow” in *Management Science* **13**, February 1967, pages C-107–C-111.
- Dantzig, G.B. (1965b). “Large-Scale System Optimization: A Review,” Technical Report 65-9, Operations Research Center, University of California, Berkeley.
- Dantzig, G.B. (1965c). “Optimization in Operations Research,” Technical Report 65-10, Operations Research Center, University of California, Berkeley; also in in Wayne Kalenich (ed.), *Proceedings, International Federation for Information Processing Congress, 1965*, May.
- Dantzig, G.B. (1965c). “The Simplex Method,” in R. Machol (ed.), *Engineering Handbook*, McGraw-Hill, New York, June 1965, Chapter 25, 10 pages.
- Dantzig, G.B. (1966a). “Linear Programming and its Progeny,” Naval Research Reviews XIX (6), June 1966, 1; also in E.M.L. Beale (ed.), *Application of Mathematical Programming Techniques*, English Universities Press, Ltd., London, 1970, 3–16.
- Dantzig, G.B. (1966b). “All Shortest Routes in a Graph,” Technical Report 66-3, Department of Operations Research, Stanford University, Stanford, CA, November; also in *Théorie des Graphes, International Symposium, Rome, Italy, July 1966*, 91–92, published by DUNOD, Paris.
- Dantzig, G.B. (1966c). “On Positive Principal Minors,” Technical Report 67-1, Department of Operations Research, Stanford University, Stanford, CA, January.
- Dantzig, G.B. (1967). “Large-Scale Linear Programming,” Technical Report 67-8, Department of Operations Research, Stanford University, Stanford, CA, November; also in “Large-Scale Systems and the Computer Revolution,” in H.W. Kuhn (ed.), *Proceedings of the Princeton Symposium on Mathematical Programming*, Princeton University Press, Princeton, New Jersey, August 1967, 51–72. Also in G.B. Dantzig, and A. F. Veinott, Jr. (eds.), *Mathematics of the Decision Sciences*, the American Mathematical Society Summer Seminar, Providence, RI, 1968, 77–92.
- Dantzig, G.B. (1969a). “Complementary Spanning Trees,” Technical Report CS 126, Department of Computer Science, Stanford University, Stanford, CA, March; also in J. Abadie (ed.), *Integer and Nonlinear Programming*, North-Holland, Amsterdam, the Netherlands, 1970, 499–505.
- Dantzig, G.B. (1969b). “A Hospital Admission Problem,” Technical Report 69-15, Department of Operations Research, Stanford University, Stanford, CA, December.
- Dantzig, G.B. (1970a). “On a Model for Computing Round-Off Error of a Sum,” Technical Report STAN-CS-70-156, Department of Computer Science, Stanford University, Stanford, CA.
- Dantzig, G.B. (1970b). “A Control Problem of Bellman,” Technical Report 70-15, Department of Operations Research, Stanford University, Stanford, CA, September. Also in *Management Science: Theory* 16, May 1971, 542–546.
- Dantzig, G.B. (1972a). “Health Care in Future Cities,” Technical Report 72-22, Department of Operations Research, Stanford University, Stanford, CA, September.
- Dantzig, G.B. (1972b). “On the Relation of Operations Research to Mathematics,” Technical Report 72-23, Department of Operations Research, Stanford University, Stanford, CA, October.

- Dantzig, G.B. (1972c). "The Price Lecture on Compact City," Price Lecture Series University of Pittsburgh.
- Dantzig, G.B. (1973a). "The ORSA New Orleans Address on Compact City," *Management Science* **19**, 1151-1161.
- Dantzig, G.B. (1973b). "Solving Staircase Linear Programs by a Nested Block-Angular Method," Technical Report 73-1, Department of Operations Research, Stanford University, Stanford, CA, January.
- Dantzig, G.B. (1973c). "Drews' Institutionalized Divvy Economy," Technical Report 73-7, Department of Operations Research, Stanford University, Stanford, CA, September. Revised, International Institute for Applied Systems Analysis, December 1973. Revised 1974, as Technical Report TR 74-14. Also in "An Institutionalized Divvy Economy," Technical Report SOL 75-17 (revision of TR 74-14), Department of Operations Research, Stanford University, Stanford, CA. Also in *Journal of Economic Theory* **11**, 1975, 372-384.
- Dantzig, G.B. (1974a). "On a Convex Programming Problem of Rozanov," *Applied Mathematics and Optimization* **1**, 189-192; also entitled "A Generalized Programming Solution to a Convex Programming Problem with a Homogeneous Objective," IIASA Research Report RR-73-21, December 1973; also in *Symposia Mathematica*, Monograf, Italy, Vol. XIX, Academic Press, 1976, 209-214.
- Dantzig, G.B. (1974b). "On the Reduction of an Integrated Energy and Interindustry Model to a Smaller Linear Program," Technical Report SOL 74-20, Department of Operations Research, Stanford University, Stanford, CA, December; also in *The Review of Economics and Statistics* **LVIII**, May 1976, 248-250.
- Dantzig, G.B. (1974c). "Formulating a PILOT Model for Energy in Relation to the National Economy," Technical Report SOL 75-10, Department of Operations Research, Stanford University, Stanford, CA, April.
- Dantzig, G.B. (1975). "Note on the Objective Function for the PILOT Model," Technical Report SOL 75-20, Department of Operations Research, Stanford University, Stanford, CA, August; also in A. Prekopa (ed.), *Survey of Mathematical Programming*, Proceedings, IX International Symposium on Mathematical Programming, Publishing House of the Hungarian Academy of Sciences, Budapest, 1980, 325-328.
- Dantzig, G.B. (1976). "Linear Programming: Its Past and Its Future," in Salkovitz, Edward I. (ed.), *Science Technology, and the Modern Navy, Thirtieth Anniversary*, ONR-37, Office of Naval Research, 84-95.
- Dantzig, G.B. (1977). "Large-Scale Systems Optimization with Application to Energy," Technical Report SOL 77-3, Department of Operations Research, Stanford University, Stanford, CA, April; also in Proceedings of Symposia in Applied Mathematics, American Mathematical Society, Providence, RI.
- Dantzig, G.B. (1978). "Are Dual Variables Prices? If Not, How to Make them More So," Technical Report SOL 78-6, Department of Operations Research, Stanford University, Stanford, CA, March; also in Franco Angeli (ed.), *Mathematical Programming and its Economics Applications*, Milano, Italy, 1981, 135-148.
- Dantzig, G.B. (1979a). "The Role of Models in Determining Policy for Transition to a More Resilient Technological Society," IIASA Distinguished Lecture Series /1, Vienna, June 12, International Institute for Applied Systems Analysis, Laxenburg, 1979.

- Dantzig, G.B. (1979b). "Comments on Khachian's Algorithms for Linear Programming," Technical Report SOL 79-22, Department of Operations Research, Stanford University, Stanford, CA, November. Also in *SiamNews* **13**, October 1980.
- Dantzig, G.B. (1980a). "Expected Number of Steps of the Simplex Method for a Linear Program with a Convexity Constraint," Technical Report SOL 80-3, Department of Operations Research, Stanford University, Stanford, CA, March (revised October 1980).
- Dantzig, G.B. (1980b). "Time-Staged Methods in Linear Programming; Comments and Early History, Technical Report SOL 80-18, Department of Operations Research, Stanford University, Stanford, CA, June; also in G.B. Dantzig, M.A.H. Dempster, and M.J. Kallio. (eds.), *Large-Scale Linear Programming*, Vol. 2, CP-81-51, IIASA Collaborative Proceedings Series, Laxenberg, Austria, 1981, 3-16; also in "Large Scale Systems.," in Haims, Y.Y. (ed.), *Studies in Management Science and Systems*, Vol. 7, North-Holland Publishing Company, Amsterdam, 1982, 19-30.
- Dantzig, G.B. (1980c). "Time-Staged Linear Programs," Technical Report SOL 80-28, Department of Operations Research, Stanford University, Stanford, CA, October.
- Dantzig, G.B. (1981a). "Reminiscences About the Origins of Linear Programming," Technical Report SOL 81-5, Department of Operations Research, Stanford University, Stanford, CA, April; also in Contemporary Mathematics, American Mathematical Society; also in R.W. Cottle, M.L. Kelmanson, B. Korte (eds.), *Mathematical Programming*, Proceedings of the International Congress on Mathematical Programming, Rio de Janeiro, Brazil, April 6-8, 1981, North-Holland Publishing Co., Amsterdam, 1984, 105-112; also in *Operations Research Letters* **1**, 43-48, April 1982. Also in A. Bachem, M. Grotschel, and B. Korte (eds.), *Mathematical Programming: The State of the Art, Bonn 1982*, Springer-Verlag, Berlin and New York, 78-86. Also in *Australian Society for OR Bulletin*, 1986.
- Dantzig, G.B. (1981b). "Concerns About Large-Scale Models," Technical Report SOL 81-27, Department of Operations Research, Stanford University, Stanford, CA, December; also in *Large-Scale Energy Models*, AAAS Selected Symposia Series 73, Westview Press, Inc. Boulder, CO, for the Amer. Assoc. for the Adv. of Sc., Washington, DC, 15-20, 1983.
- Dantzig, G.B. (1981c). "The PILOT Energy-Economic Model for Policy Planning," Technical Report SOL 81-26, Department of Operations Research, Stanford University, Stanford, CA, December; also in T. Nejat Veziroglu (ed.), *Energy Programs Policy Economics*, Proceedings of the 4th Miami International Conference on Alternative Energy Sources, Volume 8, Ann Arbor Science Publishers, Ann Arbor Michigan, 1982, 409-415.
- Dantzig, G.B. (1981d). "Contributions of Mathematics to Planning During and Immediately After World War II," working paper, November. Also in *History of Mathematics in World War II*, MAA Series.
- Dantzig, G.B. (1982a). "Time-Staged Methods in Linear Programs," in Y.Y. Haims (ed.), *Studies in Management Science*, Vol. 7: *Large-Scale Systems*, North-Holland, Amsterdam, the Netherlands, 19-30.
- Dantzig, G.B. (1982b). "Mathematical Programming and Decision Making in a Technological Society," Technical Report SOL 82-11, Department of Operations Research, Stanford University, Stanford, CA, August; also in *Information Processing Society of Japan* **24**, May 1983 (in Japanese).

- Dantzig, G.B. (1983). "Can Leontief and P-Matrices be Rescaled Positive Definite," Technical Report SOL 83-23, Department of Operations Research, Stanford University, Stanford, CA, November.
- Dantzig, G.B. (1984). "Economic Growth and Dynamic Equilibrium," Technical Report SOL 84-8, Department of Operations Research, Stanford University, Stanford, CA, October.
- Dantzig, G.B. (1985a). "Deriving a Utility Function for the Economy," Technical Report SOL 85-6, Department of Operations Research, Stanford University, Stanford, CA, June; revised April 1986.
- Dantzig, G.B. (1985b). "Impact of Linear Programming on Computer Development," Technical Report SOL 85-7, Department of Operations Research, Stanford University, Stanford, CA. Revised version, Stanford University, July 1986, in D.V. Chudnovsky and R.D. Jenks (eds.), *Proceedings Computers in Mathematics*, Lecture Notes in Pure and Applied Mathematics, Marcel Dekker, Inc., 1990, 233–240; also in *ORMS Today* **14**, August 1988, 12–17.
- Dantzig, G.B. (1986). "Need to do Planning under Uncertainty and the Possibility of Using Parallel Processors for this Purpose," Technical Report SOL 86-11, Department of Operations Research, Stanford University, Stanford, CA, April.
- Dantzig, G.B. (1987a). "Simplex Method for Solving Linear Programs," *The New Palgrave: A Dictionary of Economic Theory and Doctrine*, Macmillan, London.
- Dantzig, G.B. (1987b). "Linear Programming," *The New Palgrave: A Dictionary of Economic Theory and Doctrine*, Macmillan, London, to appear.
- Dantzig, G.B. (1987c). "Planning Under Uncertainty Using Parallel Computing," Technical Report SOL 87-1, Department of Operations Research, Stanford University, Stanford, CA, January; also in *Annals of Operations Research* **14**, 1988, 1–16.
- Dantzig, G.B. (1987d). "Origins of the Simplex Method," Technical Report SOL 87-5, Department of Operations Research, Stanford University, Stanford, CA, May; also in Nash, S.G. (ed.), *Proceedings of the ACM Conference on a History of Scientific Computing*, ACM Press, Addison-Wesley Publishing Company, 1990, 141–151.
- Dantzig, G.B. (1988a). "Dikin's Interior Method for Solving LP," manuscript, Department of Operations Research, Stanford University, Stanford, CA.
- Dantzig, G.B. (1988b). "Making Progress During a Stall in the Simplex Algorithm," Technical Report SOL 88-5, Department of Operations Research, Stanford University, Stanford, CA, February; also in *Linear Algebra and Its Applications* **114/115**, 1989, 251–259.
- Dantzig, G.B. (1989). "Decomposition Techniques for Large-Scale Electric Power Systems Planning Under Uncertainty," in R. Sharda, B.L. Golden, E. Wasil, O. Balci, and W. Steward (eds.), *Impact of Recent Computer Advances on Operations Research* North Holland, 3–20.
- Dantzig, G.B. (1990). "The Diet Problem," *Interfaces* **20**:4, July/Aug, 43–47.
- Dantzig, G.B. (1991). "Converting a Converging Algorithm into a Polynomial Bounded Algorithm," Technical Report SOL 91-5, Department of Operations Research, Stanford University, Stanford, CA, March.

- Dantzig, G.B. (1992a). "An ϵ -Precise Feasible Solution to a Linear Program with a Convexity Constraint in $1/\epsilon^2$ Iterations Independent of Problem Size," Technical Report SOL 92-5, Department of Operations Research, Stanford University, Stanford, CA, October.
- Dantzig, G.B. (1992b). "Bracketing to Speed Convergence Illustrated on the von Neumann Algorithm for Finding a Feasible Solution to a Linear Program with a Convexity Constraint," Technical Report SOL 92-6, Department of Operations Research, Stanford University, Stanford, CA, October.
- Dantzig, G.B. (1995). Working paper.
- Dantzig, G.B., Cottle, R.W., Eaves, B.C., Golub, G.H., Hillier, F.S., Manne, A.S., Wilde, D.J., and Wilson, R.B. (1973). "On The Need for a Systems Optimization Laboratory," in T.C. Hu and S.M. Robinson (eds.), *Mathematical Programming*, Proceedings of an Advanced Seminar Conducted by the Mathematics Research Center, University of Wisconsin, September 1972, Academic Press, London and New York, 1-31; previously published in September 1972 in Technical Report 72-11, Department of Operations Research, Stanford University, Stanford, CA, September.
- Dantzig, G.B. and Adler, I. (1971). "Maximum Diameter of Abstract Polytopes," Technical Report 71-12, Department of Operations Research, Stanford University, Stanford, CA, August; also in Harry Williams (ed.), *IDA Economics Papers*, January 1972; also in *Mathematical Programming Study* 1, 1974, 20-40.
- Dantzig, G.B., Adler, I., and Murty, K. (1970). "Existence of x-Paths in Abstract Polytopes," Technical Report 70-1, Department of Operations Research, Stanford University, Stanford, CA, March.
- Dantzig, G.B., Adler, I., and Murty, K. (1974). "Existence of A-Avoiding Paths in Abstract Polytopes," *Mathematical Programming Study* 1, 41-42.
- Dantzig, G.B., Avi-Itzhak, B., Connolly, T.J., McAllister, P.H., and Winkler, W.D. (1982a). "A Dynamic Equilibrium Model for Energy-Economic Planning," Technical Report SOL 82-2, Department of Operations Research, Stanford University, Stanford, CA, March.
- Dantzig, G.B., Avi-Itzhak, B., Connolly, T.J., McAllister, P.H., and Winkler, W.D. (1982b). "Mathematical Appendix: A Dynamic Equilibrium Model for Energy-Economic Planning," Technical Report SOL 82-3, Department of Operations Research, Stanford University, Stanford, CA, March.
- Dantzig, G.B., Avi-Itzhak, and Iusem, A. (1983). "The Consumers Energy Services Model of the PILOT System," in Benjamin Lev (ed.), *Energy Models and Studies*, Studies in Management Science and Systems 9, North-Holland Publishing Co., Amsterdam, 195-220.
- Dantzig, G.B., Beale, E.M.L., Watson, R.D. (1986). "A First Order Approach to a Class of Multi-Time-Period Stochastic Programming Problems," *Mathematical Programming Study* 27, 103-117.
- Dantzig, G.B. and Avriel, M. (1976). "Determining Prices and Monetary Flows of the PILOT Energy Model," Technical Report SOL 76-28, Department of Operations Research, Stanford University, Stanford, CA, October.

- Dantzig, G.B., Bigelow, J., Golub, G., Gordon, R., Montalbano, M., Pinsky, P., Sahberwal, F., Wirth, N., and Witzgall, C. (1967). "Mathematical Programming Language," Technical Report 67-4, Department of Operations Research, Stanford University, Stanford, CA, June.
- Dantzig, G.B., Bigelow, J., Golub, G., Gordon, R., Montalbano, M., Pinsky, P., Sahberwal, F., Wirth, N., and Witzgall, C. (1968). "Mathematical Programming Language," Technical Report CS 119, Department of Computer Science, Stanford University, Stanford, CA.
- Dantzig, G.B. and Blattner, W.O., and Rao, M.R. (1966a). "Finding a Cycle in a Graph with Minimum Cost to Time Ratio with Application to a Ship Routing Problem," Technical Report 66-1, Department of Operations Research, Stanford University, Stanford, CA, November; also in *Theorie Des Graphes, International Symposium, Rome, Italy, July 1966*, 77–84, published by DUNOD, Paris.
- Dantzig, G.B. and Blattner, W.O., and Rao, M.R. (1966b). "All Shortest Routes from a Fixed Origin in a Graph," Technical Report 66-2, Department of Operations Research, Stanford University, Stanford, CA, November; also in *Theorie Des Graphes, International Symposium, Rome, Italy, July 1966*, 85–90, published by DUNOD, Paris.
- Dantzig, G.B., Connolly, T.J., and Parikh, S.C. (1977). "Stanford PILOT Energy/Economic Model," Technical Report SOL 77-19, Department of Operations Research, Stanford University, Stanford, CA, July; also in *Advances in the Economics of Energy and Resources, Volume 1 – The Structure of Energy Markets*, JAI Press, 1979, 77–103; also in El Mallakh, R. and El Mallakh, D.H. (eds.), *Proceedings of the 4th International Conference on Energy Options and Conservation*, October 17–19, 1977, The International Research Center for Energy and Economic Development, Boulder, Colorado, 1978, 87–119. *Policy Analysis and Information Systems* 2, 1978, 23–51.
- Dantzig, G.B., Connolly, T.J., and Parikh, S.C. (1978). "Stanford PILOT Energy/Economic Model," EA-626, Volumes 1 and 2, Interim Report, Electric Power Research Institute, Palo Alto, California, May.
- Dantzig, G.B., Connolly, T.J., Parikh, S.C., Riddell, J.M. (1978). "A Description and Demonstration of the Stanford PILOT Energy/Economic Model," *Stanford PILOT Energy/Economic Model*, EA-626, Volumes 1 & 2, Electric Power Research Institute, May, 1–40; also in *Proceedings of Second US-USSR Symposium on Econometric Modeling*, Skyland, Virginia, 1978, University of Maryland, College Park, Maryland 1980.
- Dantzig, G.B. and Cottle, R.W. (1963). "Positive (Semi-) Definite Matrices and Mathematical Programming," Technical Report RR-35, Operations Research Center, University of California, Berkeley; also in "Positive (Semi-) Definite Programming," in J. Abadie (ed.), *Nonlinear Programming*, North-Holland, Amsterdam, the Netherlands, 55–73.
- Dantzig, G.B. and Cottle, R.W. (1967). "Complementary Pivot Theory of Mathematical Programming," Technical Report 67-2, Department of Operations Research, Stanford University, Stanford, CA, April; also in G.B. Dantzig, and A. F. Veinott, Jr. (eds.), *Mathematics of the Decision Sciences*, the American Mathematical Society Summer Seminar, Providence, RI, 1968; also in *Linear Algebra and Its Applications* 1, 103–125; also in G. B. Dantzig and B. C. Eaves (eds.), *Studies in Optimization*,

- MAA Studies in Mathematics, Vol. 10, Mathematical Association of America, 1974, 27–51.
- Dantzig, G.B. and Cottle, R.W. (1968). “A Generalization of the Linear Complementarity Problem,” Technical Report 68-9, Department of Operations Research, Stanford University, Stanford, CA; also in *Journal of Combinatorial Theory* **8**, January 1970, 79–90.
- Dantzig, G.B. and Cottle, R.W. (1974). “Optimization, Mathematical Theory of (Linear and Nonlinear Programming),” *Encyclopaedia Britannica*, Vol. 13, 628–632.
- Dantzig, G.B., Collen, M.F., Rubin, L., Neyman J., Baer, R.M., and Siegelau, A.B. (1964). “Automated Multiphase Screening and Diagnosis,” *American Journal of Public Health* **54**, 741–750.
- Dantzig, G.B. and DeHaven, J. (1961). “On The Reduction of Certain Multiplicative Chemical Equilibrium Systems to Mathematically Equivalent Additive Systems,” *P-2419*, The RAND Corporation, August; also *Journal of Chemical Physics* **36**, May, 1962, 2620–2627.
- Dantzig, G.B., DeHaven, J., and Sams, C.F. (1960). “A Mathematical Model of the Respiratory System,” in Proceedings, Fourth Air Pollution Medical Research Conference, San Francisco, December, 72–95; also in *P-2048*, The RAND Corporation, July 1960.
- Dantzig, G.B., DeHaven, J., and Sams, C.F. (1961a). “A Mathematical Model of the Chemistry of the External Respiratory System,” in J. Neyman (ed.), *Proceedings 4th Berkeley Symposium on Mathematical Statistics and Probability, 1961*, University of California Press, Berkeley, California, 181–196.
- Dantzig, G.B., DeHaven, J., and Sams, C.F. (1961b). “A Mathematical Model of the Human External Respiratory System,” *Perspectives of Biology and Medicine* IV (3), Spring 1961, 324–376.
- Dantzig, G.B., Dempster, M.A.H., and Kallio, M.J. (eds.), (1981). *Large-Scale Linear Programming*, Vol. 1, CP-81-51, IIASA Collaborative Proceedings Series, Laxenberg, Austria.
- Dantzig, G.B., Dempster, M.A.H., and Kallio, M.J., editors (1981). *Large-Scale Linear Programming*, Vol. 2, CP-81-51, IIASA Collaborative Proceedings Series, Laxenberg, Austria.
- Dantzig, G.B. and Eaves, B.C. (1972). “Fourier-Motzkin Elimination and Its Dual,” Technical Report 72-18, Department of Operations Research, Stanford University, Stanford, CA, June 1972; also in *Journal of Combinatorial Theory* **14**, May 1973, 288–297; also in B. Roy (ed.), *Combinatorial Programming: Methods and Applications*, D. Reidel Publishing Co., Boston, 1975, 93–102.
- Dantzig, G.B. and Eaves, B.C., editors (1974). *Studies in Optimization, MAA Studies in Mathematics*, Vol. 10, Mathematical Association of America.
- Dantzig, G.B., Eaves, B.C., and Gale, D. (1976). “An Algorithm for a Piecewise Linear Model of Trade and Production with Negative Prices and Bankruptcy,” Technical Report SOL 76-19, Department of Operations Research, Stanford University, Stanford, CA; also in *Mathematical Programming* **16**, 1979, 190–209.

