MATHEMATICS
IN SCIENCE
AND
ENGINEERING

Volume 137

Integer and Mixed Programming: Theory and Applications

Arnold Kaufmann
Arnaud Henry-Labordère

INTEGER AND MIXED PROGRAMMING

Theory and Applications

This is Volume 137 in
MATHEMATICS IN SCIENCE AND ENGINEERING
A Series of Monographs and Textbooks
Edited by RICHARD BELLMAN, *University of Southern California*

The complete listing of books in this series is available from the Publisher upon request.

Integer and Mixed Programming

THEORY AND APPLICATIONS

Arnold Kaufmann

Université de Louvain
Belgium

Arnaud Henry-Labordère

L'École Nationale des Ponts et Chaussées
Paris, France

Translated by Henry C. Sneyd

ACADEMIC PRESS New York San Francisco London 1977

A Subsidiary of Harcourt Brace Jovanovich, Publishers

ACADEMIC PRESS, INC.
111 Fifth Avenue, New York, New York 10003

United Kingdom Edition published by
ACADEMIC PRESS, INC. (LONDON) LTD.
24/28 Oval Road, London NW1

Library of Congress Cataloging in Publication Data

Kaufmann, Arnold.
Integer and mixed programming.

(Mathematics in science and engineering)
Translation of v. 3 of Méthodes et modèles de la recherche opérationnelle.
Bibliography: p.
1. Integer programming. I. Henry-Labordère, A. II. Title. III. Series.
T57.7.K3913 519.7'7 75-32029
ISBN 0-12-402365-7

PRINTED IN THE UNITED STATES OF AMERICA

Originally published in the French language under the title, "Méthodes et Modèles de la Recherche Opérationnelle," Tome III in 1974 by Dunod, Paris.

CONTENTS

Part 1. METHODS AND MODELS

Chapter I. PROGRAMS WITH INTEGER AND MIXED VALUES

Part 2. MATHEMATICAL THEORY

Chapter II. ALGORITHMS AND HEURISTICS FOR INTEGER OR MIXED PROGRAMS

PREFACE

In this third volume of Methods and Models of Operations Research our loyal readers will discover that the same organization has been adopted as in the first two volumes: a first part in which mathematics is subordinated to the practical aspects of the concepts to be studied and a second part devoted to the mathematical side of the various problems. This method of presentation in the earlier volumes has been widely welcomed, as shown by the numerous editions and by translations into a variety of languages.

This volume deals with integer programs and programs with mixed values and will complete a small library for engineers and specialist groups. Operations research is now a part of their equipment, but advances in this field take place every year and it is necessary that they should become acquainted with them.

For the present volume I have had the collaboration of my friend A. Henry-Labordère, Engineer in Arts and Manufacturing, Master of Science, and Ph.D. He is an engineer with a wide reputation in operations research, an advisor to a very important firm of European consultants, and has also taught mathematical programming at l'École Centrale des Arts et Manufactures in Paris for several years.[1] The latter experience has assisted him in presenting numerous sections of this work in an instructional form. We have shared the production between us, with the author of the previous volumes retaining the responsibility for its coordination.

Integer programming is a subject that is of ever-increasing interest to engineers, economists, and informaticians since problems with integer solutions occur in every field of science and technological research. Such problems are, as a rule, appreciably more difficult to solve than those with continuous

[1] At present, Dr. Henry-Labordère is teaching at l'École Nationale des Ponts et Chaussées in Paris.

and linear values. For the latter, the storehouse is well stocked with algorithms, but the same does not yet apply for problems with integer values, although considerable progress has been made, especially during the past five years. The reason for this lies in the fact that diophantine mathematics contains combinatorial difficulties that do not occur with continuous values. This is a situation that cannot be altered, but considerable progress has nevertheless been made and some essential results are now available.

As all mathematicians concerned are aware, the subject of this volume is a mathematically difficult one, but we have endeavoured to balance the strictness of the theory with the instructional needs of our readers. Among the more useful methods of procedure are some very difficult algorithms such as Gomory's asymptotic algorithm as well as the methods of Benders and Trubin. These have been grouped in a Supplement, but they have still been given the same instructional presentation.

The largest category of programs and the one involving the greatest difficulties, that of nonlinear programs, will be treated in a fourth volume now in preparation. I have again asked A. Henry-Labordère to be my collaborator, while we have been joined by my friend and former pupil at Grenoble, M. D. Coster, who is currently a consultant in informatics and operations research. During recent years he has acquired a wide knowledge of these nonlinear problems.

Returning to the present volume, I would like to outline my attitude toward the publication of new material in the series MMOR (Methods and Models of Operations Research) as they are now known by a wide circle of engineers. Instead of bringing the volumes up to date with each new edition I have preferred to leave them as published and to publish the new material every five or six years in fresh works that will not render the earlier ones obsolete.

In writing these MMOR volumes we have often recalled one of the rules of St. Benoît: "Encourage the strong without discouraging the weak." By means of this precept each student can progress according to the individual's mental speed and available resources. What is needed is to progress, slowly and surely or quickly and dangerously, according to one's wishes and ability, as long as progress is made. It is not in human nature not to advance or to attempt, since this is reserved for the negligent and the idle, for those who do not wish to confer any benefit on their fellows but merely to live for themselves. The latter are those to whom I scathingly referred in one of my books[1] as "subhumans," and this is the lot of far too many who refuse to realize that self-improvement at all levels is the object of existence.

The conquest of knowledge and of mental, moral, and emotional equilibrium is the basic adventure of our species; and if in this respect it has

[1] A. Kaufmann and J. Pezé, "Des sous-hommes et des super-machines," Albin Michel.

something of the tortoise and the hare, its only real goal is that of self-mastery.

I wish to thank our friends: Hervé Thiriez, Professor at the Centre d'Enseignement Supérieur des Affaires at Jouy-en-Sosas, and Michel Gondran, Research Engineer with Électricité de France, who have taken meticulous care in rereading and finalizing the manuscript. We are additionally indebted to them for a number of constructive suggestions about the models and the proofs.

My son Alain has also had an important part in checking the manuscript and the proofs.

Finally, we wish to thank the editor and his collaborators for their usual care in the publication of this series, as well as the Director of the Collection, Professor Ad. André-Brunet who has always given me his sincere encouragement and support.

L'Institut National Polytechnique A. KAUFMANN
Grenoble, France

Part 1. METHODS AND MODELS

Chapter I. PROGRAMS WITH INTEGER AND MIXED VALUES

Section 1. Introduction

In this chapter we shall consider such practical problems as can be