- Dantzig, G.B., Eaves, B.C., and Rothblum, U. (1983). "A Decomposition and Scaling-inequality for Line-sum-symmetric Nonnegative Matrices," Technical Report SOL 83-21, Department of Operations Research, Stanford University, Stanford, CA, December; also in *SIAM Journal on Discrete Mathematics* **6**, April 1985.
- Dantzig, G.B., Eisenberg, E., and Cottle, R.W. (1962). "Symmetric Dual Nonlinear Programs," Technical Report RR-35, Operations Research Center, University of California, Berkeley; also in *Pacific Journal of Mathematics* **15**, 1965, 809–812.
- Dantzig, G.B., Eisenstat, S., Magnanti, T.L., Maier, S., McGrath, M., Nicholson, V., and Reidl, C. (1970). "MPL: Mathematical Programming Language Specification Manual for Committee Review," Technical Report STAN-CS-70-187, Department of Computer Science, Stanford University, Stanford, CA.
- Dantzig, G.B., Folkman, J., and Shapiro, M. (1965). "On the Continuity of the Minimum Set of a Continuous Function," The RAND Corporation, June; also in *Journal of Mathematical Analysis and Applications* **17**, March 1967, 519–548.
- Dantzig, G.B., Ford, L.R., and Fulkerson, D.R. (1956). "A Primal-Dual Algorithm for Linear Programs," in H.W. Kuhn and A.W. Tucker (eds.), *Linear Inequalities and Related Systems*, Annals of Mathematics Study No. 38, Princeton University Press, Princeton, New Jersey, 171–181; also, *RM-1709*, The RAND Corporation, May.
- Dantzig, G.B., Friel, J., Golightly, R., Harvey, R.P., and McKnight, R.D. (1975). "Solution of a Large-Scale Air Force Ordnance Planning Problem by Mathematical Programming," Proceedings of the Fire Support Requirements Methodology Workshop, Ketrion, Inc., August.
- Dantzig, G.B. and Fulkerson, D.R. (1955). "Computation of Maximal Flows in Networks," *Naval Research Logistics Quarterly*, **2**, 277–283.
- Dantzig, G.B. and Fulkerson, D.R. (1954). "Minimizing the Number of Tankers to Meet a Fixed Schedule," *Naval Research Logistics Quarterly*, **1**, 217–222.
- Dantzig, G.B. and Fulkerson, D.R. (1956). "On the Max-Flow Min-Cut Theorems of Networks," in H.W. Kuhn and A.W. Tucker (eds.), *Linear Inequalities and Related Systems*, Annals of Mathematics Study No. 38, Princeton University Press, Princeton, New Jersey, 215–221.
- Dantzig, G.B., Fulkerson, D.R., and Johnson, S.M. (1954). "Solution for a Large-Scale Traveling Salesman Problem," *Journal of Operations Research Society of America* **2**, 393–410.
- Dantzig, G.B., Fulkerson, D.R., and Johnson, S.M. (1959). "On a Linear Programming Combinatorial Approach to the Traveling Salesman Problem," *Operations Research* **7**, 58–66.
- Dantzig, G.B. and Glynn, P.W. (1990). "Parallel Processors for Planning Under Uncertainty," *Annals of Operations Research* **22**, 1–21; also in Technical Report SOL 88-8, Department of Operations Research, Stanford University, Stanford, CA, June.
- Dantzig, G.B., Glynn, P.W., Avriel, M., Stone, J.C., Entriken, R., and Nakayama, M. (1989). "Decomposition Techniques for Multi-Area Transmission Planning Under Uncertainty," Report EL-6484, Electrical Systems Division, Electric Power Research Institute, Palo Alto, California.

- Dantzig, G.B., Harvey, R.P., McKnight, R.D., and Smith S.S. (1969). "Sparse Matrix Techniques in Two Mathematical Programming Codes," Technical Report 69-1, Department of Operations Research, Stanford University, Stanford, CA; also in D.J. Rose and R.A. Willoughby (eds.), *Sparse Matrices and their Applications*, Plenum Press, New York, , Proceedings of the Symposium on Sparse Matrices and Their Applications IBM RA-1, IBM Watson Research Center, September 1968), March 1969, 85–100.
- Dantzig, G.B., Harvey, R.P., Lansdowne, Z.F., Maier, S.F., and Robinson, D.W. (1977a). "Computational Experience with a Continuous Network Design Code," GSBA Working Paper 236, Graduate School of Business Administration, Duke University, NC, December.
- Dantzig, G.B., Harvey, R.P., Lansdowne, Z.F., Maier, S.F., and Robinson, D.W. (1977b). "Formulating and Solving the Network Design Problem by Decomposition," GSBA Working Paper 215, Graduate School of Business Administration, Duke University, Durham, NC, January 1977; also in "A Convex Network Design Model Based on a Decomposition Procedure," *Transportation Research B*, **13B**, 1979, 5–17,
- Dantzig, G.B., Harvey, R.P., Lansdowne, Z.F., and Muth, R. (1979). "Framework for a System of Transportation Spatial Form Research Tools," Report DOT-TSC-RSPA-79-12, Final Report to U.S. Department of Transportation, Washington, DC, April.
- Dantzig, G.B., Harvey, R.P., Lapin, L.L., and Uren, J. (1966). "An Integer Branching Algorithm for Locating Warehouses," Standard Oil Company of California Report, Operations Research Division, October (revised December 1968).
- Dantzig, G.B., Harvey, R.P., and McKnight, R. (1964). "Updating the Product Form of the Inverse for the Revised Simplex Method," Technical Report 64-33, Operations Research Center, University of California, Berkeley; also in Association for Computing Machinery Proceedings, August 1965; and Abstract in Journal of the Association for Computing Machinery, October 1965.
- Dantzig, G.B., Hax, R., Pomeroy, R., Sanderson, R., and van Slyke, R. [with contributions by G. Buerk, I. Durrer, B. Laurent, S. Mukerjee] (1970). "Natural Gas Transmission System Optimization," American Gas Association, Inc., April.
- Dantzig, G.B. and Hirsch, W. (1954). "The Fixed Charge Problem," *P-648*, The RAND Corporation; also in *Naval Research Logistics Quarterly*, **15**, 413–424.
- Dantzig, G.B., Ho, J.K., and Infanger, G. (1991). "Solving Stochastic Linear Programs on a Hypercube Multicomputer," Technical Report SOL 91-10, Department of Operations Research, Stanford University, Stanford, CA, August.
- Dantzig, G.B. and Hoffman, A. (1956). "Dilworth's Theorem on Partially Ordered Sets," in H.W. Kuhn and A.W. Tucker (eds.), *Linear Inequalities and Related Systems*, Annals of Mathematics Study No. 38, Princeton University Press, Princeton, New Jersey, 207–213; also in "New Directions in Mathematical Programming," RAND Symposium on Mathematical Programming, March 16-20, 1959, RAND R-351, page 1.
- Dantzig, G.B. Hoffman, A.J., and Hu, T.C. (1983). "Triangulations (Tilings) and Certain Block Triangular Matrices," Technical Report SOL 83-17, Department of Operations Research, Stanford University, Stanford, CA, September; also in *Mathematical Programming* **31**, 1985, 1–14.

- Dantzig, G.B. and Holling, C.S. (1974). "Determining Optimal Policies for Ecosystems," Technical Report 74-11, Department of Operations Research, Stanford University, Stanford, CA, August.
- Dantzig, G.B., Holling, C.S., Baskerville, C., Jones, D.D., and Clark, W. C. (1975). "A Case Study of Forest Ecosystem/Pest Management," Prepared for Proceedings International Canadian Conference on Applied Systems Analysis, 1975, WP-75-60, International Institute for Applied Systems Analysis, Laxenburg, Austria, June 1975.
- Dantzig, G.B., Holling, C.S., Clark, W.C., Jones, D.D., Baskerville, G., and Peterman, R.M. (1976). "Quantitative Evaluation of Pest Management Options: The Spruce Budworm Case Study," in D.L. Wood (ed.), *Proceedings of the XVth International Congress of Entomology*, August; also in W.E. Waters (ed.), *Current Topics in Forest Entomology*, U.S. Government Printing Office, Washington, DC, February 1979, 82-102.
- Dantzig, G.B., Holling, C.S., and Winkler, C. (1986). "Determining Optimal Policies for Ecosystems," *TIMS Studies in the Management Sciences* **21**, 1986, 453-473.
- Dantzig, G.B. and Infanger, G. (1992a). "Large-Scale Stochastic Linear Programs: Importance Sampling and Benders Decomposition," in C. Brezinski and U. Kulisch (eds.), *Computational and Applied Mathematics I—Algorithms and Theory*, Proceedings of the 13th IMACS World Congress, Dublin, Ireland, July 22-26, 1991, North Holland, 111-120; also Technical Report SOL 91-4, Department of Operations Research, Stanford University, Stanford, CA.
- Dantzig, G.B. and Infanger, G. (1992b). "Approaches to Stochastic Programming with Applications to Electric Power Systems," in K. Frauendorfer, H. Glavitsch, and R. Bacher (eds.), *Optimization in Planning and Operations of Electric Power Systems*, Lecture Notes of the SVOR/ASRO Tutorial, October 15-16, 1992, Thun, Switzerland, Physica Verlag, 125-138.
- Dantzig, G.B. and Infanger, G. (1993). "Multi-Stage Stochastic Linear Programs for Portfolio Optimization," *Annals of Operations Research* **45**, 59-76; also in Technical Report SOL 91-11, Department of Operations Research, Stanford University, Stanford, CA, September; also in *Proceedings of the Annual Symposium of RAMP (Research Association on Mathematical Programming)*, Tokyo, 1991.
- Dantzig, G.B. and Iusem, A. (1981). "Analyzing Labor Productivity Growth with the PILOT Model," Technical Report SOL 81-4, Department of Operations Research, Stanford University, Stanford, CA, March; also in Energy, Productivity and Economic Growth, A Workshop sponsored by the Electric Power Research Institute, Oelgeschlager, Gunn & Hain, Cambridge, Mass., 1983, 347-366.
- Dantzig, G.B. and Jackson, P. (1979). "Pricing Underemployed Capacity in a Linear Economic Model," Technical Report SOL 79-2, Department of Operations Research, Stanford University, Stanford, CA, February; also in R. W. Cottle, F. Giannessi, and J.L. Lions (eds.), *Variational Inequalities and Complementarity Problems: Theory and Applications*, John Wiley and Sons, Ltd., London, 1980, 127-134.
- Dantzig, G.B. and Johnson, D.L. (1963). "Maximum Payloads per Unit Time Delivered Through an Air Network," Report D1-82-0265, Boeing Scientific Research Laboratories, June; also in *Operations Research* **12**, 230-236.

- Dantzig, G.B. and Johnson, S. (1955). "A Production Smoothing Problem," Proceedings, Second Symposium on Linear Programming, National Bureau of Standards and Comptroller, U.S.A.F. Headquarters, January, 151–176.
- Dantzig, G.B., Johnson, S., and Wayne, W. (1958). "A Linear Programming Approach to the Chemical Equilibrium Problem," *Management Science* **5**, 38–43.
- Dantzig, G.B., Kawaratani, T.K., and Ullman, R.J. (1960). "Computing Tetraethyl-Lead Requirements in a Linear-Programming Format," *Operations Research* **8**, 24–29.
- Dantzig, G.B., Leichner, S.A., and Davis, J.W. (1992). "A Strictly Improving Phase I Algorithm Using Least-Squares Subproblems," Technical Report SOL 92-1, Department of Operations Research, Stanford University, Stanford, CA, April.
- Dantzig, G.B., Levin, S., and Bigelow, J. (1965). "On Steady-State Intercompartmental Flows," Technical Report 65-26, Operations Research Center, University of California, Berkeley; also in *Journal of Colloid and Interface Science* **23**, April 1967, 572–576.
- Dantzig, G.B. and Madansky, A. (1961). "On the Solution of Two-Stage Linear Programs Under Uncertainty," in J. Neyman (ed.), *Proceedings 4th Berkeley Symposium on Mathematical Statistics and Probability, 1961*, University of California Press, Berkeley, California, 165–176; also in *P-2039*, The RAND Corporation.
- Dantzig, G.B., Magnanti, T.L., and Maier, S. (1972). "The User's Guide to MPL/T.1 (Revised)," GSBA Working Paper 76, Graduate School of Business Administration, Duke University, Durham, NC, December 1972.
- Dantzig, G.B., Maier, S.F., and Lansdowne, Z.F. (1976). "The Application of Decomposition to Transportation Network Analysis," DOT Report, Control Analysis Corporation, Palo Alto, California, March.
- Dantzig, G.B. and Manne, A. (1974). "A Complementarity Algorithm for an Optimal Capital Path with Invariant Proportions," International Institute for Applied Systems Analysis, Laxenburg, Austria; also in Technical Report 74-1, Department of Operations Research, Stanford University, Stanford, CA, March; also in *Journal of Economic Theory* **9**, November, 312–323.
- Dantzig, G.B., McAllister, P.H., and Stone, J.C. (1985). "Changes Made for the PILOT-1983 Model," Technical Report SOL 85-12, Department of Operations Research, Stanford University, Stanford, CA, July.
- Dantzig, G.B., McAllister, P.H., and Stone, J.C. (1988a). "Deriving a Utility Function for the U.S. Economy," Technical Report SOL 88-6, Department of Operations Research, Stanford University, Stanford, CA, April. Parts I, II, III, in *Journal for Policy Modeling* **11**, 1989, 391–424 and Parts IV, V in *Journal for Policy Modeling* **11**, 1989, 569–592.
- Dantzig, G.B., McAllister, P.H., and Stone, J.C. (1988b). "Analyzing the Effects of Technological Change: A Computational General Equilibrium Approach," Technical Report SOL 88-12, Department of Operations Research, Stanford University, Stanford, CA, July.
- Dantzig, G.B., McAllister, P.H., and Stone, J.C. (1990). "An Interactive Model Management System: User Interface and System Design," Technical Report SOL 90-3, Department of Operations Research, Stanford University, Stanford, CA, January.

- Dantzig, G.B. and Orchard-Hays, W. (1953). "Alternate Algorithm for the Revised Simplex Method Using Product Form for the Inverse (Notes on Linear Programming: Part V)," *RM-1268*, The RAND Corporation, November.
- Dantzig, G.B. and Orchard-Hays, W. (1954). "The Product Form for the Inverse in the Simplex Method," *Mathematical Tables and Other Aids to Computation VIII*, April, 64–67.
- Dantzig, G.B. and Orden, A. (1952). "A Duality Theorem Based on the Simplex Method," Symposium on Linear Inequalities and Programming, Report 10, Project SCOOP, Planning Research Division, Director of Management Analysis Service, Comptroller, U.S.A.F. Headquarters, April, 51–55.
- Dantzig, G.B. and Orden, A. (1953). "Duality Theorems (Notes on Linear Programming: Part II)," *RM-1526*, The RAND Corporation.
- Dantzig, G.B., Orden, A., and Wolfe, P. (1955). "The Generalized Simplex Method for Minimizing a Linear Form Under Linear Inequality Constraints (Notes on Linear Programming: Part I)," *Pacific Journal of Mathematics* **5**, 183–195; also in *RM-1264*, The RAND Corporation, April 5, 1954.
- Dantzig, G.B. and Pace, N. (1963a). "Molecular-Sized Channels and Flows Against the Gradient," Technical Report 63-14, Operations Research Center, University of California, Berkeley.
- Dantzig, G.B. and Pace, N. (1963b). "A Model for Sodium-Potassium Transport in Red Cells," Technical Report 63-26, Operations Research Center, University of California, Berkeley.
- Dantzig, G.B. and Parikh, S.C. (1975). "On a PILOT Linear Programming Model for Assessing Physical Impact on the Economy of a Changing Energy Picture," Technical Report SOL 75-14, Department of Operations Research, Stanford University, Stanford, CA, June 1975 (revised SOL 75-14R, August 1975); also in F.S. Roberts (ed.), *Energy: Mathematics and Models*, Proceedings of a SIMS Conference on Energy, held at Alta, Utah, July 7–11, 1975, SIAM, 1976, 1–23; also in IIASA Conference '76, May 10–13 1976, 183–200; also in Proceedings of Symposia in Applied Mathematics, Vol. 21, American Mathematical Society, 1977, pp. 93–106.
- Dantzig, G.B. and Parikh, S.C. (1976). "Energy Models and Large-Scale Systems Optimization," Technical Report SOL 76-23, Department of Operations Research, Stanford University, Stanford, CA, November; also in W.W. White (ed.), *Computers and Mathematical Programming*, Proceedings of the Bicentennial Conference on Mathematical Programming, November 1976; also in NBS Special Publication 502, February 1978, 4–10.
- Dantzig, G.B. and Parikh, S.C. (1977). "At the Interface of Modeling and Algorithms Research" Technical Report SOL 77-29, Department of Operations Research, Stanford University, Stanford, CA, October. Proceedings of Nonlinear Programming Symposium 3, University of Wisconsin, July 1977, Academic Press, 1978, 283–302.
- Dantzig, G.B. and Parikh, S.C. (1978). "PILOT Model for Assessing Energy-Economic Options," in Bagiotti, T. and Franco, G. (eds.), *Pioneering Economics*, Edizioni Cedam - Padova, Italy, 271–276.
- Dantzig, G.B. and Pereira, M.V.F. (1988). "Mathematical Decomposition Techniques for Power System Expansion Planning," EPRI EL-5299, Volumes 1–5, February 1988, Electric Power Research Institute, Palo Alto, CA.

- Dantzig, G.B. and Perold, A.F. (1978). "A Basic Factorization Method for Block Triangular Linear Programs," SOL78-7, April; also in I.S. Duff and G.W. Stewart (eds.), *Sparse Matrix Proceedings*, SIAM, Philadelphia, 1979, 283–312.
- Dantzig, G.B. and Ramser, J.H. (1959a). "Optimum Routing of Gasoline Delivery Trucks," Proceedings, World Petroleum Congress, Session VIII, Paper 19, 1959.
- Dantzig, G.B. and Ramser, J.H. (1959b). "The Truck Dispatching Problem," *Management Science* **6**, 80–91.
- Dantzig, G.B. and Reynolds, G.H. (1966). "Optimal Assignment of Computer Storage by Chain Decomposition of Partially Ordered Sets," Technical Report 66-6, Operations Research Center, University of California, Berkeley.
- Dantzig, G.B. and Saaty, T.L. (1973). *Compact City*, Freeman, San Francisco.
- Dantzig, G.B. and Sethi, S.P. (1981). "Linear Optimal Control Problems and Generalized Linear Programming," *Journal of the Operational Research Society* **32**, 467–476.
- Dantzig, G.B. and Shapiro, M. (1960). "Solving the Chemical Equilibrium Problem Using the Decomposition Principle," P-2056, The RAND Corporation, August.
- Dantzig, G.B., Stone, J.C., and McAllister, P.H. (1986). "Using the PILOT Model to Study the Effects of Technological Change," Technical Report SOL 86-16, Department of Operations Research, Stanford University, Stanford, CA, December; also in B. Lev, J. Bloom, A. Gleit, F. Murphy, and C. Shoemaker (eds.), *Strategic Planning in Energy and Natural Resources*, Studies in Management Science and Systems, Vol. 15, Proceedings of the 2nd Symposium on Analytic Techniques for Energy, Natural Resources and Environmental Planning April 1986; North-Holland, Amsterdam, 1987, 31–42.
- Dantzig, G.B., Stone, J.C., and McAllister, P.H. (1988). "Formulating an Objective for an Economy," Proceedings of the Martin Beale Memorial Symposium, Mathematical Programming 42, (Series B), 11–32.
- Dantzig, G.B. and Tomlin, J.A. (1987). "E.M.L. Beale, FRS: Friend and Colleague," Technical Report SOL 87-2, Department of Operations Research, Stanford University, Stanford, CA, January 1987; also in *Mathematical Programming* **38**, 117–131.
- Dantzig, G.B. and Van Slyke, R.M. (1964a). "Generalized Upper Bounded Techniques for Linear Programming — I," Technical Report 64-17, Operations Research Center, University of California, Berkeley; also in Proceedings IBM Scientific Computing Symposium, Combinatorial Problems, March 16–18, 1964, 249–261.
- Dantzig, G.B. and Van Slyke, R.M. (1964b). "Generalized Upper Bounded Techniques for Linear Programming — II," Technical Report 64-18, Operations Research Center, University of California, Berkeley.
- Dantzig, G.B. and Van Slyke, R.M. (1967). "Generalized Upper Bounding Techniques," *Journal of Computer and System Science* **1**, 213–226.
- Dantzig, G.B. and Van Slyke, R.M. (1971). "Generalized Linear Programming," in David Wismer (ed.), *Optimization Methods and Applications for Large Systems*, McGraw-Hill, New York, 75–120.
- Dantzig, G.B. and Veinott, A.F. (eds.), (1967). *Mathematics of the Decision Sciences*, Proceedings of the American Mathematical Society Summer Seminar, Providence, RI.

- Dantzig, G.B. and Veinott, A.F., editors (1968a). *Lectures in Applied Mathematics*, Vol. 11, American Mathematics Society, Providence, RI.
- Dantzig, G.B. and Veinott, A.F. (1968b). "Integral Extreme Points," Technical Report 67-7, Department of Operations Research, Stanford University, Stanford, CA, November; also in *SIAM Review* **10**, 371–372.
- Dantzig, G.B. and Veinott, A.F. (1977). "Discovering Hidden Totally Leontief Substitution Systems," Technical Report SOL 77-17, Department of Operations Research, Stanford University, Stanford, CA, June. Revised for *Mathematics of Operations Research* **3**, May 1978, 102–103.
- Dantzig, G.B. and Wald, A. (1951). "On the Fundamental Lemma of Neyman and Pearson," *Annals of Mathematical Statistics* **22**, 87–93.
- Dantzig, G.B. and Wolfe, P. (1960). "Decomposition Principle for Linear Programs," *Operations Research* **8**, 101–111; also in P. Wolfe (ed.), *RAND Symposium on Mathematical Programming*, March 1959, RAND R-351, page 5; also in G. B. Dantzig and B. C. Eaves (eds.), *Studies in Optimization*, MAA Studies in Mathematics, Vol. 10, Mathematical Association of America, 1974.
- Dantzig, G.B. and Wolfe, P. (1961). "The Decomposition Algorithm for Linear Programming," *Econometrica*, **29**, 767–778.
- Dantzig, G.B. and Wolfe, P. (1962). "Linear Programming in a Markov Chain," *Operations Research* **10**, 702–710; also *RM-2957-PR*, The RAND Corporation, April.
- Dantzig, G.B. and Wood, M.K. (1951). "Programming of Interdependent Activities, I: General Discussion," *Econometrica*, **17**, 193–199; also in T.C. Koopmans (ed.), *Activity Analysis of Production and Allocation*, July–October 1949, Cowles Commission Monograph 13, Proceedings of Linear Programming Conference, June 20–24, 1949, John Wiley and Sons, New York, 15–18.
- Dantzig, G.B. and Ye, Y. (1990). "A Build-Up Interior Method for Linear Programming: Affine Scaling Form," Technical Report SOL 90-4, Department of Operations Research, Stanford University, Stanford, CA, March.
- Davidon, W.C. (1979). "Variable Metric Methods for Optimization," *Atomic Energy Commission (AEC) Research and Development Report ANL-5990*, Argonne National Laboratory, Argonne, Illinois.
- Davis, K.D. and McKeown, P.G. (1981). *Quantitative Models for Management*, Kent Publishing Company, Boston, Massachusetts.
- Davis, P.J. and Rabinowitz, P. (1984). *Methods of Numerical Integration*, Academic Press, London and New York.
- Day, J. and Peterson, B. (1988). "Growth in Gaussian Elimination," *The American Mathematical Monthly* **95**, 489–513.
- de Boor, C.W. (1971). "CADRE: An Algorithm for Numerical Quadrature," in J. Rice (ed.), *Mathematical Software*, Academic Press, London and New York, 417–449.
- Deák, I. (1988). "Multidimensional Integration and Stochastic Programming," in Y. Ermoliev and R.J.B. Wets (eds.), *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin and New York, 187–200.
- Dembo, R.S. and Steihaug, T. (1983). "Truncated Newton Methods for Large Scale Unconstrained Optimization," *Mathematical Programming* **26**, 190–212.

- Demmel, J.W. (1987). "On the Distance to the Nearest Ill-Posed Problem," *Numerische Mathematik* **51**, 251–289.
- Denardo, E.V. and Fox, B.L. (1979). "Shortest-Route Methods: 1. Reaching, Pruning, and Buckets," *Operations Research* **27**, 161–186.
- Dennis, J.E., Jr. (1977). "Nonlinear Least Squares," in D. Jacobs (ed.), *The State of the Art in Numerical Analysis*, Academic Press, London and New York, 269–312.
- Dennis, J.E. and Moré, J.J. (1977). "Quasi-Newton Methods, Motivation and Theory," *SIAM Review* **19**, 46–89.
- Dennis, J.E. and Moré, J.J. (1974). "A Characterization of Superlinear Convergence and its Application to Quasi-Newton Methods," *Mathematics of Computation* **28**, 549–560.
- Dial, R.B. (1969). "Algorithm 360: Shortest Path Forest with Topological Ordering," *Communications of the Association for Computing Machinery* **12**, 632–633.
- Dijkstra, E. (1959). "A Note on Two Problems in Connection with Graphs," *Numerische Mathematik* **1**, 269–271.
- Dikin, I.I. (1967). "Iterative Solution of Problems of Linear and Quadratic Programming," *Doklady Akademii Nauk USSR* **174**, 747–748, translated in *Soviet Mathematics Doklady* **8**, 674–675.
- Dikin, I.I. (1974). "On the Convergence of an Iterative Process," *Upravlyaemye Sistemy* **12**, 54–60.
- Dikin, I.I. (1990). "The Convergence of Dual Variables," Technical Report, Siberian Energy Institute, Irkutsk, Russia.
- Dikin, I.I. (1992). "Determination of an Interior Point of one System of Linear Inequalities," *Kibernetika and System Analysis* **1**, 74–96.
- Dikin, I.I. and Zorkaltsev, V.I. (1980). *Iterative Solution of Mathematical Programming Problems: Algorithms for the Method of Interior Points*, Nauka, Novosibirsk, USSR.
- Dilworth, R.P. (1950). "A Decomposition Theorem for Partially Ordered Sets," *Annals of Mathematics* **51**, 161–166.
- Dinic, E.A. (1970). "Algorithm for Solution of a Problem of Maximum Flow in a Network with Power Estimation," *Soviet Mathematics Doklady* **11**, 1277–1280.
- Dixon, L.C.W. (1972a). "Quasi-Newton Methods Generate Identical Points," *Mathematical Programming* **2** 383–387.
- Dixon, L.C.W. (1972b). "Quasi-Newton Methods Generate Identical Points. II. The Proof of Four New Theorems," *Mathematical Programming* **3** 345–358.
- Dodson, D.S. and Lewis, J.G. (1985). "Proposed Sparse Extensions to the Basic Linear Algebra Subprograms," *SIGNUM Newsletter* **20**, 22–25.
- Doig, A.G. and Belz, M.H. (1956). "Report on Trim Problems for May, 1956." Department of Statistics, University of Melbourne, Australia. The report is addressed to Australia Paper Manufacturers, Melbourne, Australia, July 31, 1956.
- Doig, A.G. and Land, A.H. (1960). "An Automatic Method of Solving Discrete Programming Problems," *Econometrica*, **28**, 497–520.
- Dongarra, J.J., Bunch, J.R., Moler, C.B., and Stewart, G.W. (1979). "LINPACK Users Guide," SIAM, Philadelphia.

- Dongarra, J.J., DuCroz, J., Hammarling, S. and Hanson, R.J. (1985). "A Proposal for an Extended Set of Fortran Basic Linear Algebra Subprograms," *SIGNUM Newsletter* **20**, 2–18.
- Dongarra, J.J., DuCroz, J., Hammarling, S. and Hanson, R.J. (1988a). "An Extended Set of Fortran Basic Linear Algebra Subprograms," *ACM Transactions on Mathematical Software* **14**, 1–17.
- Dongarra, J.J., DuCroz, J., Hammarling, S. and Hanson, R.J. (1988b). "Algorithm 656 An Extended Set of Fortran Basic Linear Algebra Subprograms: Model Implementation and Test Programs," *ACM Transactions on Mathematical Software* **14**, 18–32.
- Dongarra, J.J., DuCroz, J., Hammarling, S. and Hanson, R.J. (1988c). "A Set of Level 3 Basic Linear Algebra Subprograms," Report ANL-MCS-TM-88, Argonne National Laboratory, Argonne, Illinois.
- Dongarra, J.J., Duff, I.S., Sorensen, D.C., and van der Vorst, H.A. (1991). *Solving Linear Systems on Vector and Shared Memory Computers*, SIAM, Philadelphia.
- Dongarra, J.J. and Eisenstat, S. (1984). "Squeezing the Most out of an Algorithm in Cray Fortran," *ACM Transactions on Mathematical Software* **10**, 221–230.
- Dongarra, J.J., Gustavson, F.G., and Karp, A. (1984). "Implementing Linear Algebra Algorithms for Dense Vectors on a Vector Pipeline Machine," *SIAM Review* **26**, 91–112.
- Dongarra, J.J. and Hinds, A. (1979). "Unrolling Loops in Fortran," *Software Practice and Experience* **9**, 219–229.
- Duff, I.S. (1976). "A Survey of Sparse Matrix Research," *Report AERE CSS 28*, Atomic Energy Research Establishment, Harwell, England.
- Duff, I.S. (1977). "MA28—A Set of Fortran Subroutines for Sparse Unsymmetric Linear Equations," *Report AERE AERE R8730*, Atomic Energy Research Establishment, Harwell, England.
- Duff, I.S. (1981). "An Algorithm for Obtaining a Maximum Transversal," *ACM Transactions on Mathematical Software* **7**, 315–330.
- Duff, I.S., Erisman, A.M., and Reid, J.K. (1986). *Direct Methods for Sparse Matrices*, Oxford University Press, Oxford and New York.
- Duff, I.S. and Reid, J.K. (1978). "An Implementation of Tarjan's Algorithm for the Block Triangularization of a Matrix," *ACM Transactions on Mathematical Software* **4**, 137–147.
- Duff, I.S. and Stewart, G.W. (eds.) (1979). *Sparse Matrix: Proceedings*, SIAM, Philadelphia.
- Duffin, R.J. (1974). "On Fourier's Analysis of Linear Inequality Systems," *Mathematical Programming Study* **1**, 71–95.
- Dupačová, J. (1990). "Stability and Sensitivity Analysis for Stochastic Programming," *Annals of Operations Research* **27**, 115–142.

E

- Eaves, B.C. (1979). "A View of Complementary Pivot Theory (or Solving Equations with Homotopies)," in C.V. Coffman and G.J. Fix (eds.), *Constructive Approaches to Mathematical Models*, 1979, Academic Press, London and New York, 153–170; and in H-O. Peitgen and H-O. Walther (eds.), *Functional Differential Equations and Approximation of Fixed Points*, 1979, Springer-Verlag, Berlin and New York
- Edmonds, J. and Karp, R.M. (1972). "Theoretical Improvements in Algorithmic Efficiency for Network Flow Algorithms," *Journal of the Association for Computing Machinery* **19**, 248–264.
- Edmondson, J.H. (1951). "Proof of the Existence Theorem of an Epsilon Transformation," class exercise dated March 28, 1951, for Department of Agriculture Graduate School course in linear programming given by George Dantzig.
- Egerváry, E. (1931). "Matrixok Kombinatorius Tulajfonságairól," *Matematikai és Fizikai Lapok*, No. 38, 16–28. "On Combinatorial Properties of Matrices;," translated by H.W. Kuhn, Paper No. 4, George Washington University Logistics Research Project. Published in *Logistics Papers*, Issue No. 11, Appendix I to Quarterly Progress Report No. 21, (November 16, 1954 to February 15, 1955).
- Elias, P., Feinstein, A., and Shannon, C.E. (1956). "Note on Maximum Flow through a Network," *IRE Transactions on Information Theory*, **IT-2**, 117–119.
- Elmaghraby, S.E. (1977). *Activity Networks: Project Planning and Control by Network Models*, Wiley, New York.
- Entriken, R. (1989). "The Parallel Decomposition of Linear Programs," Ph.D. thesis, Department of Operations Research, Stanford University, Stanford, CA.
- Erisman, A.M., Grimes, R.G., Lewis, J.G., and Poole, W.G., Jr. (1985). "A Structurally Stable Modification of Hellerman-Rarick's P^4 Algorithm for Reordering Unsymmetric Sparse Matrices," *SIAM Journal on Numerical Analysis* **22**, 369–385.
- Erisman, A.M. and Reid, J.K. (1974). "Monitoring the Stability of the Triangular Factorization of a Sparse Matrix," *Numerische Mathematik* **22**, 183–186.
- Ermoliev, Y. (1983). "Stochastic Quasi-Newton Methods and Their Applications to Systems Optimization," *Stochastics*, **9**, 1–36.
- Ermoliev, Y. (1988). "Stochastic Quasi-Gradient Methods," in Y. Ermoliev and R.J.B. Wets (eds.), *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin and New York, 141–186.
- Ermoliev, Y. and Wets, R.J.B., eds. (1988). *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin and New York.
- Eisemann, K. (1957). "The Trim Problem," *Management Science* **3**, 279–284.

F

- Fabian, T. (1954). "Process Type Analysis of the Iron and Steel Industry, Part IV—Programming Open Hearth Steel Production," Discussion Paper No. 46, Management Sciences Research Project, University of California, Los Angeles, California, 12 pages.

- Fabian, T. (1955). "Process Analysis of the Iron and Steel Industry: A Model," *Econometrica*, **23**, 347–348 (abstract); also Research Report No. 47, Management Sciences Research Project, University of California, Los Angeles, California.
- Fabian, T. (1958). "A Linear Programming Model of Integrated Iron and Steel Production," *Management Science* **4**, 425–449.
- Faddeev, D.K., and Faddeeva, V.N. (1963). *Computational Methods in Linear Algebra*, Freeman, San Francisco; translated from the Russian version.
- Farkas, J. (1902). "Über die Theorie der einfachen Ungleichungen," *Journal für die Reine and Angewandte Mathematik* **124**, 1–24.
- Faulkner, J.C. (1988). "Bus crew scheduling and the set partitioning model," Ph.D. thesis, Department of Theoretical and Applied Mechanics, University of Auckland, Auckland, New Zealand.
- Feinstein, C.D. and Thapa, M.N. (1993). "A Reformulation of a Mean-Absolute Deviation Portfolio Optimization Model," *Management Science* **39**, 1552–1553.
- Feller, W. (1957). *An Introduction to Probability Theory and Its Applications*, Volume I, John Wiley and Sons, New York.
- Feller, W. (1969). *An Introduction to Probability Theory and Its Applications*, Volume II, John Wiley and Sons, New York.
- Fenner, T. and Loizou, G. (1974). "Some New Bounds on the Condition Numbers of Optimally Scaled Matrices," *Journal of the Association for Computing Machinery* **21**, 514–524.
- Ferguson, A.R. and Dantzig, G.B. (1955). "Notes on Linear Programming: Part XVI—The Problem of Routing Aircraft—a Mathematical Solution," *Aeronautical Engineering Review* **14**, 51–55; also in *RM-1369*, The RAND Corporation, September 1, 1954, and *P-561*, The RAND Corporation, 1954.
- Ferguson, A.R. and Dantzig, G.B. (1956). The Allocation of Aircraft to Routes—An Example of Linear Programming under Uncertain Demand," *Management Science* **3** 1, 45–73; also in *P-727*, The RAND Corporation, December 7, 1956; also in Bowman and Fetter (eds.), *Analysis of Industrial Operations*, Richard D. Irwin, Inc., Homewood, Illinois, 1959.
- Fisher, W.D. and Schruben, L.W. (1953). "Linear Programming Applied to Feed-Mixing under Different Price Conditions," *Journal of Farm Economics* **35**, 471–483.
- Fiacco, A.V. and McCormick, G.P. (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York.
- Fletcher, R. (1985). "Degeneracy in the Presence of Round-Off Errors," Technical Report NA/89, Department of Mathematical Sciences, University of Dundee, Dundee.
- Fletcher, R. (1987). "Recent Developments in Linear and Quadratic Programming", in A. Iserles and M.J.D. Powell (eds.), *The State and Art in Numerical Analysis*, Oxford University Press, Oxford and New York, 213–243.
- Fletcher, R., and Mathews, S.P.J. (1984). "Stable Modification of Explicit LU Factors for Simplex Updates," *Mathematical Programming* **30**, 267–284.
- Ford, L.R., Jr., and Fulkerson, D.R. (1956). "Maximal Flow Through a Network," *Canadian Journal of Mathematics* **8**, 399–404. This appeared first as the RAND Corporation Research Memorandum RM-1400, November 19, 1954.

- Ford, L.R., Jr., and Fulkerson, D.R. (1957). "A Simple Algorithm for Finding Maximal Network Flows and an Application to the Hitchcock Problem," *Canadian Journal of Mathematics* **9**, 210–218.
- Ford, L.R., Jr., and Fulkerson, D.R. (1958a). "Constructing Maximal Dynamic Flows from Static Flows," *Operations Research* **6**, 419–433.
- Ford, L.R., Jr., and Fulkerson, D.R. (1958b). "Suggested Computation for Maximal Multi-Commodity Network Flows," *Management Science* **5**, 97–101.
- Ford, L.R., Jr., and Fulkerson, D.R. (1962). *Flows in Networks*, Princeton University Press, Princeton, New Jersey.
- Forrest, J.J.H. and Goldfarb, D. (1992). "Steepest-edge Simplex Algorithms for Linear Programming," *Mathematical Programming* **57**, 341–376.
- Forrest, J.J.H. and Tomlin, J. A. (1972). "Updating Triangular Factors of the Basis to Maintain Sparsity in the Product Form Simplex Method," *Mathematical Programming* **2**, 263–278.
- Forsgren, A.L. and Murray, W. (1990). "Newton Methods for Large-Scale Linearly Constrained Minimization," Technical Report SOL 90-6, Department of Operations Research, Stanford University, Stanford, CA.
- Forsythe, G.E. (1960). "Crout with Pivoting," *Communications of the Association for Computing Machinery* **3**, 507–508.
- Forsythe, G.E. (1970). "Pitfalls in Computation, or Why a Math Book Isn't Enough," *The American Mathematical Monthly* **9**, 931–956.
- Forsythe, G.E., Malcolm, M.A., and Moler, C.B. (1977). *Computer Methods for Mathematical Computations*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Forsythe, G.E. and Moler, C.B. (1967). *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Forsythe, G.E. and Wastow, W.R. (1960). *Finite Difference Methods for Partial Differential Equations*, John Wiley and Sons, New York.
- Foster, L.V. (1986). "Rank and Null Space Calculations Using Matrix Decomposition Without Column Interchanges," *Linear Algebra and Its Applications* **74**, 47–71.
- Fourer, R. (1979). "Sparse Gaussian Elimination of Staircase Linear Systems," Technical Report SOL 79-17, Department of Operations Research, Stanford University, Stanford, CA.
- Fourer, R. (1982). "Solving Staircase Linear Programs by the Simplex Method, 1: Inversion," *Mathematical Programming* **23**, 274–313.
- Fourer, R. (1983a). "Solving Staircase Linear Programs by the Simplex Method, 2: Pricing," *Mathematical Programming* **25**, 251–292.
- Fourer, R. (1983b). "Modeling Languages versus Matrix Generators for Linear Programming," *ACM Transactions on Mathematical Software* **9**, 143–183.
- Fourer, R. (1984). "Staircase Systems and Matrices," *SIAM Review* **26**, 1–70.
- Fourer, R. (1985). "A Simplex Algorithm for Piecewise-Linear Programming I: Derivation and Proof," *Mathematical Programming* **33** 204–233.
- Fourer, R., Gay, D.M., and Kernighan, B.W. (1992). *AMPL: A Modeling Language for Mathematical Programming*, Scientific Press, South San Francisco, California.

- Fourer, R. and Mehrotra, S. (1992). "Performance of an Augmented System Approach for Solving Least-Squares Problems in an Interior-Point Method for Linear Programming," *Mathematical Programming Society COAL Newsletter* **19**, 26–31; previously published in 1991 as a technical report, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL.
- Fourer, R. and Mehrotra, S. (1993). "Solving Symmetric Indefinite Systems in an Interior-Point Method for Linear Programming," *Mathematical Programming* **62**, 15–39; previously published in 1992 as Technical Report 92-01, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL.
- Fourier, J.B.J. (1826). "Solution d'une question particulière du calcul des inégalités," 1826, and extracts from "Histoire de l'Académie," 1823, 1824, *Oeuvres* II, 317–328.
- Fourier, J.B.J. (1890). "Second Extrait," in G. Darboux (ed.), *Oeuvres*, Gauthiers-Villars, Paris, 325–328; English translation by Kohler [1973].
- Fox, L. (1973). *Introduction to Numerical Linear Algebra*, Oxford University Press, Oxford and New York; originally published in 1964.
- Fraley, C. and Vial, J.-Ph. (1989). "Numerical Study of Projective Methods for Linear Programming," in S. Dolecki (ed.), *Optimization Proceedings of the 5th French-German Conference in Castel-Novel, Varetz, France, October 1988*, 25–38.
- Frauentorfer, K. (1988). "Solving SLP Recourse Problems with Arbitrary Multivariate Distributions — The Dependent Case," *Mathematics of Operations Research* **13**, 3, 377–394.
- Frauentorfer, K. (1992). *Stochastic Two-Stage Programming*, Lecture Notes in Economics and Mathematical Systems 392, Springer-Verlag, Berlin and New York.
- Frauentorfer, K., and Kall, P. (1988). "Solving SLP Recourse Problems with Arbitrary Multivariate Distributions — The Independent Case," *Problems of Control and Information Theory* **17** (4), 177–205.
- Fredman, M.L. and Willard, D.E. (1994). "Trans-dichotomous Algorithms for Minimum Spanning Trees and Shortest Paths," *Journal of Computer and System Sciences* **48**, 533–551.
- Freund, R. (1988a). "Projective Transformation for Interior Point Methods, Part I: Basic Theory and Linear Programming," working paper OR 179-88, Operations Research Center, Massachusetts Institute of Technology.
- Freund, R. (1988b). "An Analog of Karmarkar's Algorithm for Inequality Constrained Linear Programs, with a 'New' Class of Projective Transformations for Centering a Polytope," *Operations Research Letters* **7**, 9–14; previously published in 1987 as a working paper OR 1921-87, Operations Research Center, Massachusetts Institute of Technology.
- Freund, R. (1991a). "Theoretical Efficiency of a Shifted Barrier Function Algorithm for Linear Programming," *Linear Algebra and Its Applications* **152**, 19–41.
- Freund, R. (1991b). "Polynomial-Time Algorithms for Linear Programming Based only on Primal Scaling and Projected Gradients of a Potential Function," *Mathematical Programming* **51**, 203–222.
- Frisch, K.R. (1955). *The Logarithmic Potential Method of Convex Programs*, Memorandum of May 13, 1955, UIEOSLO.

Frisch, K.R. (1957). *Linear Dependencies and a Mechanized Form of the Multiplex Method Linear Programming*, Memorandum of September, 1957, University Institute of Economics, Oslo, Norway.

Fulkerson, D.R., and Dantzig, G.B. (1955). "Computations of Maximal Flows in Networks," *Naval Research Logistics Quarterly*, **2**, 277–283.

G

Gabow, H.N. and Tarjan, R.E. (1989). "Faster Scaling Algorithms for Network Problems," *SIAM Journal on Computing* **18**, 1013–1036.

Gainen, L. (1955). "Linear Programming in Bid Evaluations," in H.A. Antosiewicz (ed.), *Proceedings of the Second Symposium in Linear Programming*, Vol. 2, National Bureau of Standards and Directorate of Management Analysis, DCS/Comptroller, USAF, Washington, DC, 29–38.

Gaivoronski, A. (1988). "Implementation of Stochastic Quasi-Gradient Methods," in Y. Ermoliev and R.J.B. Wets (eds.), *Numerical Techniques for Stochastic Optimization*, Springer-Verlag, Berlin and New York, 313–352.

Gale, D.H. (1951). "Convex Polyhedral Cones and Linear Inequalities," in T.C. Koopmans (ed.), *Activity Analysis of Production and Allocation*, July–October 1949, Cowles Commission Monograph 13, Proceedings of Linear Programming Conference, June 20–24, 1949, John Wiley and Sons, New York, 287–297.

Gale, D.H., Kuhn, H.W., and Tucker, A.W. (1951). "Linear Programming and the Theory of Games," in T.C. Koopmans (ed.), *Activity Analysis of Production and Allocation*, July–October 1949, Cowles Commission Monograph 13, Proceedings of Linear Programming Conference, June 20–24, 1949, John Wiley and Sons, New York, 317–329.

Galil, Z. (1978). "A New Algorithm for the Maximal Flow Problem," *Proceedings of the Nineteenth Annual Symposium on Foundations of Computer Science, IEEE*, 231–245.

Gallivan, K., Jalby, W., Meier, U., and Sameh, A.H. (1988). "Impact of Hierarchical Memory Systems on Linear Algebra Algorithm Design," *International Journal of Supercomputer Applications* **2**, 12–48.

Gallo, G. and Pallottino, S. (1988). "Shortest Path Algorithms," *Annals of Operations Research* **13**, 3–79.

Garbow, B.S., Boyle, J.M., Dongarra, J.J., and Moler, C.B. (1977). *Matrix Eigensystem routines—EISPACK Guide Extensions*, *Lecture Notes in Computer Science* **51**, Springer-Verlag, Berlin and New York.

Garvin, W.W., Crandall H.W., John, J.B., and Spellman, R.A. (1957). "Applications of Linear Programming in the Oil Industry," *Management Science* **3**, 407–430.

Gass, S.I. (1991). "Model World: Models at the OK Corral," *Interfaces* **21**:6, 80–86.

Gass, S.I. (1985). *Linear Programming: Methods and Applications*, McGraw-Hill, New York.

Gass, S.I., and Saaty, T.L. (1955a). "The Computational Algorithm for the Parametric Objective Function," *Naval Research Logistics Quarterly*, **2**, 39–45.

- Gass, S.I., and Saaty, T.L. (1955b). "The Parametric Objective Function, Part 1," *Operations Research* **2**, 316–319.
- Gass, S.I., and Saaty, T.L. (1955c). "The Parametric Objective Function, (Part 2)—Generalization," *Operations Research* **3**, 395–401.
- Gassner, B.J. (1964). "Cycling in the Transportation Problem," *Naval Research Logistics Quarterly*, **11**, 43–58.
- Gay, D.M. (1978). "On Combining the Schemes of Reid and Saunders for Sparse LP Bases," in I.S. Duff and G.W. Stewart (eds.), *Sparse Matrix Proceedings*, SIAM, Philadelphia, 313–334.
- Gay, D.M. (1985). "Electronic Mail Distribution of Linear Programming Test Problems," *Mathematical Programming Society COAL Newsletter* **13**, 10–12.
- Gay, D.M. (1987). "A Variant of Karmarkar's Linear Programming Algorithm for Problems in Standard Form," *Mathematical Programming* **37**, 81–90.
- Gentleman, M. (1973). "Least Squares Computations by Givens Transformations Without Square Roots," *Journal of the Institute of Mathematics and Its Applications* **12**, 329–336.
- Geoffrion, A.M. (1970). "Elements of Large-Scale Mathematical Programming," *Management Science* **16**, 652–691.
- George, A. and Liu, J. (1981). *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- George, A. and Ng, E. (1984). "Symbolic Factorization for Sparse Gaussian Elimination with Partial Pivoting," Technical Report CS-84-43, Department of Computer Science, University of Waterloo, Waterloo, Ontario, Canada.
- de Ghellinck, G. and Vial, J.-Ph. (1986). "A Polynomial Newton Method for Linear Programming," special issue of *Algorithmica* **1**, 425–453.
- Gill, P.E., Golub, G.H., Murray, W. and Saunders, M.A. (1974). "Methods for Modifying Matrix Factorizations," *Mathematics of Computation* **28**, 505–535.
- Gill, P. E., and Murray, W. (1974a). *Safeguarded Steplength Algorithms for Optimization Using Descent Methods*, Report NAC 37, National Physical Laboratory, England.
- Gill, P.E. and Murray, W. (1974b). "Newton Type Methods for Unconstrained and Linearly Constrained Optimization," *Mathematical Programming* **7**, 311–350.
- Gill, P.E. and Murray, W. (1974c). "Quasi-Newton Methods for Linearly Constrained Optimization," in P.E. Gill and W. Murray (eds.), *Numerical Methods for Unconstrained Optimization*, Academic Press, London and New York, 67–92.
- Gill, P.E. and Murray, W. (1974d). *Numerical Methods for Constrained Optimization*, Academic Press, London and New York.
- Gill, P.E. and Murray, W. (1977). "Linearly Constrained Problems Including Linear and Quadratic Programming," in D. Jacobs (ed.), *The State of the Art in Numerical Analysis*, Academic Press, London and New York, 313–363.
- Gill, P.E. and Murray, W. (1978). "Algorithms for the Solution of the Nonlinear Least Squares Problem," *SIAM Journal on Numerical Analysis* **15**, 977–992.

- Gill, P.E. and Murray, W. (1979). *Conjugate-Gradient Methods for Large Scale Nonlinear Optimization*, Technical Report SOL 79-15, Department of Operations Research, Stanford University, Stanford, CA.
- Gill, P.E., Murray, W. and Picken, S.M. (1972). "The Implementation of Two Modified Newton Algorithms for Unconstrained Optimization," Report NAC 11, National Physical Laboratory, England.
- Gill, P.E., Murray, W., Picken, S.M., and Wright, M.H. (1979). "The Design and Structure of a FORTRAN Program Library for Optimization," *ACM Transactions on Mathematical Software* **5**, 259–283.
- Gill, P.E., Murray, W., Ponceleón, D.B., and Saunders, M.A. (1994). "Solving reduced KKT systems in barrier methods for linear programming," in J.H. Watson and D. Griffiths (eds.), *Numerical Analysis 1993*, Pitman Research Notes in Mathematics 303, Longmans Press, 89–104; originally published in 1991 as "Solving Reduced KKT Systems in Barrier Methods for Linear and Quadratic Programming," Technical Report SOL 91-7, Department of Operations Research, Stanford University, Stanford, CA.
- Gill, P.E., Murray, W., Ponceleón, D.B., and Saunders, M.A. (1995). "Primal-dual methods for linear programming," *Mathematical Programming* **70**, 251–277; originally published in 1991, Technical Report SOL 91-3, Department of Operations Research, Stanford University, Stanford, CA.
- Gill, P.E., Murray, W., and Saunders, M.A. (1988). "Interior-Point Methods for Linear Programming: A Challenge to the Simplex Method," Technical Report SOL 88-14, Department of Operations Research, Stanford University, Stanford, CA.
- Gill, P.E., Murray, W., Saunders, M.A., Tomlin, J.A. and Wright, M.H. (1986). "On Projected Newton Barrier Methods for Linear Programming and an Equivalence to Karmarkar's Projective Method," *Mathematical Programming* **36**, 183–209.
- Gill, P.E., Murray, W., Saunders, M.A. and Wright, M.H. (1981a). "A Procedure for Computing Forward Finite-Difference Intervals for Numerical Optimization," Technical Report SOL 81-25, Department of Operations Research, Stanford University, Stanford, CA.
- Gill, P.E., Murray, W., Saunders, M.A., and Wright, M.H. (1981b). "QP-based Methods for Large Scale Nonlinearly Constrained Optimization," in O.L. Mangasarian, R.R. Meyer, and S.M. Robinson (eds.), *Nonlinear Programming 4*, Academic Press, London and New York, 57–98.
- Gill, P.E., Murray, W. Saunders, M.A. and Wright, M.H. (1983). "On the Representation of a Basis for the Null Space," Technical Report SOL 83-19, Department of Operations Research, Stanford University, Stanford, CA.
- Gill, P.E., Murray, W., Saunders, M.A. and Wright, M.H. (1984a). "Sparse Matrix Methods in Optimizations," *SIAM Journal on Scientific and Statistical Computing* **5**, 562–589.
- Gill, P.E., Murray, W., Saunders, M.A., and Wright, M.H. (1984b). "Software and its Relationship to Methods," Technical Report SOL 84-10, Department of Operations Research, Stanford University, Stanford, CA.

- Gill, P.E., Murray, W., Saunders, M.A., and Wright, M.H. (1984c). "Model Building and Practical Implementation Aspects in Nonlinear Programming," Presented at the NATO Advanced Study Institute on *Computational Mathematical Programming* Bad Windsheim, July 23–August 2, 1984.
- Gill, P.E., Murray, W., Saunders, M.A. and Wright, M.H. (1985). "LUSOL User's Guide," Working Technical Report, Department of Operations Research, Stanford University, Stanford, CA.
- Gill, P.E., Murray, W., Saunders, M.A. and Wright, M.H. (1987). "Maintaining LU Factors of a General Sparse Matrix," *Linear Algebra and Its Applications* **88/99** 239–270.
- Gill, P.E., Murray, W., Saunders, M.A. and Wright, M.H. (1989). "A Practical Anti-Cycling Procedure for Linearly Constrained Optimization," *Mathematical Programming* **45**, 437–474.
- Gill, P.E., Murray, W., and Wright, M.H. (1981). *Practical Optimization*, Academic Press, London and New York.
- Gill, P.E., Murray, W., and Wright, M.H. (1991). *Numerical Linear Algebra and Optimization*, Addison-Wesley Publishing Company, Reading, Massachusetts.
- Gille, P. and Loute, E. (1982). "Updating the LU Gaussian Decomposition for Rank-One Corrections; Application to Linear Programming Basis Partitioning Techniques," *Cahier No. 8201, Séminaire de Mathématiques Appliquées aux Sciences Humaines, Facultés Universitaires Saint-Louis, Brussels, Belgium*.
- Givens, W. (1954). "Numerical Computation of the Characteristic Values of a Real Symmetric Matrix," *Technical Report ORNL-1574*, Oak Ridge National Laboratory, Oak Ridge, Tennessee.
- Glasse, C.R. (1971). "Dynamic LP's for Production Scheduling," *Operations Research* **19**, 45–56.
- Glover, F., Karney, D., and Klingman, D. (1972). "The Augmented Predecessor Index Method for Locating Stepping Stone Paths and Assigning Dual Prices in Distribution Problems," *Transportation Science* **6**, 171–180.
- Glover, F., Karney, D., and Klingman, D. (1973). "A Note on Computational Studies for Solving Transportation Problems," *Proceedings of the ACM Annual Conference*, Atlanta, Georgia, 7–11.
- Glover, F., Karney, D., and Klingman, D. (1974). "Implementation and Computational Comparisons of Primal, Dual, and Primal-Dual Computer Codes for Minimum Cost Flow Network Problems," *Networks*, **4**, 191–212.
- Glover, F., Karney, D., Klingman, D., and Napier, A. (1974). "A Computational Study on Start Procedures, Basis Change Criteria and Solution Algorithms for Transportation Problems," *Management Science* **20**, 793–813.
- Glynn, P.W. and Iglehart, D.L. (1989). "Importance Sampling for Stochastic Simulation," *Management Science* **35**, 1367–1392.
- Goffin, J.L. and Vial, J.-Ph. (1990). "Cutting Plane and Column Generation Techniques with the Projective Algorithm," *Journal of Optimization Theory and Applications* **65**, 409–429.
- Goldberg, A.V. (1993). "Scaling Algorithms for the Shortest Path Problems," in *Proceedings 4th ACM-SIAM Symposium on Discrete Algorithms*, 222–231.

- Goldberg, A.V. and Radzik, T. (1993). "A Heuristic Improvement of the Bellman-Ford Algorithm," *Applied Mathematics Letters* **6**, 3–6.
- Goldfarb, D. and Reid, J.K. (1977). "A Practical Steepest-Edge Simplex Algorithm," *Mathematical Programming* **12**, 361–371.
- Goldman, A.J. (1956). "Resolution and Separation Theorems for Polyhedral Convex Sets," in H.W. Kuhn and A.W. Tucker (eds.), *Linear Inequalities and Related Systems*, Annals of Mathematics Study No. 38, Princeton University Press, Princeton, New Jersey, 41–51.
- Goldman, A.J. and Tucker, A.W. (1956a). "Polyhedral Convex Cones," in H.W. Kuhn and A.W. Tucker (eds.), *Linear Inequalities and Related Systems*, Annals of Mathematics Study No. 38, Princeton University Press, Princeton, New Jersey, 19–39.
- Goldman, A.J. and Tucker, A.W. (1956b). "Theory of Linear Programming," in H.W. Kuhn and A.W. Tucker (eds.), *Linear Inequalities and Related Systems*, Annals of Mathematics Study No. 38, Princeton University Press, Princeton, New Jersey, 53–97.
- Goldstein, L. (1952). "Problem of Contract Awards," in A. Orden and L. Goldstein (eds.), *Symposium on Linear Inequalities and Programming*, Project SCOOP No. 10, Planning Research Division, Director of Management Analysis Service, Comptroller, USAF, Washington, DC, April, 147–154.
- Golub, G.H. and Van Loan, C.F. (1989). *Matrix Computations*, John Hopkins University Press, Baltimore.
- Gonzaga, C.C. (1989). "Conical Projection Algorithms for Linear Programming," *Mathematical Programming* **43**, 151–173.
- Gonzaga, C.C. (1991). "Large Steps Path-Following Methods for Linear Programming, Part II: Potential Reduction Method," *SIAM Journal on Optimization* **1**, 280–292.
- Gonzaga, C.C. (1992). "Path Following Methods for Linear Programming," *SIAM Review* **34**, 167–227.
- Goodman, S.E. and Hedetniemi, S.T. (1985). *Introduction to the Design and Analysis of Algorithms*, McGraw-Hill, San Francisco, California.
- Gordan, P. (1873). "Über die Auflösung linearer Gleichungen mit reellen Koeffizienten," *Mathematical Annals* **6**, 23–28.
- Gould, N. (1991). News Clip in *SiamNews*, January, 9.
- Grandzol, J.R. and Traaen, T. (1995). "Using Mathematical Programming to Help Supervisors Balance Workloads," *Interfaces* **25**:4, 92–103.
- Graves, G.W. (1965). "A Complete Constructive Algorithm for the General Mixed Linear Programming Problem," *Naval Research Logistics Quarterly*, **12**, 1–34.
- Grear, J.F. (1990). "Matrix Stretching for Linear Equations," Report SAND90-8723, Sandia National Laboratories, Albuquerque, NM.
- Greenberg, H.J. (ed.), (1978a). *Design and Implementation of Optimization Software*, Sijthoff and Noordhoff, Alphen aan den Rijn.
- Greenberg, H.J. (1978b). "A Tutorial on Matricial Packing," in H.J. Greenberg (ed.), *Design and Implementation of Optimization Software*, Sijthoff and Noordhoff, Alphen aan den Rijn, 109–142.

- Greenberg, H.J. (1978c). "Pivot Selection Tactics," in H.J. Greenberg (ed.), *Design and Implementation of Optimization Software*, Sijthoff and Noordhoff, Alphen aan den Rijn, 143–178.
- Greenberg, H.J. and Kalan, J. (1975). "An Exact Update for Harris' TREAD," *Mathematical Programming Study* **4**, 26–29.
- Greenstadt, J.L. (1967). "On the Relative Efficiencies of Gradient Methods," *Mathematics of Computation* **24**, 145–166.
- Grimes, R.G. and Lewis, J.G. (1981). "Condition Number Estimation for Sparse Matrices," *SIAM Journal on Scientific and Statistical Computing* **2**, 384–388.
- Grötschel, M., Lovász, L., and Schrijver, A. (1988). *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin and New York.
- Grunbaum, B. (1967). *Convex Polytopes*, John Wiley and Sons, New York.
- Gunderman, R.E. (1973). "A Glimpse into Program Maintenance," *Datamation* **19** 99–101.

H

- Hadley, G. (1972). *Linear Programming*, Addison-Wesley Publishing Company, Reading, Massachusetts.
- Hall, P. (1985). "Rates of Convergence in the Central-Limit Theorem," *Bulletin of the London Mathematical Society* **17**, 151–156; previously published in 1982, *Research Notes in Mathematics*, **62**.
- Hall, L.A. and Vanderbei, R.J. (1993). "Two-Thirds is Sharp for Affine Scaling," *Operations Research Letters* **13**, 197–201.
- Hammersly, J.M. and Handscomb, D.C. (1964). *Monte Carlo Methods*, Matheun, London.
- Harris, P.M.J. (1975). "Pivot Selection Codes in the Devex LP Code," *Mathematical Programming Study* **4**, 30–57.
- Hellerman, E. and Rarick, D. (1971). "Reinversion in the Preassigned Pivot Procedure," *Mathematical Programming* **1**, 195–216.
- Hellerman, E. and Rarick, D. (1972). "The Partitioned Preassigned Pivot Procedure," in D.J. Rose and R.A. Willoughby (eds.), *Sparse Matrices and their Applications*, Plenum Press, New York, 67–76.
- Henderson, A. and Schlaifer, R. (1954). "Mathematical Programming," *Harvard Business Review*, **32**, May-June, 73–100.
- Hertog, D.D. and Roos, C. (1991). "A Survey of Search Directions in Interior Point Methods for Linear Programming," *Mathematical Programming* **52**, 481–509.
- Hestenes, M.R. and Stiefel, E. (1952). "Methods of Conjugate Gradients for Solving Linear Systems," *Journal of Research of the National Bureau of Standards* **49**, 409–436.
- Higham, N.J. (1985). "Nearness Problems in Numerical Algebra," Ph.D. thesis, University of Manchester, Manchester, England.
- Higham, N.J. (1987). "A Survey of Condition Number Estimation for Triangular Matrices," *SIAM Review* **29**, 575–596.

- Higham, N.J. and Higham, D.J. (1989). "Large Growth Factors in Gaussian Elimination with Pivoting," *SIAM Journal on Matrix Analysis and Applications* **10**, 155–164.
- Higle, J.L. and Sen, S. (1991). "Stochastic Decomposition: An Algorithm for Two Stage Linear Programs with Recourse," *Mathematics of Operations Research* **16/3**, 650–669.
- Hillier, F.S., and Lieberman, G.J. (1995). *Introduction to Operations Research*, McGraw-Hill, San Francisco, California.
- Hitchcock, F.L. (1941). "The Distribution of a Product from Several Sources to Numerous Localities," *Journal of Mathematical Physics* **20**, 224–230.
- Hirsch, W.M. (1957). "Hirsch Conjecture," verbal communication to Dantzig.
- Ho, J.K. (1974). "Nested Decomposition for Large-Scale Programs with the Staircase Structure," Technical Report SOL 74-4, Department of Operations Research, Stanford University, Stanford, CA.
- Ho, J.K. (1984). "Convergence Behaviour of Decomposition Algorithms for Linear Programs," *Operations Research Letters* **3**, 91–94.
- Ho, J.K. and Loute, E. (1981). "An Advanced Implementation of the Dantzig-Wolfe Decomposition Algorithm for Linear Programming," *Mathematical Programming* **20**, 303–326.
- Ho, J.K. and Manne, A. (1974). "Nested Decomposition for Dynamic Models," *Mathematical Programming* **6**, 121–140.
- Hoffman, A.J. (1953). "Cycling in the Simplex Algorithm," *National Bureau of Standards Report No. 2974*.
- Hoffman, A.J. and Hirsch, W. (1961). "Extreme Varieties, Concave Functions and the Fixed Charge Problem," *Communications in Pure and Applied Mathematics* **14**, 355–369.
- Hoffman, A.J., Mannos, M., Sokolowsky, D., and Wiegmann, N. (1953). "Computational Experience in Solving Linear Programs," *Journal of the Society for Industrial and Applied Mathematics*, **1**, 17–33.
- Hockney, R.W. and Jesshope, C.R. (1988). *Parallel Computers 2*, Adam Hilger, Bristol and Philadelphia.
- Holmes, D. (1994). "A Collection of Stochastic Programming Problems," Technical Report 94-11, Department of Industrial Engineering and Operations Research, University of Michigan, Ann Arbor, Michigan.
- Hooker, J.N. (1986). "Karmarkar's Linear Programming Algorithm," *Interfaces* **16:4**, 75–90.
- Householder, A.S. (1964). *The Theory of Matrices in Numerical Analysis*, Dover Publications, New York.
- Huang, C.C., Ziemba, W.T., and Ben-Tal, A. (1977). "Bounds on the Expectation of a Convex Function with a Random Variable with Applications to Stochastic Programming," *Operations Research* **25**, 315–325.
- Huard, P. (1970). "A Method of Centers by Upper-Bounding Functions with Applications," in J.B. Rosen, O.L. Mangasarian, and K. Ritter (eds.), *Nonlinear Programming: Proceedings of a Symposium held at the University of Wisconsin, Madison, May 1970*, Academic Press, London and New York, 1–30.

I

- Ignizio, J.P. (1976). *Goal Programming and Extensions*, Heath, Lexington, Massachusetts.
- Infanger, G. (1991). "Monte Carlo (Importance) Sampling within a Benders Decomposition Algorithm for Stochastic Linear Programs Extended Version: Including Results of Large-Scale Problems," Technical Report SOL 91-6, Department of Operations Research, Stanford University, Stanford, CA; shorter version in *Annals of Operations Research* **39**, 69–95.
- Infanger, G. (1994). *Planning Under Uncertainty: Solving Large-Scale Stochastic Linear Programs*, Boyd and Fraser Publishing Company, Massachusetts.
- International Business Machines Corporation (1978). *Mathematical Programming System Extended and Generalized Upper Bounding (GUB)*, IBM Manual SH20-0968-1, White Plains, New York.
- Isaacson, E. and Keller, H.B. (1966). *Analysis of Numerical Methods*, John Wiley and Sons, New York.

J

- Jack, C., Kai, S., and Shulman, A. (1992). "NETCAP—An Interactive Optimization System for GTE Telephone Network Planning," *Interfaces* **22**:1, 72–89.
- Jackson, J.R. (1957). "Simulation Research on Job Shop Production," *Naval Research Logistics Quarterly*, **1**, 287.
- Jacobs, W.W. (1954). "The Caterer Problem," *Naval Research Logistics Quarterly*, **1**, 154–165.
- Jankowski, M. and Wozniakowski, M. (1977). "Iterative Refinement Implies Numerical Stability," *BIT* **17**, 303–311.
- Jewell, W.S. (1958). "Optimal Flows Through Networks," Interim Technical Report No. 8, on Fundamental Investigations in Methods of Operations Research, Massachusetts Institute of Technology, Cambridge, Massachusetts.
- Johnson, D.B. (1977). "Efficient Algorithms for Shortest Paths in Sparse Networks," *Journal of the Association for Computing Machinery* **24**, 1–13.
- Jones, M.T. and Plassmann, P.E. (1995). "An Improved Incomplete Cholesky Factorization," *ACM Transactions on Mathematical Software* **21**, 5–17.
- Judin, D.B., and Nemirovskii, A.S. (1976a). "Estimation of the Informational Complexity of Mathematical Programming Problems," *Ekonomika i Matematicheskie Metody*, **12**, 128–142. (English translation in *Matekon: Translation of Russian and East European Mathematical Economics*, **13**, 3–25, 1977.)
- Judin, D.B., and Nemirovskii, A.S. (1976b). "Informational Complexity and Effective Methods for the Solution of Convex Extremal Problems," *Ekonomika i Matematicheskie Metody*, **12**, 357–369. (English translation in *Matekon: Translation of Russian and East European Mathematical Economics*, **13**, 25–45, 1977.)
- Judin, D.B., and Nemirovskii, A.S. (1976c). "Informational Complexity of Strict Convex Programming," *Ekonomika i Matematicheskie Metody*, **12**, 550–559.

K

- Kahan, W. (1966). "Numerical Linear Algebra," *Canadian Mathematical Bulletin* **9**, 757–801.
- Kahaner, D., Moler, C.B., and Nash, S. (1988). *Numerical Methods and Software*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Kalaba, R.E. and Juncosa, M.L. (1956). "Optimal Design and Utilization of Communications Networks," *Management Science* **3**, 33–44.
- Kalan, J.E. (1971). "Aspects of Large-Scale In-Core Linear Programming," *Proceedings of the ACM Annual Conference*, 304–313.
- Kalan, J.E. (1976). "Machine Inspired Enhancements of the Simplex Algorithm," Technical Report CS75001-R, Computer Science Department, Virginia Polytechnical University, Blacksburg, Virginia.
- Kall, P. (1974). "Approximations to Stochastic Programs with Complete Fixed Recourse," *Numerische Mathematik* **22**, 333–339.
- Kall, P. (1979). "Computational Methods for Two Stage Stochastic Linear Programming Problems," *Zeitschrift für angewandte Mathematik und Physik* **30**, 261–271.
- Kall, P. and Stoyan, D. (1982). "Solving Stochastic Programming Problems with Recourse Including Error Bounds," *Mathematische Operationsforschung und Statistik, Series Optimization* **13**, 431–447.
- Kall, P. and Wallace, S.W. (1994). *Stochastic Programming*, John Wiley and Sons, New York.
- Kantorovich, L.V. (1939). "Mathematical Methods in the Organization and Planning of Production," Publication House of the Leningrad State University; translated in *Management Science* **6**, 1960, 366–422.
- Kantorovich, L. V. (1942). "On the Translocation of Masses," *Comptes Rendus de l'Académie des Sciences U.S.S.R.* **37**, 199–201.
- Kantorovich, L.V. and Gavurin, M.K. (1949). "The Application of Mathematical Methods to Freight Flow Analysis," (translation), in: Collection of Problems of Raising the Efficiency of Transport Performance, Akademiia Nauk SSSR, Moscow-Leningrad, 1949, 110–138.
- Karmarkar N. (1984). "A New Polynomial-Time Algorithm for Linear Programming," *Combinatorica* **4**, 373–395.
- Katz, P., Sadrian, A., and Tendick, P. (1994). "Telephone Companies Analyze Price Quotations with Bellcore's PDSS Software," *Interfaces* **24**:1, 50–63.
- Kaul, R.N. (1965). "An Extension of Generalized Upper Bounding Techniques for Linear Programming," Technical Report ORC 65-27, Operations Research Center, University of California, Berkeley.
- Kennedy, W.J., Jr. and Gentle, J.E. (1980). *Statistical Computing*, Marcel Dekker, Inc. New York and Basel.
- Kennington, J.L. and Wang, Z. (1991). "An Empirical Analysis of the Dense Assignment Problem: Sequential and Parallel Implementations," *ORSA Journal on Computing* **3**, 299–306.

- Kernighan, B.W. and Plauser, P.J. (1970). *The Elements of Programming Style*, McGraw-Hill, New York.
- Khachian, L.G. (1979). "A Polynomial Algorithm for Linear Programming," *Doklady Akademii Nauk USSR* **244**, 1093–1096. [English translation: *Soviet Mathematics Doklady* **20**, 191–194].
- Khan, M.R. and Lewis, D.A. (1987). "A Network Model for Nurse Staff Scheduling," *Zeitschrift für Operations Research* **31**, B161–B171.
- Kotiah, T.C.T. and Steinberg, D.I. (1977). "Occurrences in Cycling and Other Phenomena Arising in a Class of Linear Programming Models," *Communications of the Association for Computing Machinery* **20**, 107–112.
- Kotiah, T.C.T. and Steinberg, D.I. (1978). "On the Possibility of Cycling with the Simplex Method," *Operations Research* **26**, 374–375.
- Kim, K. and Nazareth, J.L. (1994). "A Primal Null-Space Affine Scaling Method," *ACM Transactions on Mathematical Software* **20**, 373–392.
- Klee, V.L. and Minty, G.J. (1972). "How Good is the Simplex Algorithm?" in O. Shisha (ed.), *Inequalities III*, Academic Press, London and New York, 159–175.
- Klee, V.L. and Walkup, D.W. (1967). "The d -Step Conjecture for Polyhedra of Dimension $d < 6$," *Acta Mathematica* **117**, 53–78.
- Klingman, D., Phillips, N., Steiger, D., Wirth, R., and Young, W. (1986). "The Challenges and Success Factors in Implementing an Integrated Products Planning System for Citgo," *Interfaces* **16**:, 1–19.
- Klingman, D., Phillips, N., Steiger, D., and Young, W. (1987). "The Successful Deployment of Management Science throughout Citgo Petroleum Corporation," *Interfaces* **17**:, 4–25.
- Klotz, E.S. (1988). "Dynamic Pricing Criteria in Linear Programming," Technical Report SOL 88-15, Department of Operations Research, Stanford University, Stanford, CA.
- Kohler, D.A. (1973). "Translation of a Report by Fourier on His Work on Linear Inequalities," *OPSEARCH* **10**, 38–42.
- Kojima, M., Megiddo, N., and Mizuno, S. (1993). "A Primal-Dual Infeasible Interior Point Algorithm for Linear Programming," *Mathematical Programming* **61**, 263–280.
- Kojima, M., Mizuno, S., and Yoshise, A. (1989a). "A Primal-Dual Interior Point Method for Linear Programming," in N. Megiddo (ed.), *Progress in Mathematical Programming: Interior Point and Related Methods* Springer-Verlag, Berlin and New York, 29–47.
- Kojima, M., Mizuno, S., and Yoshise, A. (1989b). "A Polynomial-Time Algorithm for a Class of Linear Complementarity Problems," *Mathematical Programming* **44**, 1–26.
- Koopmans, T.C. (1947). "Optimum Utilization of the Transportation System," *Proceedings of the International Statistical Conference*, Washington, DC. Volume 5 was reprinted as a supplement to *Econometrica*, **17**, 1949.
- Koopmans, T.C. (ed.), (1951). *Activity Analysis of Production and Allocation*, John-Wiley and Sons, New York.

- Koopmans, T.C. and Reiter, S. (1951). "A Model of Transportation," in T.C. Koopmans (ed.), *Activity Analysis of Production and Allocation*, July–October 1949, Cowles Commission Monograph 13, Proceedings of Linear Programming Conference, June 20–24, 1949, John Wiley and Sons, New York, 33–97.
- Kortanek, K.O. and Zhu, J. (1988). "New Purification Algorithms for Linear Programming," *Naval Research Logistics Quarterly*, **35**, 571–583.
- Knuth, D.E. (1973a). *Fundamental Algorithms, The Art of Computer Programming, Vol. 1*, Addison-Wesley Publishing Company, Reading, Massachusetts; originally published in 1968.
- Knuth, D.E. (1973b). *Sorting and Searching, The Art of Computer Programming, Vol. 3*, Addison-Wesley Publishing Company, Reading, Massachusetts.
- Knuth, D.E. (1981). *Seminumerical Algorithms, The Art of Computer Programming, Vol. 2*, Addison-Wesley Publishing Company, Reading, Massachusetts; originally published in 1969.
- Kranich, E. (1991). "Interior Point Methods for Mathematical Programming: A Bibliography Discussion Paper 171, Institute of Economy and Operations Research, Fern Universität Hagen, P.O. Box 940, D-5800 Hagen 1, West Germany.
- Krishna, A.S. (1989). Unpublished Manuscript.
- Krishna, A.S. (1993). "Enhanced Algorithms for Stochastic Programming," Technical Report SOL 93-8, Department of Operations Research, Stanford University, Stanford, CA.
- Kruskal, J.B., Jr. (1956). "On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem," *Proceedings of the American Mathematical Society* **7**, 48–50.
- Kuhn, H.W. (1955). "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, **2**, 1 and 2, 83–97.
- Kuhn, H.W. (1956). "Solvability and Consistency for Linear Equations and Inequalities," *The American Mathematical Monthly* **63**, 217–232.
- Kuhn, H.W. and Tucker, A.W. (1950). "Nonlinear Programming," in J. Neyman (ed.), *Proceedings 2nd Berkeley Symposium on Mathematical Statistics and Probability, 1950*, University of California Press, Berkeley, California, 481–492.
- Kuhn, H.W. and Tucker, A.W. (1958). "John von Neumann's Work in the Theory of Games and Mathematical Economics," *Bulletin of the American Mathematical Society* **64**, 100–122.
- Kusy, M.I. and Ziemba, W.T. (1986). "A Bank Asset and Liability Management Model," *Operations Research* **34**, 356–378.

L

- Laderman, J. (1947).
- Lansdowne, Z.F. (1979). "Survey of Research on Model Simplification," Technical Report SOL 79-26, Department of Operations Research, Stanford University, Stanford, CA.
- Lasdon, L.S. (1970). *Optimization Theory for Large Systems*, Macmillan, London.

- Lawler, E.L. (1976). *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, New York.
- Lawson, C.L. and Hanson, R.J. (1974). *Solving Least Squares Problems*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Lawson, C.L., Hanson R.J., Kincaird, D.R., and Krogh, F.T. (1979a). "Basic Linear Algebra Subprograms for Fortran Usage," *ACM Transactions on Mathematical Software* **5**, 308–323.
- Lawson, C.L., Hanson R.J., Kincaird, D.R., and Krogh, F.T. (1979b). "Algorithm 539, Basic Linear Algebra Subprograms for Fortran Usage," *ACM Transactions on Mathematical Software* **5**, 324–325.
- Lee, S.M. (1972). *Goal Programming for Decision Analysis*, Auerbach, Philadelphia.
- Leichner, S.A., Dantzig, G.B., and Davis, J.W. (1993). "A Strictly Improving Linear Programming Phase I Algorithm," *Annals of Operations Research* **14**, 409–430.
- Lemke, C.E. (1954). "The Dual Method of Solving the Linear Programming Problem," *Naval Research Logistics Quarterly*, **1**, 36–47.
- Lenstra, J.K., Rinnooy Kan, A.H.G., and Schrijver, A. (eds.) (1991). *History of Mathematical Programming*, Elsevier Science Publishing Company, New York.
- Levin, A.Y. (1965). "On an Algorithm for the Minimization of Convex Functions," *Soviet Mathematics Doklady* **6**, 286–290.
- Lewis, R.E. (1955). "Top Management Looks at Linear Programming and Inventory Management," *Proceedings of the Linear Programming and Inventory Management Seminar*, Methods Engineering Council, Pittsburgh, B-1 to B-8.
- Litty, C.J. (1994). "Optimal Lease Structuring at GE Capital," *Interfaces* **24**:3, 34–45.
- Liu, J. (1985). "Modification of the Minimum-Degree Algorithm by Multiple Elimination," *ACM Transactions on Mathematical Software* **11**, 141–153.
- Luenberger, D.G. (1989). *Introduction to Linear and Non-Linear Programming*, Addison-Wesley Publishing Company, Reading, Massachusetts.
- Lustig, I.J. (1987). "An Analysis of an Available Set of Linear Programming Test Problems," Technical Report SOL 87-11, Department of Operations Research, Stanford University, Stanford, CA.
- Lustig, I.J. (1991). "Feasibility Issues in an Interior Point Method for Linear Programming," *Mathematical Programming* **49**, 145–162.
- Lustig, I.J., Marsten, R.E., and Shanno, D.F. (1990). "The Primal-Dual Interior Point Method on the Cray Supercomputer, in T.F. Coleman and Y. Li (eds.), *Large-Scale Numerical Optimization*, papers from the workshop held at Cornell University, Ithaca, New York, October 1989, vol. 46 of SIAM Proceedings in Applied Mathematics, Society of Industrial and Applied Mathematics, Philadelphia, PA, 70–80.
- Lustig, I.J., Marsten, R.E., and Shanno, D.F. (1991a). "Computational Experience with a Primal-Dual Interior Point for Linear Programming," *Linear Algebra and Its Applications* **152**, 191–222.
- Lustig, I.J., Marsten, R.E., and Shanno, D.F. (1991b). "Interior Method vs Simplex Method: Beyond NETLIB," *Mathematical Programming Society COAL Newsletter* **19**, 41–44.

- Lustig, I.J., Marsten, R.E., and Shanno, D.F. (1992a). "Computational Experience with a Globally Convergent Primal-Dual Predictor-Corrector Algorithm for Linear Programming," Technical Report SOR 92-10, School of Engineering and Applied Science, Department of Civil Engineering and Operations Research, Princeton University, Princeton, New Jersey.
- Lustig, I.J., Marsten, R.E., and Shanno, D.F. (1992b). "The Interaction of Algorithms and Architectures for Interior Point Methods, in P.M. Pardalos (ed.), *Advances in Optimization and Parallel Computing*, North-Holland, the Netherlands, 190–205.
- Lustig, I.J., Marsten, R.E., and Shanno, D.F. (1992c). "On Implementing Mehrotra's Predictor-Corrector Interior Point Method for Linear Programming," *SIAM Journal on Optimization* **2**, 435–449.
- Lustig, I.J., Marsten, R.E., and Shanno, D.F. (1994). "Interior Point Methods for Linear Programming: Computational State of the Art," *ORSA Journal on Computing* **6**, 1–14.
- Lustig, I.J., Mulvey, J.M., and Carpenter, T.J. (1991). "Formulating Two-Stage Stochastic Programs for Interior Point Methods," *Operations Research* **39**, 757–770; previously published in 1990 as Technical Report SOR 89-16, School of Engineering and Applied Science, Department of Civil Engineering and Operations Research, Princeton University, Princeton, New Jersey.

M

- Madansky, A. (1959). "Bounds on the Expectation of a Convex Function of a Multivariate Random Variable," *Annals of Mathematical Statistics* **30**, 743–746.
- Makuch, W.M., Dodge, J.L., Ecker, J.G., Granfors, D.C., and Hahn, G.J. (1992). "Managing Consumer Credit Delinquency in the US Economy: A Multi-billion Dollar Management Science Application," *Interfaces* **22:1**, 90–109.
- Malhotra, V.M., Kumar, M.P., and Maheshwari, S.N. (1978). "An $O(|V|^3)$ Algorithm for Finding Maximum Flows in Networks," *Information Processing Letters*, **7**, 277–278.
- Manley, B.R. and Threadgill, J.A. (1991). "LP Used for Valuation and Planning of New Zealand Plantation Forests," *Interfaces* **21:6**, 66–79.
- Manne, A.S. (1956). *Scheduling of Petroleum Refinery Operations*, Harvard University Press, Cambridge, Massachusetts.
- Manne, A.S. (1974). "Waiting for the Breeder," in *Review of Economic Studies Symposium*, 47–65.
- Marcus, M. (1960). "Some Properties of Doubly Stochastic Matrices," *The American Mathematical Monthly* **67**, 215–221.
- Markowitz, H.M. (1952). "Portfolio Selection," *The Journal of Finance* **7**, 77–91.
- Markowitz, H.M. (1957). "The Elimination Form of the Inverse and its Applications to Linear Programming," *Management Science* **3**, 255–269.
- Markowitz, H.M. (1959). *Portfolio Selection: Efficient Diversification of Investments*, John Wiley and Sons, New York.

- Maros, I.G. (1986). "A General Phase-I Method in Linear Programming," *European Journal of Operations Research* **23**, 64–77.
- Marshall, K.T. and Suurballe, J.W. (1969). "A Note on Cycling in the Simplex Method," *Naval Research Logistics Quarterly*, **16**, 121–137.
- Marsten, R.E. (1981). "The Design of the XMP Linear Programming Library," *ACM Transactions on Mathematical Software* **7**, 481–497.
- Marsten, R.E., Saltzman, M.J., Shanno, D.F., Ballinton, J.F., and Pierce, G.S. (1989). "Implementation of a Dual Affine Interior Point Algorithm for Linear Programming," *ORSA Journal on Computing* **1**, 287–297.
- Mascarenhas, W.F. (1993). "The Affine Scaling Algorithm Fails for $\lambda = 0.999$," Technical Report, Universidade Estadual de Campinas, Campinas S. P., Brazil.
- Massé, P. and Gibrat, R. (1957). "Applications of Linear Programming to Investments in the Electric Power Industry," *Management Science* **3**, 149–166.
- Maynard, H.B. (1955). "Putting New Management Tools to Work," *Proceedings of the Linear Programming and Inventory Management Seminar*, Methods Engineering Council, Pittsburgh.
- McCarthy, C. and Strang, G. (1973). "Optimal Conditioning of Matrices," *SIAM Journal on Numerical Analysis* **10**, 370–388.
- McCormick, S.T. (1983). "Optimal Approximation of Sparse Hessians and its Equivalence to a Graph Coloring Problem," *Mathematical Programming* **26**, 153–171.
- McDiarmid, C. (1990). "On the Improvement per Iteration in Karmarkar's Algorithm for Linear Programming," *Mathematical Programming* **46**, 299–320.
- McGowan, C.L. and Kelly, J.R. (1975). *Top-Down Structured Programming Techniques*, Petrocelli/Charter, New York.
- McKeeman, W.M. (1962). "Crout with Equilibration and Iteration," *Communications of the Association for Computing Machinery* **5**, 553–555.
- McShane, K.A., Monma, C.L., and Shanno, D.F. (1989). "An Implementation of a Primal-Dual Interior Point Method for Linear Programming," *ORSA Journal on Computing* **1**, 70–83.
- Megiddo, N. (1986). "Introduction: New Approaches to Linear Programming," special issue of *Algorithmica* **1**, 387–394, Springer-Verlag, Berlin and New York.
- Megiddo, N. (1988). "Pathways to the Optimal Set in Linear Programming," in N. Megiddo (ed.), *Progress in Mathematical Programming: Interior Point and Related Methods* Springer-Verlag, Berlin and New York, 131–158.
- Megiddo, N. (1991). "On Finding Primal- and Dual-Optimal Bases," *ORSA Journal on Computing* **3**, 63–65.
- Mehrotra, S. (1992a). "On the Implementation of a Primal-Dual Interior Point Method," *SIAM Journal on Optimization* **2**, 575–601.
- Mehrotra, S. (1992b). "Implementation of Affine Scaling Methods: Approximate Solutions of Systems of Linear Equations Using Preconditioned Conjugate Gradient Methods," *ORSA Journal on Computing* **4**, 103–118.
- Mehrotra, S. (1993). "Quadratic Convergence in a Primal-Dual Method," *Mathematics of Operations Research* **18**, 741–751.

- Meijerink, J. and van der Vorst, H.A. (1981). "Guidelines for the Usage of Incomplete Decomposition in Solving Sets of Linear Equations as They Occur in Practical Problems," *Journal of Computational Physics* **44**, 134–155.
- Millham, C.B. (1976). "Fast Feasibility Methods for Linear Programming," *OPSEARCH* **13**, 198–204.
- Minkowski, H. (1896). *Geometrie der Zahlen*, B.G. Teubner, Leipzig and Berlin, 1910. First printing, 1896; also reprinted by Chelsea Publishing Company, New York, 1953.
- Mitchell, J.E. and Borchers, B. (1992). "A Primal-Dual Interior Point Cutting Plane Method for the Linear Ordering Problem," *COAL Bulletin* **21**, 13–18.
- Mitchell, J.E. and Todd, M.J. (1992). "Solving Combinatorial Optimization Problems Using Karmarkar's Algorithm," *Mathematical Programming* **56**, 245–284.
- Mizuno, S. (1992). "Polynomiality of the Kojima-Megiddo-Mizuno Infeasible Interior Point Algorithm for Linear Programming," Technical Report 1006, School of Operations Research and Industrial College of Engineering, Cornell University, Ithaca, New York 14853.
- Mizuno, S., Kojima, M., and Todd, M. (1995). "Infeasible-Interior-Point-Primal-Dual Potential-Reduction Algorithms for Linear Programming," *SIAM Journal on Optimization* **5**, 52–67.
- Monma, C.L. and Morton, A.J. (1987). "Computational Experiments with a Dual Affine Variant of Karmarkar's Method for Linear Programming," *Operations Research Letters* **6**, 261–267.
- Monteiro, R.C. and Adler, I. (1989a). "Interior Path Following Primal-Dual Algorithms—Part I: Linear Programming," *Mathematical Programming* **44**, 27–42.
- Monteiro, R.C. and Adler, I. (1989b). "Interior Path Following Primal-Dual Algorithms—Part II: Convex Quadratic Programming," *Mathematical Programming* **44**, 43–66.
- Monteiro, R.C., Tsuchiya, T., and Wang, Y. (1993). "A Simplified Global Convergence Proof of the Affine Scaling Algorithm," *Annals of Operations Research* **47**, 443–482.
- Mood, A.M., Graybill, F.A., and Boes, D.C. (1974). *Introduction to the Theory of Statistics*, McGraw-Hill, New York.
- Mooney, J.W. (1975). "Organized Program Maintenance," *Datamation* **21**, 63–64.
- Moore, E.F. (1959). "The Shortest Path Through a Maze," in *Proceedings of the International Symposium on the Theory of Switching*, Harvard University Press, Cambridge, Massachusetts, 285–292.
- More, J.J. and Sorenson, D.C. (1981). "Computing a Trust Region step," Report ANL-81-83, Argonne National Laboratory, Argonne, Illinois.
- Morin, G. (1955). "More Effective Production Planning with Linear Programming," Paper F from *Proceedings of the Linear Programming and Inventory Management Seminar*, Methods Engineering Council, Pittsburgh.
- Morris, W.T. (1967). "On the Art of Modeling," *Management Science* **13**, 707–717.
- Morton, D.P. (1993). "Algorithmic Advances in Multi-Stage Stochastic Programming," Ph.D. thesis, Department of Operations Research, Stanford University, Stanford, CA.

- Morton, D.P. (1995). "Stochastic Network Interdiction," working paper presented at a Workshop in Stochastic Programming at University of California, Davis, March, 1995.
- Motzkin, T.S. (1936). "Beiträge zur Theorie der linearen Ungleichungen," doctoral thesis, University of Zurich.
- Muir, T. (1960). *Determinants*, Dover Publications, New York.
- Mulvey, J.M. (1978). "Pivot Strategies for Primal-Simplex Network Codes," *Journal of the Association for Computing Machinery* **25**, 266–270.
- Munksgaard, N. (1980). "Solving Sparse Symmetric Sets of Linear Equations by Preconditioned Conjugate Gradient," *ACM Transactions on Mathematical Software* **6**, 206–219.
- Murray, W. (1976). "Constrained Optimization" in L.C.W. Dixon (ed.), *Optimization in Action*, Academic Press, London and New York, 217–251.
- Murray, W. and Wright, M.H. (1978). *Projected Lagrangian Methods Based on Trajectories of Penalty and Barrier Functions*, Technical Report SOL 78-23, Department of Operations Research, Stanford University, Stanford, CA.
- Murtagh, B.A. and Saunders, M.A. (1978). "Large-Scale Linearly Constrained Optimization," *Mathematical Programming* **14**, 41–72.
- Murtagh, B.A. and Saunders, M.A. (1995). "MINOS 5.4 User's Guide," Technical Report SOL 83-20R, Department of Operations Research, Stanford University, Stanford, CA.
- Murty, K.G. (1980). "Computational Complexity of Parametric Linear Programming," *Mathematical Programming* **19**, 213–219.
- Murty, K.G. (1983). *Linear Programming*, John Wiley and Sons, New York.
- Musalem, J. (1980). "Converting Linear Models to Network Models," Ph.D. thesis, University of California, Los Angeles, California.
- Myers, D.C. and Shih, W. (1988). "A Constraint Selection Technique for a Class of Linear Programs," *Operations Research Letters* **7**, 191–194.

N

- Nazareth, J.L. (1984). "Numerical Behavior of LP Algorithms Based Upon the Decomposition Principle," *Linear Algebra and Its Applications* **57**, 181–189.
- Nazareth, J.L. (1986). "Implementation Aids for the Optimization Algorithms that Solve Sequences of Linear Programs," *ACM Transactions on Mathematical Software* **12**, 307–323.
- Nazareth, J.L. (1987). *Computer Solution of Linear Programs*, Monographs on Numerical Analysis, Oxford University Press, Oxford and New York.
- Nering, E.D. and Tucker, A.W. (1993). *Linear Programs and Related Problems*, Academic Press, London and New York.
- Ng, E. (1988). "On the Solution of Sparse Linear Least-Squares Problems," Presentation at Stanford, Mathematical Science Section, Oak Ridge National Laboratory, Oak Ridge, Tennessee.
- Nishiya, T. (1983). "A Basis Factorization Method for Multi-Stage Linear Programming with an Application to Optimal Operation of an Energy Plant," Draft Report.

O

- Oettli, W., and Prager, W. (1964). "Computability of Approximate Solutions of Linear Equations With Given Error Bounds For Coefficients and Right-Hand Sides," *Numerische Mathematik* **6**, 405–409.
- O'Leary, D.P. (1980). "Estimating Matrix Condition Numbers," *SIAM Journal on Scientific and Statistical Computing* **1**, 205–209.
- Orchard-Hays, W. (1954). *A Composite Simplex Algorithm—II, RM-1275*, The RAND Corporation, May.
- Orchard-Hays, W. (1956). "Evolution of Computer Codes for Linear Programming," The RAND Corporation, Paper P-810, (March 14), 22–24.
- Orchard-Hays, W. (1968). *Advanced Linear-Programming Computing Techniques*, McGraw-Hill, New York.
- Orchard-Hays, W. (1978). "History of Mathematical Programming Systems," in H.J. Greenberg (ed.), *Design and Implementation of Optimization Software*, Sijthoff and Noordhoff, Alphen aan den Rijn, 1–26.
- Orchard-Hays, W., Cutler, L., and Judd, H. (1956). "Manual for the RAND IBM Code for Linear Programming on the 704," The RAND Corporation, Paper P-842, (May 16), 24–26.
- Orden, A. (1956). "The Transshipment Problem," *Management Science* **2**, 276–285.
- Orden, A. (1993). "LP from the 40s to the 90s," *Interfaces* **23**:5, 2–12.
- Orlin, J.B. and Ahuja, R.K. (1992). "New Scaling Algorithms for the Assignment and Minimum Cycle Mean Problems," *Mathematical Programming* **54**, 41–56.
- Ortega, J.M. (1973). *Numerical Analysis: A Second Course*, Academic Press, London and New York.
- Ortega, J.M. (1988). "The *ijk* Forms of Factorization Methods I: Vector Computers," *Parallel Computing* **7**, 135–147.
- Ortega, J.M. and Rheinboldt, W.C. (1970). *Iterative Solutions of Nonlinear Equations in Several Variables*, Academic Press, London and New York.
- Osterby, O. and Zlatev Z. (1983). "Direct Methods for Sparse Matrices," *Lecture Notes in Computer Science* **157**, Springer-Verlag, Berlin and New York.

P

- Pan, V. (1984). "How Can We Speed Up Matrix Multiplication," *SIAM Review* **26**, 393–416.
- Papadimitriou, C.H. and Steiglitz, K. (1982). *Combinatorial Optimization, Algorithms and Complexity*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Pape, U. (1974). "Implementation and Efficiency of Moore Algorithms for the Shortest Root Problem," *Mathematical Programming* **7**, 212–222.
- Parlett, B.N., and Wang, Y. (1975). "The Influence of the Compiler on the Cost of Mathematical Software—in Particular on the Cost of Triangular Factorization," *ACM Transactions on Mathematical Software* **1**, 35–46.

- Paull, A. E. and Walter, J.R. (1955). "The Trim Problem: An Application of Linear Programming to the Manufacture of Newsprint Paper," *Econometrica*, **23**, 336 (abstract).
- Pereira, M.V. and Pinto, L.M.V.G. (1989). "Stochastic Dual Dynamic Programming," Technical Note, DEE-PUC/RJ-Catholic University of Rio de Janeiro, Caixa Postal 38063 Gávea, Rio de Janeiro, RJ CEP, Brazil.
- Pereira, M.V., Pinto, L.M.V.G., Oliveira, G.C., and Cunha, S.H.F. (1989). "A Technique for Solving LP-Problems with Stochastic Right-Hand Sides," CEPEL, Centro del Pesquisas de Energia Electria, Rio de Janeiro, Brazil.
- Perold, A. F. (1984). "Large Scale Portfolio Optimizations," *Management Science* **30**, 1143-1160.
- Peters, G. and Wilkinson, J.H. (1970). "The least-squares problem and pseudo-inverses," *Computer Journal* **13**, 309-316.
- Pissanetzky, Sergio (1984). *Sparse Matrix Technology*, Academic Press, London and New York.
- Powell, M.J.D. (1970). A Hybrid Method for Nonlinear Equations, in P. Rabinowitz (ed.), *Numerical Methods for Nonlinear Algebraic Equations*, Academic Press, London and New York, 29-55.
- Powell, M.J.D. (1971). "On the Convergence of the Variable Metric Algorithm," *Journal of the Institute of Mathematics and Its Applications* **7**, 21-36.
- Powell, M.J.D. (1976). "Some Global Convergence Properties of a Variable Metric Algorithm Without Exact Line Searches," in R.C. Cottle and C.E. Lemke (eds.), *SIAM-AMS proceedings, Volume IX, Mathematical Programming*, American Mathematical Society, Providence, Rhode Island, 53-72.
- Powell, M.J.D. (1991). "On the Number of Iterations of Karmarkar's Algorithm for Linear Programming," Technical Report DAMTP 1991/NA23, Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK.
- Powell, M.J.D. and Toint, Ph.L. (1979). "On the Estimation of Sparse Hessian Matrices," *SIAM Journal on Numerical Analysis* **16**, 1060-1073.
- Prékopa, A. (1990). "Sharp Bounds on Probabilities Using Linear Programming," *Operations Research* **38**, 227-239.

R

- Reid, J.K. (1971). "A Note on the Stability of Gaussian Elimination," *Journal of the Institute of Mathematics and Its Applications* **8**, 374-375.
- Reid, J.K. (1976). "Fortran Subroutines for Handling Sparse Linear Programming Bases," *Report AERE R8269*, Atomic Energy Research Establishment, Harwell, England.
- Reid, J.K. (1982). "A Sparsity-Exploiting Variant of the Bartels-Golub Decomposition for Linear Programming Bases," *Mathematical Programming* **24**, 55-69.
- Reinfeld, N.V., and Vogel, W.R. (1958). *Mathematical Programming*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

- Renegar, J. (1988). "A Polynomial-Time Algorithm, Based on Newton's Method, for Linear Programming," *Mathematical Programming* **40**, 59–93.
- Rice, J.R. (1966a). "Experiments with Gram-Schmidt orthogonalization," *Mathematics of Computation* **20**, 325–328.
- Rice, J.R. (1966b). "A Theory of Condition," *SIAM Journal on Numerical Analysis* **3**, 287–310.
- Rice, J.R. (1976a). "Parallel Algorithm for Adaptive Quadrature III: Program Correctness," *ACM Transactions on Mathematical Software* **2**, 1–30.
- Rice, J.R. (1976b). "The Algorithm Selection Problem," in M. Rubicoff and M. Yovits (eds.), *Advances in Computers*, Vol. 15, Academic Press, London and New York, 65–118.
- Rice, J.R. (1985). *Matrix Computations and Mathematical Software*, McGraw-Hill, San Francisco, California.
- Rigby, B., Lasdon, L.L., and Waren, A.D. (1995). "The Evolution of Texaco's Blending Systems from OMEGA to StarBlend," *Interfaces* **25**:5, 64–83.
- Robertson, H.H. (1977). "The Accuracy of Error Estimates for Systems of Linear Algebraic Equations," *Journal of the Institute of Mathematics and Its Applications* **20**, 409–414.
- Rockafellar, R.T. (1970). *Convex Analysis*, Princeton University Press, Princeton, New Jersey.
- Rockafellar, R.T. (1984). *Network Flows and Monotropic Optimization*, Wiley, New York.
- Rockafellar, R.T. and Wets, R.J. (1989). "Scenario and Policy Aggregation in Optimization under Uncertainty," *Mathematics of Operations Research* **16**, 119–147.
- Rose, D.J. and Willoughby, R.A. (eds.) (1972). *Sparse Matrices and Their Applications*, Plenum Press, New York, 67–76.
- Rosen, S. (1969). "Electronic Computers: A Historical Survey," *ACM Computing Surveys* **1**, 7–36.
- Rosen, J.B. (1960). "The Gradient Projection Method for Nonlinear Programming, Part I—Linear Constraints," *SIAM Journal of Applied Mathematics* **9**, 181–217.
- Rosen, J.B. (1964). "Primal Partition Programming for Block Diagonal Matrices," *Numerische Mathematik* **6**, 250–260.
- Roush, W.B., Stock, R.H., Cravener, T.L., and D'Alfonso, T.H. (1994). "Using Chance-Constrained Programming for Animal Feed Formulation at Agway," *Interfaces* **24**:2, 53–58.
- Roy, J. and Crainic, T.G. (1992). "Improving Freight Routing with a Tactical Planning Model," *Interfaces* **22**:3, 31–44.
- Russel, E.J. (1969). "Extensions of Dantzig's Algorithm for Finding an Initial Near-Optimal Basis for the Transportation Problem," *Operations Research* **17**, 187–191.
- Ruszczynski, A. (1986). "A Regularized Decomposition Method for Minimizing a Sum of Polyhedral Functions," *Mathematical Programming* **35**, 309–333.
- Ryan, D.M., and Osborne, M.R. (1988). "On the Solution of Highly Degenerate Linear Programs," *Mathematical Programming* **41**, 385–392.
- Ryder, B.G. (1974). "The PFORT Verifier," *Software Practice and Experience* **4**, 359–378.

S

- Saigal, R. (1992). "A Simple Proof of the Primal Affine Scaling Method," Technical Report 92-60, Department of Industrial Engineering and Operations Research, University of Michigan, Ann Arbor, Michigan, to appear in *Annals of Operations Research*.
- Saigal, R. (1993a). "A Three Step Quadratically Convergent Implementation of the Primal Affine Scaling Method," Technical Report 93-9, Department of Industrial Engineering and Operations Research, University of Michigan, Ann Arbor, Michigan.
- Saigal, R. (1993b). "The Primal Power Affine Scaling Method," Technical Report 93-21, Department of Industrial Engineering and Operations Research, University of Michigan, Ann Arbor, Michigan, to appear in *Annals of Operations Research*.
- Sakarovitch, M. and Saigal, R. (1967). "An Extension of Generalized Upper Bounding Techniques for Structured Linear Programs," *SIAM Journal of Applied Mathematics* **15**, 4, 906–914.
- Salveson, M.E. (1953). "A Computational Technique for the Fabrication Scheduling Problem," Management Sciences Research Project, University of California, Los Angeles, California.
- Samelson, H. (1974). *An Introduction to Linear Algebra*, John Wiley and Sons, New York.
- Sargent, R.W.H., and Westerberg, A.W. (1964). "'Speed-up' in Chemical Engineering Design," *Transactions of the Institute of Chemical Engineers*, **42**, 190–197.
- Saunders, M.A. (1972). "Large-Scale Linear Programming Using the Cholesky Factorization," Report STAN-CS-72-252, Department of Computer Science, Stanford University, Stanford, CA.
- Saunders, M.A. (1976a). "The Complexity of LU Updating in the Simplex Method," in R. S. Anderssen and R. P. Brent (eds.), *The Complexity of Computational Problem Solving*, University Press, Queensland, 214–230.
- Saunders, M.A. (1976b). "A Fast, Stable Implementation of the Simplex Method Using Bartels-Golub Updating," in J.R. Bunch and D.J. Rose (eds.), *Sparse Matrix Computations*, Academic Press, London and New York, 213–226.
- Saunders, M.A. (1979). "Sparse Least Squares by Conjugate Gradients: A Comparison of Preconditioning Methods," in J.F. Gentlemen (ed.), *Computer Science and Statistics: 12th Annual Symposium on the Interface*, University of Waterloo, Waterloo, Ontario, Canada, 15–20.
- Saunders, M.A. (1980). "Large-scale Linear Programming," Notes for the tutorial conference *Practical Optimization*, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA.
- Schönauer, W. (1987). *Scientific Computing on Vector Computers*, North-Holland, Amsterdam, the Netherlands.
- Schrage, L. (1975). "Implicit Representation of Variable Upper Bounds in Linear Programming," *Mathematical Programming Study* **4**, 118–132.
- Schrage, L. (1981). "Some Comments on Hidden Structure in Linear Programs," in H. Greenberg and J. Maybee (eds.), *Computer Assisted Analysis and Model Simplification*, Academic Press, London and New York, 389–395.

- Scott, D.M. (1985). "A Dynamic Programming Approach to Time-Staged Convex Programs," Technical Report SOL 85-3, Department of Operations Research, Stanford University, Stanford, CA.
- Sethi, A.P. and Thompson, G.L. (1984). "The Pivot and Probe Algorithm for Solving a Linear Program," *Mathematical Programming* **29**, 219–233.
- Shapiro, J.F. (1979). *Mathematical Programming: Structures and Algorithms*, John Wiley and Sons, New York.
- Shapiro, R.D. (1984). *Optimization Models for Planning and Allocation: Text and Cases in Mathematical Programming*, Wiley, New York.
- Shanno, D.F. and Bagchi, A. (1990). "A Unified View of Interior Point Methods for Linear Programming," *Annals of Operations Research* **22**, 55–70.
- Sharpe, W.F. (1971). "A Linear Programming Approximation for the General Portfolio Selection Problem," *Journal of Financial Quantitative Analysis* **6**, 1263–1275.
- Shefi, A. (1969). "Reduction of Linear Inequality Constraints and Determination of All Feasible Extreme Points," Ph.D. thesis, Department of Engineering-Economic Systems, Stanford University, Stanford, CA.
- Sherman, A.H. (1978). "Algorithms for Sparse Gaussian Elimination with Partial Pivoting," *ACM Transactions on Mathematical Software* **4**, 330–338.
- Sherman, D.H. and Ladino, G. (1995). "Managing Bank Productivity Using Data Envelopment Analysis (DEA)," *Interfaces* **25:2**, 60–73.
- Sherman, J. and Morrison, W.J. (1949). "Adjustment of an Inverse of Matrix Corresponding to Changes in Elements of a Given Column or a Given Row of the Original Matrix," *Annals of Mathematical Statistics* **20**, 621.
- Shor, N.Z. (1970a). "Utilization of the Operation of Space Dilatation in the Minimization of Convex Functions," *Kibernetika* **6**, 6–11; English translation in *Cybernetics* **6**, 7–15.
- Shor, N.Z. (1970b). "Convergence Rate of the Gradient Descent Method with Dilatation of the Space," *Kibernetika* **6**, 80–85; English translation in *Cybernetics* **6**, 102–108.
- Shor, N.Z. (1971a). "A Minimization Method Using the Operation of Extension of the Space in the Direction of the Difference of Two Successive Gradients," *Kibernetika* **7**, 51–59; English translation in *Cybernetics* **7**, 450–459.
- Shor, N.Z. (1971b). "Certain Questions of Convergence of Generalized Gradient Descent," *Kibernetika* **7**, 82–84; English translation in *Cybernetics* **7**, 1033–1036.
- Shor, N.Z. (1972a). "Solution of the Minimax Problems by the Method of Generalized Gradient Descent with Dilatation of the Space," *Kibernetika* **8**, 82–88; English translation in *Cybernetics* **8**, 88–94.
- Shor, N.Z. (1972b). "A Class of Almost-Differentiable Functions and a Minimization Method for Functions in this Class," *Kibernetika* **8**, 65–70; English translation in *Cybernetics* **8**, 599–606.
- Shor, N.Z. (1975). "Convergence of a Gradient Method with Space Dilatation in the Direction of the Difference Between Two Successive Gradients," *Kibernetika* **11**, 48–53; English translation in *Cybernetics* **11**, 564–570.

- Shor, N.Z. (1977a). "Cut-off Method with Space Extension in Convex Programming Problems," *Kibernetika* **13**, 94–95; English translation in *Cybernetics* **13**, 94–96.
- Shor, N.Z. (1977b). "New Development Trends in Nondifferentiable Optimization," *Kibernetika* **13**, 87–91; English translation in *Cybernetics* **13**, 881–886.
- Simonard, M. (1966). *Linear Programming*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey; English translation by W. S. Jewell.
- Sinha, G.P., Chandrasekaran, B.S., Mitter, N., Dutta, G., Singh, S.B., Choudhry, A.R., and Roy, P.N. (1995). "Strategic and Operational Management with Optimization at Tata Steel," *Interfaces* **25:1**, 6–19.
- Skeel, R.D. (1979). "Scaling for Numerical Stability in Gaussian Elimination," *Journal of the Association for Computing Machinery* **26**, 494–526.
- Skeel, R.D. (1980). "Iterative Refinement Implies Numerical Stability for Gaussian Elimination," *Mathematics of Computation* **35**, 817–832.
- Sleator, D.K. (1980). "An $O(nm \log n)$ Algorithm for Maximum Network Flows," Ph.D. thesis, Department of Computer Science, Stanford University, Stanford, CA.
- Smale, S. (1976). "A Convergent Process of Price Adjustment and Global Newton Methods," *Journal of Mathematical Economics* **3**, 107–120.
- Smale, S. (1982). "On the Average Speed of the Simplex Method of Linear Programming," Department of Mathematics, University of California, Berkeley.
- Smale, S. (1983). "The Problem of the Average Speed of the Simplex Method," in A. Bachem, M. Grötschel, and B. Korte (eds.), *Mathematical Programming: The State of the Art*, Springer-Verlag, Berlin and New York, , 530–539.
- Smith, B.T., Boyle, J.M., Dongarra, J.J., Garbow, B.S., Ikebe, Y., Klema, V.C., and Moler, C.B. (1976). *Matrix Systems Routines—EISPACK Guide, Lecture Notes in Computer Science* **6**, Springer-Verlag, Berlin and New York.
- Solow, D. (1984). *Linear Programming: An Introduction to Finite Improvement Algorithms*, North-Holland, Amsterdam, the Netherlands.
- Sonnevend, G. (1986). "An 'Analytic Center' for Polyhedrons and New Classes of Global Algorithms for Linear (smooth, convex) Programming, in A. Prékopa, J. Szelezsan, and B. Strazicky (eds.), *System Modelling and Optimization: Proceedings of the 12th IFIP Conference Held in Budapest, Hungary, September, 1985, Volume 84 of Lecture Notes in Control and Information Science*, Springer-Verlag, Berlin, West Germany, 866–876.
- Sorensen, D. (1980). "Newton's Method with a Model Trust Region Modification," Report ANL-80-106, Argonne National Laboratory, Argonne, Illinois.
- Sorensen, D. (1985). "Analysis of Pairwise Pivoting in Gaussian Elimination," *IEEE Transactions on Computers*, **C-34**, 274–278.
- Sponk, J. (1981). *Interactive Multiple Goal Programming: Applications to Financial Management*, Martinus Nijhoff, Boston, Massachusetts.
- Steinitz, E. (1913). *Journal für die Reine and Angewandte Mathematik* **143**, 128–275.
- Steuer, R.E. (1985). *Multiple Criteria Optimization: Theory, Computation, and Application*, Wiley, New York

- Stiemke, E. (1915). "Über positive Lösungen homogener linearer Gleichungen," *Mathematical Annals* **76**, 340–342.
- Stigler, G.J. (1945). "The Cost of Subsistence," *Journal of Farm Economics* **27**, 303–314.
- Starfield, A.M., Smith, K. A., and Bleloch, A. L. (1990). *How to Model It—Problem Solving for the Computer Age*, McGraw-Hill, New York.
- Stewart, G.W. (1973). *Introduction to Matrix Computations*, Academic Press, London and New York.
- Strang, G. (1986). *Introduction to Applied Mathematics*, Wellesley-Cambridge Press, Massachusetts.
- Strang, G. (1988). *Linear Algebra*, Academic Press, London and New York.
- Strassen, V. (1969). "Gaussian Elimination is Not Optimal," *Numerische Mathematik* **13**, 354–356.
- Stone, H.S. (1972). *Introduction to Computer Organization and Data Structures*, McGraw-Hill, San Francisco, California.

T

- Tarjan, R.E. (1972). "Depth-First Search and Linear Graph Algorithms," *SIAM Journal on Computing* **1**, 146–160.
- Thapa, M.N. (1976). "Operations Research Techniques in Iron and Steel Making," B. Tech. Thesis, Indian Institute of Technology, Bombay, India.
- Thapa, M.N. (1983). "Optimization of Unconstrained Functions with Sparse Hessian Matrices—Quasi-Newton Methods," *Mathematical Programming* **25**, 158–182.
- Thapa, M.N. (1984a). "Optimal Leveraged Leasing Within LEA2," Final Report for ATEL, Inc.
- Thapa, M.N. (1984b). "Optimization of Unconstrained Functions with Sparse Hessian Matrices—Newton-Type Methods," *Mathematical Programming* **29**, 156–186.
- Thapa, M.N. (1991). "A Gasoline/Diesel Retail Distribution Model," Final Report for United Refining Company.
- Thapa, M.N. (1992). "A Multi-time Period Asphalt Distribution Model," Final Report for United Refining Company.
- Thie, P.R. (1988). *An Introduction to Linear Programming and Game Theory*, John Wiley and Sons, New York.
- Tilanus, C.B., DeGans, O.B., and Lenstra, J.K. (1986). *Quantitative Methods in Management: Case Studies of Failures and Successes*, Wiley, New York.
- Todd, M.J. (1980). "The Monotonic Hirsch Conjecture is False for Dimension at Least 4," *Mathematics of Operations Research* **5**, 599–601.
- Todd, M.J. (1982). "An Implementation of the Simplex Method for Linear Programming Problems with Variable Upper Bounds," *Mathematical Programming* **23**, 23–49.
- Todd, M.J. (1983). "Large Scale Linear Programming: Geometry, Working Bases and Factorizations," *Mathematical Programming* **26**, 1–20.

- Todd, M.J. (1989). "Recent Developments and New Directions in Linear Programming," in M. Iri and K. Tanabe (eds.), *Mathematical Programming: Recent Developments and Applications*, Kluwer Academic Press, Dordrecht, the Netherlands, 109–157.
- Todd, M.J. (1990a). "The Effects of Degeneracy, Null, and Unbounded Reduction Algorithm for Linear Programming," Technical Report No. 902, School of Operations Research and Industrial College of Engineering, Cornell University, Ithaca, New York 14853.
- Todd, M.J. (1990b). "Combining Phase I and Phase II in a Potential Reduction Algorithm for Linear Programming," Technical Report No. 902, School of Operations Research and Industrial College of Engineering, Cornell University, Ithaca, New York 14853.
- Todd, M.J. (1994a). "Theory and Practice of Interior-Point Methods," *ORSA Journal on Computing* **6**, 28–31.
- Todd, M.J. (1994b). "A Lower Bound on the Number of Iterations of Primal-Dual Interior Point Methods for Linear Programming," Technical Report 1050, School of Operations Research and Industrial College of Engineering, Cornell University, Ithaca, New York 14853.
- Todd, M.J. and Burrell, B.P. (1986). "An Extension of Karmarkar's Algorithm for Linear Programming Using Dual Variables," special issue of *Algorithmica* **1**, Springer-Verlag, Berlin and New York, 409–424.
- Toint, Ph.L. (1977). "On Sparse and Symmetric Matrix Updating Subject to a Linear Equation," *Mathematics of Computation* **32**, 839–851.
- Toint, Ph.L. (1979). "On the Superlinear Convergence of an Algorithm for Solving a Sparse Minimization Problem," *SIAM Journal on Numerical Analysis* **16**.
- Tomlin, J.A. (1972a). "Pivoting for Size and Sparsity in Linear Programming Inversion Routines," *Journal of the Institute of Mathematics and Its Applications* **10**, 289–295.
- Tomlin, J.A. (1972b). "Modifying Triangular Factors of the Basis in the Simplex Method," in D.J. Rose and R.A. Willoughby (eds.), *Sparse Matrices and their Applications*, Plenum Press, New York, .
- Tomlin, J.A. (1975a). "LPM1 User's Guide," Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA.
- Tomlin, J.A. (1975b). "On Scaling Linear Programming Problems," *Mathematical Programming Study* **4**, 146–166.
- Tomlin, J.A. (1975c). "An Accuracy Test for Updating Triangular Factors," *Mathematical Programming Study* **4**, 142–145.
- Tomlin, J.A. (1987). "An Experimental Approach to Karmarkar's Projective Method for Linear Programming," *Mathematical Programming Study* **31**, 175–191.
- Tomlin, J.A. and Welch, J.S. (1983a). "Formal Optimization of some Reduced Linear Programming Problems", *Mathematical Programming* **27**, 232–240.
- Tomlin, J.A. and Welch, J.S. (1983b). "A Pathological Case in the Reduction of Linear Programs," *Operations Research Letters* **2**, 53–57.
- Tomlin, J.A. and Welch, J.S. (1985). "Integration of a Primal Simplex Network Algorithm with a Large-Scale Mathematical Programming System," *ACM Transactions on Mathematical Programming* **II**, 1–11.

- Tomlin, J.A. and Welch, J.S. (1986). "Finding Duplicate Rows in a Linear Programming Model," *Operations Research Letters* **5**, 7–11.
- Tompkins, C.B. (1955). "Projection Methods in Calculation," in H.A. Antosiewicz (ed.), *Proceedings of the Second Symposium in Linear Programming*, Vol. 2, National Bureau of Standards and Directorate of Management Analysis, DCS/Comptroller, USAF, Washington, DC, 425–488. 1036–1045.
- Tompkins, C.B. (1957). "Some Methods of Computational Attack on Programming Problems, Other than the Simplex Method," *Naval Research Logistics Quarterly*, **4**, 95–96.
- Tsuchiya, T. and Monteiro, R.D.C. (1996). "Superlinear Convergence of the Affine Scaling Algorithm," *Mathematical Programming* **75**, 77–110.
- Tsuchiya, T. and Muramatsu, M. (1995). "Global Convergence of the Long-Step Affine Scaling Algorithm for Degenerate Linear Programming Problems," *SIAM Journal on Optimization* **5**, 525–551; previously published as a Research Memorandum 423, The Institute of Statistical Mathematics, 4-6-7 Minami-Azubu, Minato-ku, Tokyo 106, Japan, 1992.
- Tucker, A.W. (1955). "Linear Inequalities and Convex Polyhedral Sets," in H.A. Antosiewicz (ed.), *Proceedings of the Second Symposium in Linear Programming*, Vol. 2, National Bureau of Standards and Directorate of Management Analysis, DCS/Comptroller, USAF, Washington, DC, 569–602.
- Tucker, A.W. (1956). "Dual Systems of Homogeneous Linear Relations," in H.W. Kuhn and A.W. Tucker (eds.), *Linear Inequalities and Related Systems*, Annals of Mathematics Study No. 38, Princeton University Press, Princeton, New Jersey, 3–18.

U

- Ubhaya, V.A. (1974a). "Isotone Optimization, I," *Journal of Approximation Theory* **12**, 146–159.
- Ubhaya, V.A. (1974b). "Isotone Optimization, II," *Journal of Approximation Theory* **12**, 315–331.

V

- Vajda, S. (1961). *Mathematical Programming*, Addison-Wesley Publishing Company, Reading, Massachusetts.
- Vanderbei, R.J. (1991). "Splitting Dense Columns in Sparse Linear Systems," *Linear Algebra and Its Applications* **152**, 107–117.
- Vanderbei, R.J. (1992). "LOQO User's Manual," School of Engineering and Applied Science, Department of Civil Engineering and Operations Research, Princeton University, Princeton, New Jersey, 92–95.
- Vanderbei, R.J. (1994). "Interior Point Methods: Algorithms and Formulations," *ORSA Journal on Computing* **6**, 32–34.

- Vanderbei, R.J. (1995). "Symmetric Quasi-Definite Matrices," *SIAM Journal on Optimization* **5**, 100-113; previously published in 1991 in School of Engineering and Applied Science, Department of Civil Engineering and Operations Research, Princeton University, Princeton, New Jersey 91-100.
- Vanderbei, R.J. and Carpenter, T.J. (1993). "Symmetric Indefinite Systems for Interior Point Methods," *Mathematical Programming* **58**, 1-32.
- Vanderbei, R.J. and Lagarias, J.C. (1988). "I.I. Dikin's Convergence Result for the Affine-Scaling Algorithm," in J.C. Lagarias and M.J. Todd (eds.), *Mathematical Developments Arising from Linear Programming*, Contemporary Mathematics, 109-119.
- Vanderbei, R.J., Meketon, M.S., and Freedman, B.A. (1986). "A Modification of Karmarkar's Linear Programming Algorithm," *Algorithmica* **1**, 395-407.
- Van Loan, C. (1985). "How Near is a Stable Matrix to an Unstable Matrix?" *Contemporary Mathematics* **47**, 465-477.
- Van Slyke, R.M. and Wets, R.J. (1969). "L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming," *SIAM Journal of Applied Mathematics* **17**, 638-663.
- Van Slyke, R.M. and Wets, R.J. (1966). "Programming Under Uncertainty and Stochastic Optimal," *SIAM Journal on Control and Optimization* **4**, 179-193.
- Van Tassel, D. (1974). *Program Style, Design, Efficiency, Debugging and Testing*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- van de Panne, C. and Whinston, A. (1969). "The Symmetric Foundation of the Simplex Method for Quadratic Programming," *Econometrica*, **37**, 507-527.
- Varga, R.S. (1962). *Matrix Iterative Analysis*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Vavasis, S.A. (1993). "Stable Finite Elements for Problems with Wild Coefficients," Report TR 93-1364, Department of Computer Science, Cornell University, Ithaca, New York.
- Vavasis, S.A. (1994). "Stable Numerical Algorithms for Equilibrium Systems," *SIAM Journal on Matrix Analysis and Applications* **15:4**, 1108-1131.
- Vertinsky, I., Brown, S., Schreier, H., Thompson, W.A., and van Kooten, G.C. (1994). "A Hierarchical-GIS Based Decision Model for Forest Management: The Systems Approach," *Interfaces* **24:4**, 38-53.
- Ville, J.A. (1938). "Sur la théorie général des jeux où intervient l'habileté des joueurs," *Applications aux jeux de hasard* by Emil Borel and Jean Ville, Tome 4, Fascicule 2, in *Traité du Calculus des Probabilités et de ses Applications*, by Emile Borel, 105-113.
- Vliet, A.V., Boender, C.G., and Rinnooy Kan, A.H.G. (1991). "Interactive Optimization of Bulk Sugar Deliveries," *Interfaces* **22:3**, 4-14.
- von Neumann, J. (1928). "Zur Theorie de Geselleschaftsspiele," *Mathematical Annals* **100**, 295-320; translated by Sonya Bargmann in H.W. Kuhn and A.W. Tucker (eds.), *Contributions to the Theory of Games*, Vol. IV, Annals of Mathematics Study No. 40, Princeton University Press, Princeton, New Jersey, 13-42.
- von Neumann, J. (1937). "Über ein Ökonomisches Gleichungssystem und ein Verallgemeinerung des Brouwerschen Fixpunktsatzes," *Ergebnisse eines Mathematischen Kolloquiums*, No. 8; translated in *The Review of Economic studies* **13**, 1-9.

- von Neumann, J. (1947). "On a Maximization Problem,," (manuscript), Institute for Advanced Study, Princeton, New Jersey.
- von Neumann, J. (1948a). "A Numerical Method for the Determination of the Value and Best Strategies of a Zero-Sum Two-Person Game," (manuscript), Institute for Advanced Study, Princeton, New Jersey.
- von Neumann, J. (1948b). Private Communication on an Interior Method.
- von Neumann, J. (1953). "A Certain Zero-Sum Two-Person Game Equivalent to the Optimal Assignment Problem," in H.W. Kuhn and A.W. Tucker (eds.), *Contributions to the Theory of Games*, Vol. 2, Annals of Mathematics Study No. 28, Princeton University Press, Princeton, New Jersey, 12–15.
- von Neumann, J. and Goldstine, H.H. (1947). "Numerical Inverting of Matrices of High Order," *Bulletin of the American Mathematical Society* **53**, 1021–1089.
- von Neumann, J. and Morgenstern, O. (1944). *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, New Jersey.

W

- Wegner, P. (1968). *Programming Languages, Information Structures and Machine Organization*, McGraw-Hill, New York.
- Wets, R.J. (1984). "Programming Under Uncertainty: The Equivalent Convex Program," *SIAM Journal of Applied Mathematics* **14**, 89–105.
- Whiting, P.D. and Hillier, J.A. (1960). "A Method for Finding the Shortest Route Through a Road Network," *Operations Research Quarterly* **11**, 37–40.
- Wilkinson, J.H. (1961). "Error Analysis of Direct Methods of Matrix Inversion," *Journal of the Association for Computing Machinery* **8**, 281–330.
- Wilkinson, J.H. (1963). *Rounding Errors in Algebraic Processes*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Wilkinson, J.H. (1965). *The Algebraic Eigenvalue Problem*, Oxford University Press, Oxford and New York.
- Wilkinson, J.H. (1971). "Modern Error Analysis," *SIAM Review* **13**, 548–568.
- Wilkinson, J.H. and Reinsch, C. (eds.), (1971). *Handbook for Automatic Computation, Vol. 2, Linear Algebra*, Springer-Verlag, Berlin and New York.
- Williams, H.P. (1985). *Model Building in Operations Research*, Wiley, New York.
- Wittrock, R.J. (1983). "Advances in a Nested Decomposition Algorithm for Solving Staircase Linear Programs," Technical Report SOL 83-2, Department of Operations Research, Stanford University, Stanford, CA.
- Wolfe, P. (1960). "Accelerating the Cutting Plane Method for Nonlinear Programming," *Journal of the Society for Industrial and Applied Mathematics*, **9**, 481–488.
- Wolfe, P. (1962). "The Reduced-Gradient Method," unpublished manuscript, RAND Corporation.
- Wolfe, P. (1963). "A Technique for Resolving Degeneracy in Linear Programming," *SIAM Journal of Applied Mathematics* **11**, 205–211.

- Wolfe, P. (1965). "The Composite Simplex Algorithm," *SIAM Review* **7**, 42–55.
- Wolfe, P. (1967). "Methods for Nonlinear Programming," in J. Abadie (ed.), *Nonlinear Programming*, North-Holland, Amsterdam, the Netherlands, 67–86.
- Wolfe, P. and Cutler, L. (1963). "Experiments in Linear Programming," in R.L. Graves and P. Wolfe (eds.), *Recent Advances in Mathematical Programming*, McGraw-Hill, New York, 177–200.
- Wright, M.H. (1976). *Numerical Methods for Nonlinearly Constrained Optimization*, Ph.D. thesis, Department of Computer Science, Stanford University, Stanford, CA.
- Wright, M.H. (1992). "Interior Methods for Constrained Optimization," in A. Iserles (ed.), *Acta Numerica*, Cambridge University Press, New York, New York, 341–407.
- Wright, M.H. and Glassman, S.C. (1978). "FORTRAN Subroutines to Solve the Linear Least Squares Problem and Compute the Complete Orthogonal Factorization," Technical Report SOL 78-8, Department of Operations Research, Stanford University, Stanford, CA.
- Wright, S.J. (1993). "A Superlinear Infeasible-Interior-Point Algorithm for Monotone Nonlinear Complementarity Problems, Technical Report MCS-P344-1292, Mathematical and Computer Science Division, Argonne National Laboratory, Argonne, Illinois.
- Wright, W. (1980). "Automatic Identification of Network Rows in Large-Scale Optimization Models," M.S. thesis, Naval Postgraduate School, Monterey, CA.

Y

- Yamada, T. and Kitahara, T. (1985). "Qualitative Properties of Systems of Linear Constraints," *Journal of the Operations Research Society of Japan* **28**, 331–343.
- Yamashita, H. (1986). "A Polynomially and Quadratically Convergent Method for Linear Programming," working paper, Mathematical Systems Institute, Inc., Tokyo, Japan.
- Ye, Y. (1987). "Eliminating Columns in the Simplex Method for Linear Programming," Technical Report SOL 87-14, Department of Operations Research, Stanford University, Stanford, CA.
- Ye, Y. (1990). "A 'Build-Down' Scheme for Linear Programming," *Mathematical Programming* **46**, 61–72.
- Ye, Y. (1997). *Interior Point Algorithms: Theory and Analysis*, John Wiley and Sons, New York.
- Ye, Y., Güler, O., Tapia, R.A., and Zhang, Y. (1993). "A Quadratically Convergent $O(\sqrt{n}L)$ -Iteration Algorithm for Linear Programming," *Mathematical Programming* **59**, 151–162.
- Ye, Y., Todd, M.J., and Mizuno, S. (1994). "An $O(\sqrt{n}L)$ -Iteration Homogeneous and Self-Dual Linear Programming Algorithm," *Mathematics of Operations Research* **19**, 53–67.
- Ye, Y. and Tse, E. (1989). "An Extension of Karmarkar's Projective Algorithm for Convex Quadratic Programming," *Mathematical Programming* **44**, 157–180.
- Yohe, J.M. (1974). "An Overview of Programming Practices," *ACM Computing Surveys* **6**, 221–245.
- Young, D.M. (1971). *Iterative Solution for Large Linear Systems*, Academic Press, London and New York.

Z

- Zadeh, N. (1973). "A Bad Network Problem for the Simplex Method and Other Minimum Cost Flow Algorithms," *Mathematical Programming* **5**, 255–266.
- Zaki, H. (1990). "A Comparison of Two Algorithms for the Assignment Problem," Technical Report ORL-90-002, Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Champaign, Urbana, IL.
- Zangwill, W.I. (1969). *Nonlinear Programming: A Unified Approach*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
- Ziemba, W.J. (1970). "Computational Algorithms for Convex Stochastic Programs with Simple Recourse," *Operations Research* **18**, 273–294.
- Ziemba, W.J. (1974). "Stochastic Programs with Simple Recourse," in P.L. Hammond and H. Zoutendijk (eds.), *Mathematical Programming: Theory and Practice*, North-Holland, Amsterdam, the Netherlands.
- Zikan K. and Cottle R.W. (1987). "The Box Method for Linear Programming: Part I—Basic Theory," Technical Report SOL 87-6, Department of Operations Research, Stanford University, Stanford, CA.
- Zhang, Y. (1994). "On the Convergence of a Class of Infeasible Interior-Point Methods for the Horizontal Linear-Complementarity Problem," *SIAM Journal on Optimization* **4**, 208–227.
- Zhang, Y., Tapia, R.A., and Dennis, J.E. (1992). "On the Superlinear and Quadratic Convergence of Primal-Dual Interior Point Linear Programming Algorithms," *SIAM Journal on Optimization* **2**, 304–324.
- Zhang, Y. and Zhang, D. (1995). "On Polynomiality of the Mehrotra-Type Predictor-Corrector Interior Point Algorithms," *Mathematical Programming* **68**, 303–318.
- Zlatev, Z. (1980). "On Some Pivotal Strategies in Gaussian Elimination by Sparse Technique," *SIAM Journal on Numerical Analysis* **17**, 12–30.
- Zlatev, Z., Wasniewski, J., and Schaumburg, K. (1981). "Y12M—Solution of Large and Sparse Systems of Linear Equations," *Lecture Notes in Computer Science* **121**, Springer-Verlag, Berlin and New York.

Index

- ϵ -Optimal Polynomial Time, 70
- 2-Stage Stochastic LP
 - algorithm, 341
 - deterministic case, 335
 - estimating lower bound z_{LB} , 352
 - estimating upper bound z_{UB} , 351
 - Full Formulation, 337
 - geometric description, 338
 - theory behind the algorithm, 348
- Abadie, J., 59
- Abrahamson, P., 361
- Adjacent, 18
- Adjusted costs, 287, 302, 307
- Adler, I., 120, 142, 143, 316
- Affine combination, 1, 2
- Affine hull, 2
- Affine transformation, 133
- Ahuja, R., 243, 244
- Ali, A., 244
- Alternatives
 - theorems, 43
- Analytic center, 141
- Andersen, E., 146
- Anstreicher, K., 120, 121
- anstreicher, K., 141
- Anti-cycling, 149
- Arc, 231
 - saturated, 240
- Argmin, xxv
- Associated network, 235
- Augmenting path algorithm, 243
- Avriel, M., 31, 146, 361
- Axelsson, O., 144
- Bagchi, A., 140
- Bahn, O., 121
- Balanski, M., 166
- Ballinton, J., 142
- Barnes, E., 120, 141
- Barrier function methods, 128
 - primal logarithmic, 131
 - primal-dual, 134
- Basis norm, 197
- Basis triangularity, 210
- Bastian, M., 264
- Bazaraa, M., 38, 244
- Beale's Example, 150
- Beale, E., 150, 166, 201, 316, 332, 360
- Beale, M., 324
- Bellman, R., 244
- Ben-Tal, A., 361
- Benders decomposition, 299
 - dual of Dantzig-Wolfe decomposition, 299
 - using, 309
- Benders Full Master, 300, 301
- Benders, J., 316, 361
- Benichou, M., 166
- Bennet, 264
- Berge, C., 31
- Berry-Esséen, 361
- Bertsekas, D., 245
- Bertsekas, D., 243
- Bipartite graph, 245
- Birge, J., 332, 360, 361, 363
- Bisschop, J., 316
- Bixby, R., 146, 200, 201
- Björck, A., 202
- Bland's Rule, 158–160

- Bland, R., 166
Block-angular, 266
Block-angular system, 306
Block-triangular, 267
Blossom algorithm, 201
Borchers, B., 143
Borgwardt, K., 31
Bradley, G., 244
Bradley, S., 316
Breadth-first unblocked search, 237
Brown, G., 244, 264
Brumelle, S., 363
Burrel, B., 121
- Canonical form, 176
Carpenter, T., 143–145
Center, 141
Center-of-Gravity Problem, 72
Central Limit Theorem, 372
Charnes, A., 146, 166, 169, 264
Cherkassky, B., 244
Chi-square distribution, 373
Choi, I., 142
Cholesky factorization, 140
 partial, 144
Chvátal, V., 227, 244
Classical transportation problem, *see*
 transportation problem
Column-restricted primal, 191–196
Complementary pair, 56
Complementary slackness, 49–50
Cone, 9
 convex polyhedral, 10
 simplicial, 10
Confidence interval, 376
 one sided, 377
Confidence level, 377
Confidence limit, 377
Conjugate gradient method, 144
Conservation of flows, 232
Convex combination, 1, 2
Convex hull, 2, 35
Convex polygon, 9
Convex polyhedral cone, 10
Convex polyhedron, 9
Convex programming, 120
Convex regions
 general, 6
 two-dimensional, 3
Convex set, 4
 closed, 5
Convexity
 feasible solutions, 9
 half-space, 8
 hyperplane, 8
Convexity and linear inequality
 systems, 1
Convexity constraint, 2
Correlation coefficient, 369
Correspondence of primal and dual
 bases, 174
Covariance, 369
Cover, 245
CPM, 231, 244
Criss-Cross Method, 188
Cross-over to a basic solution, 137
Cubic fit, 115
Cunha, S., 332, 360
Cunningham, W., 227, 244
Cut, 239
 feasibility, 302
 in a network, *see* Maximal flow
 optimality, 302
 separate, 239
 value, 240
Cut value, 241
Cutler, L., 201
Cuts, 302
- D-W, 266
D-W Decomposition Principle, 280
Dantzig's Inductive Method, 154
Dantzig, G. B., 32–37, 58, 59, 70, 71,
 119, 120, 138, 146, 154, 166,
 169, 200–202, 243, 244, 264,
 315–317, 324, 332, 333, 360,
 361
Dantzig, T., 59
Dantzig-Wolfe Decomposition
 Principle, 266, 280

-
- Davis, J., 202
 - Davis, P., 361
 - de Ghellinck, G., 121
 - Decomposition
 - nested, 316
 - Benders, 299
 - central planning, 313
 - D-W, 280
 - D-W Principle, 266, 284
 - Dantzig-Wolfe, 280
 - illustrated, 292
 - initial solution, 291
 - staircase, 308
 - Decomposition of flow, 233
 - Decopositon
 - D-W Principle, 280
 - Degeneracy, 149–171
 - Bland's Rule, 158–160
 - example, 16
 - examples of cycling, 149
 - Beale's example, 150
 - Hoffman's example, 149
 - Tucker's Example, 150
 - Extra Column Rule, 160–164
 - Inductive Method, 154
 - lexicographic method, 169
 - lexicographic rule, 168
 - on resolving, 153
 - order of, 164
 - perturbation method, 167, 169
 - random choice rule, 158, 167
 - resolving, 149
 - transportation problem, 216
 - Wolfe's Rule, 156–158
 - Degenerate dual, 175
 - Degenerate pivots
 - avoidance, 164–165
 - Delayed Column Generation
 - Procedure, 288
 - Delayed Row Generation, 299
 - Dempster, M., 316
 - Denardo, E., 244
 - Dennis, J., 143
 - density, 79
 - Descent direction, 124
 - Deák, I., 361
 - Dial, R., 244
 - Dijkstra, E., 244
 - Dikin's algorithm, 87, 88
 - assumptions, 89
 - convergence, 89
 - using t in S , 93
 - ellipsoid strictly in interior, 91
 - proof of convergence, 93
 - valid steps, 90
 - Dikin's method, 68, 84–100, 133
 - dual solution, 85
 - ellipsoid, 86
 - ellipsoid subproblem, 88
 - interior solution, 85
 - partition of indices, 85
 - primal solution, 85
 - Dikin, I., 68, 84, 119, 120, 141, 142
 - Dilworth's theorem, 249
 - Dinic, E., 244
 - Dual Simplex Method, 179
 - correspondence with primal, 180
 - Duality
 - Additional Theorem, 47
 - Duality Gap, 45
 - Duality Theorem, 45, 59
 - Von Neumann, 45
 - Duality theorem, 43
 - Duality theorems, 43
 - Duff, I., 144
 - Edge, 14, 18
 - Edmonds, J., 243
 - Edmondson, J., 166
 - Egerváry, E., 201, 243
 - El-Bakry, A., 143
 - Elias, P., 243
 - Ellipsoid, 86
 - Elmaghraby, S., 245
 - Enriken, R., 316, 361
 - Erisman, A., 144
 - Ermoliev, Y., 332, 360–362
 - Event, 367
 - Expected value, 368
 - Extra Column Rule, 160–164

- Extremal problem, 285
Extreme homogeneous normalized solution, 280
Extreme point, 14, 15
 degenerate, 16

Fanning out procedure, 237
Farkas's Lemma, 59
 for homogeneous systems, 52
Farkas, J., 59
Faulkner, J., 167
Feasibility cut, 302
Feinstein, A., 243
Feller, W., 332, 377
Ferguson, A., 324, 332, 360
Fiacco, A., 119, 141, 146
Fletcher, R., 166
Flow value, 235
Ford, L., 201, 243, 316
Forrest, J., 200, 201
Forsgren, A., 145
Fourer, R., 144, 145, 316
Fourier-Motzkin Elimination, 35
Fox, B., 244
Fraley, C., 121
Frauendorfer, K., 332, 360, 361
Fredman, M., 244
Freedman, B., 120, 141
Freund, R., 121, 141
Frisch, K., 119, 141
Fulkerson, D., 201, 243, 316
Full Master, 285
 Benders, 300
Full Master Program, 307

Gabow, H., 244
Gaivoronski, A., 362
Gale, D., 59
Galil, Z., 244
Gallo, G., 244
Gamma function, 374
Gass, S., 166, 201
Gassner, B., 227
Gauthier, J., 166
Gay, D., 121, 142

General position, 14
Generalized Upper Bounds, 251
Generic Column, 273
Geoffrion, A., 361
Geometry of Vector Spaces, 1
George, A., 144
Gill, P., 31, 140, 144–146, 166, 167
Glassey, C., 316
Global minimum, 14, 15
Glynn, P., 361
Goffin, J., 121
Goldberg, A., 244
Goldfarb, D., 200, 201
Goldman, A., 59, 316
Gomory, R., 166, 202
Gonzaga, C., 121, 140, 141
Gordan's theorem for homogeneous systems, 51
Gordan, P., 59
Gradient vector, 123
Graves, G., 244
Grcar, J., 144
Greenberg, H., 166
Gregory, J., 146
Grötschel, M., 120
GUB, 251
Güler, O., 143

Hadley, G., 202
Half line, 5
Half-line, 6, 8
Half-plane, 8
Half-space, 7, 8
 closed, 8
 convexity, 8
Hall's theorem, 249
Hall, L., 141, 146
Hall, P., 361
Hammersly, J., 361
Handscomb, D., 361
Hanson, R., 202
Harris, P., 166
Hax, A., 316
Helgason, R., 244
Hentges, J., 166

- Hertog, D., 141
 Hessian matrix, 123
 Hight, J., 332, 360
 Hillier, F., 244
 Hillier, J., 244
 Hirsch conjecture, 25, 31, 33
 Hirsch, W. M., 25, 31, 33, 316
 Ho, J., 316, 317
 Hoffman's Example, 149
 Hoffman, A., 119, 149, 316
 Hoffman, A., 166
 Holmes, D., 360
 homogeneous inequalities, 47
 Homogeneous system, 59
 Huang, C., 361
 Huard, P., 141
 Hughes, 316
 Hungarian Method, 201
 Hyperplane, 7
 - convexity, 8
 - lowest dimensional, 8
 - separating, 11
 - supporting, 339
- Iglehart, D., 361
 Importance sampling, 350, 356
 Independent
 - distribution, 369
 Independent random variables, 369
 Inductive Method, 154
 Infanger, G., 360, 361
 Infeasibility Theorem, 11, 44, 59
 Infeasible equation, 44
 Interior solution to basic solution, 137
 Interior-point
 - potential reduction, 67
 Interior-point methods, 123–147
 - affine, 67
 - Early, 67–122
 - path-following, 67
 - primal affine, 67
 - projective, 67
 - versus Simplex Method, 69
- Jarvis, J., 38, 244
 Jensen's inequality, 361
 Jewell, W., 316
 Johnson, D., 244
 Johnson, L., 227
 Jones, M., 144
 Judd, H., 201
 Judin, D., 120
- König-Evary, 249
 Kalan, J., 166
 Kall, P., 332, 360, 361
 Kallio, M., 316
 Karmarkar's algorithm, 114–115
 - versus von Neumann Algorithms, 118
 - computational comments, 116
 - converting an LP, 115
 - null space, 104
 Karmarkar's method, 68, 100–118
 - assumptions, 102
 - inverse mapping, 101
 - inverse projective transformation, 101
 - mapping \mathcal{T} , 101
 - projective algorithm, 101
 - projective transformation, 101
 - proof of convergence, 105
 - simplex, 102
 - center, 102
 - radius of circumscribed sphere, 102
 - radius of inscribed sphere, 102
 - standard form LP, 102
- Karmarkar, N., 68, 120, 140, 142
 Karp, R., 243
 Kaul, R., 264
 Kelley, 36
 Kennington, J., 244
 Khachian's method, 68
 Khachian, I., 119, 120
 Khachian, L., 68
 Kim, K., 144
 Kirchoff's Law, 232
 Klee, V., 31, 201

- Klincewicz, J., 227, 244
Klotz, E., 31, 166
Kojima, M., 142, 143
Koopmans, T., 227, 228
Kortanek, K., 146
Kotiah, T., 166
Krishna's Rule, 160–164
Krishna, A., 166
Kuhn, H., 59, 146, 201, 243, 315
Kumar, M., 244
- Lagarias, J., 120
Lagrange multipliers, 59
Lall, H., 244
Law of large numbers, 372
Lawler, E., 243
Lawson, C., 202
Leichner, S., 202
Lemke, C., 200, 264
Levin, A., 120
Lexico-positive, 169
Lexicographic method, 169
Lexicographic Rule, 168
Lieberman, G., 244
Line search, 115, 121
Line segment, 5
Linear combination, 1
 nonnegative, 2
Linear least-squares problem, 127
Liu, J., 144
Local minimum, 15
Loute, E., 316
Louveaux, F., 332, 360, 363
Lovász, L., 120
Lower-blocked, 238
Lustig, I., 31, 120, 141–146
- Madansky, A., 332, 360
Magnanti, M., 316
Magnanti, T., 243
Maheshwari, S., 244
Malhotra, V., 244
Manne, A., 201, 316
Mannos, M., 119
Marginal Column, 208
Marginal row, 208
Markowitz, H., 144, 324, 332
Marshall, K., 166
Marsten, R., 120, 141–143, 145, 146
Mascarenhas, W., 141
Master Program, 285
 Benders, 301
 full, 285
 restricted, 286
Matching, 245
Matching problem, 201
Matrix
 lower triangular, 210
 triangular, 210
 upper triangular, 210
Max improvement per iteration, 176
Maximal flow, 232–241
 cuts in a network, 239–241
Maximal matching, 245
McCormick, G., 119, 141, 146
McDiarmid, C., 120
McGill, J., 363
McShane, K., 142
Mean, 368
Meeraus, A., 316
Megiddo, N., 142, 143, 146
Mehlhorn, K., 244
Mehrotra, S., 142–145
Meijerink, J., 144
Meketon, M., 120, 141
Merle, O., 121
Min-cut = max-flow, 240
Minimal cover, 245
Minimum Cost Flow Problem, 242
Minkowski, H., 36
MINOS, 166
Minty, G., 31, 201
Mitchell, J., 121, 143
Mizuno, S., 142–144
Monma, C., 120, 142
Monte Carlo methods, 355
Monteiro, R., 141–143
Moore, E., 244
Morton, A., 120
Morton, D., 246, 361

-
- Motzkin's Transposition Theorem, 54
 - Motzkin, T., 59
 - Multi-commodity flow, 245, 317
 - Mulvey, J., 143, 144, 244
 - Munksgaard, N., 144
 - Muramatsu, M., 142
 - Murray, W., 31, 140, 144–146, 166, 167
 - Murtagh, B., 166
 - Murty, K., 201, 227

 - Naive sampling, 350
 - Nakayama, M., 361
 - Nazareth, L., 144, 166, 317
 - Neighbor, 18
 - Nemirovskii, A., 120
 - Nested decomposition, 309, 316
 - Network, 221
 - Network Simplex Method, 243
 - cycling, 244
 - efficiency, 244
 - pathological example, 244
 - Newton direction, 125
 - Newton's Method, 123–127, 201
 - Nishiya, T., 316
 - Node, 231
 - Node-arc incidence matrix, 232
 - Nondegenerate dual, 176
 - Nonnegative linear combination, 2
 - nonzero coefficient density, 79
 - Norm
 - ∞ , xxiv
 - one, xxiv
 - two, xxiv
 - Normal distribution, 370
 - NP, 69
 - Null space, 104

 - Oliveira, G., 332, 360
 - Open interval, 376
 - Open set, 38
 - Optimality cut, 302
 - Orchard-Hays, W., 201
 - Orden, A., 166, 169, 219
 - Orlin, J., 243, 244

 - Ortega, J., 140
 - Osborne, M., 167

 - Pallottino, S., 244
 - Papadimitriou, C., 120, 201
 - Pape, U., 244
 - Parametric objective, 201
 - Parametric programming, 183
 - worst-case behavior, 201
 - Pereira, M., 332, 360
 - Perold, A., 316
 - PERT, 231, 244
 - Perturbation method, 167, 169
 - Phase I Least-Squares, 197–200
 - algorithm steps, 198–199
 - Pierce, G., 142
 - Pinto, L., 332, 360
 - Plassman, P., 144
 - Polygon, 9
 - Polyhedral set, 16
 - Polyhedron, 9
 - Polynomial Time
 - ϵ -Optimal, 70
 - Polynomial time, 69
 - Ponceleón, D., 144, 145
 - Positive Matrix, 39
 - Positive semi-definite, 86
 - Positive-definite, 86
 - Potential function, 107
 - Powell, M., 121
 - Preconditioned conjugate gradient
 - method, 144
 - Predictor-Corrector Method, 142, 143
 - Primal affine method, 133
 - Dikin's method, 133
 - Primal logarithmic barrier method, 131
 - Primal-Dual Algorithm, 191
 - Primal-dual logarithmic barrier
 - method, 134
 - Probability density, 368
 - joint, 371
 - Probability distribution, 368
 - Probability theory, 367–377

- Projection matrix, 128, 133
Projective algorithm, 101
Projective transformation, 101
Projective transformation inverse,
101
- QR factorization, 133
Quadratic, 86, 328
- Rabinowitz, P., 361
Radzik, T., 244
Random Choice Rule, 158, 167
Random sample, 371
Random variable, 368
 continuous, 370
 discrete, 368
- Ray, 6
Rays, 5
Reduced array, 211
Redundant constraints, 16
Reid, J., 144
Reiter, S., 227, 228
Renegar, J., 140, 141
Resende, M., 120, 142
Resolution Theorem, 281
Restricted linear program, 155
Restricted Master, 286
Rheinboldt, W., 140
Ribi re, G., 166
Rockafellar, R., 31, 166, 332, 360
Roos, C., 141
Rosen, J., 264
Ruszczynski, A., 332, 360
Ryan, D., 167
- Saaty, T., 166, 201
Saigal, R., 141, 142, 264
Sakarovitch, M., 264
Saltzman, M., 142, 146
Sample mean, 371
Sample point, 367
Sample space, 367
 discrete, 368
Sample variance, 371
 unbiased, 372
- Sampling
 importance, 350
 naive, 350
Saturated arc, 240
Saunders, M., 31, 140, 144, 145, 166,
167
Scarf, H., 332
Schrage, L., 264
Schrijver, A., 120
Scott, D., 361
Self-Dual Parametric Algorithm, 188
 worst-case behavior, 201
Sen, S., 332, 360
Separable quadratic function, 83
Separating hyperplane, 11
Shanno, D., 120, 140–143, 145, 146
Shannon, C., 243
Sherali, H., 38, 244
Shor, N., 68, 120
Shortest path, 244
Shortest route, 244
Simplex, 102
 center, 102
 extreme point, 14
 radius of circumscribed sphere,
102
 radius of inscribed sphere, 102
 vertex, 14
Simplex Algorithm related to von
 Neumann Algorithm, 83
Simplex defined, 13
Simplex Interpretation
 Simplex Method, 24–30
Simplex Method
 simplex interpretation, 24–30
 steepest descent, 20–24
 versus interior-point methods,
69
 worst-case complexity, 201
Simplicial cone, 10
Sleator, D., 244
Smale, S., 31, 201
Small, R., 316
Sokolowsky, D., 119
Sonnevend, G., 141

- Staircase, 266, 308
- Standard deviation, 369
- Standard form miscellaneous
theorems, 48
- Statistic, 371
- Steepest descent, 20–24
along edges, 24
- Steepest descent direction, 70, 100,
103, 104, 124
- Steiglitz, K., 120
- Steinberg, D., 166
- Steinitz, E., 37
- Stieglitz, K., 201
- Stiemke's theorem for homogeneous
systems, 53
- Stiemke, E., 59
- Stone, J., 361
- Stoyan, D., 361
- Strict complementary slackness, 56
- Strong Duality Theorem, 45
first known proof, 59
first rigorous proof, 59
originator, 59
- Student's t -distribution, 375
- Support, 339
- Supporting hyperplane, 11
- Suurballe, J., 166
- System of distinct representatives,
249
- Tapia, R., 143
- Tarjan, R., 244
- Thapa, M., 144
- Theorems of the alternatives, 43, 50,
59
- Thomen, D., 264
- Todd, M., 121, 143, 144, 264
- Tomlin, J., 121, 140
- Tompkins, C., 119
- Transportation problem, 207–229
basis triangularity, 210
bounded partial sums of
variables, 225
classical, 207–213
compact representatoin, 208
mathematical statement, 208
properties, 208
rank, 208
degeneracy, 216
Fundamental Theorem, 211
improved solution, 214
initial solution, 213
integer solution property, 212
simplex multipliers and reduced
costs, 214
Triangularity Rule, 213
- Transshipment problem, 219
reduction to classical form, 222
reduction to classical form by
transshipment, 222
- Triangular matrix, 210
- Triangularity Rule, 213
- Tsuchiya, T., 141, 142
- Tucker's example, 150
- Tucker's strict complementary
slackness theorem, 56
- Tucker, A., 31, 59, 146, 150, 166,
169, 315, 316
- Ülkücü, A., 316
- Unbiased estimator, 371
- Unbiased sample variance, 372
- Unboundedness Theorem, 47
- Uncertain costs, 326
- Uncertain demand, 329
convex costs, 331
- Unimodular, 212
total, 212
- Upper-blocked, 238
- Using t in S , 93
- van de Panne, C., 202
- van der Vorst, H., 144
- Van Slyke, 332, 361
- Van Slyke, R., 264, 332, 360
- Vanderbei, R., 120, 141, 144–146
- Variable cost
minimum expected cost, 326
minimum variance, 327
- Variance, 368

- Variants of the Simplex Method, 173
Vavasis, S., 144
Vector, 1
Veiga, G., 120, 142
Vertex, 14, 15
Vial, J. P., 121
Vial, J.-Ph., 121
Ville's theorem of alternatives for
 matrices, 55
Ville, J., 59
von Neumann
 skew-symmetric, 43
von Neumann Algorithm, 73
 as a Variant of the Simplex
 Algorithm, 83–84
 Improving the Rate of
 convergence, 81
 steps, 81
von Neumann symmetric form, 43
von Neumann versus Karmarkar
 Algorithms, 118
von Neumann's method, 70–84
von Neumann, J., 58, 67, 70, 119, 145
- Wallace, S., 332, 360, 361
Wang, Y., 142
Watteyne, P., 121
Weak Duality Theorem, 44
Weak law of large numbers
 (Chebyshev), 372
Wets, R., 332, 360, 361
Whinston, A., 202
Whiting, P., 244
Wiegman, N., 119
Willard, D., 244
Wittrock, R., 361
Wolfe's Generalized Program, 267
Wolfe's Rule, 156–158
Wolfe, P., 36, 146, 154, 166, 169, 267,
 315, 316
Wright, M., 31, 120, 140, 144, 146,
 166, 167
Wright, S., 143
- Yamashita, H., 121
Ye, Y., 120, 143, 144, 146
Ye, Yinyu, 40
Yoshise, A., 142, 143
- Zadeh, N., 244
Zangwill, W., 146
Zhang, D., 143
Zhang, Y., 143
Zhu, J., 146
Ziemba, W., 361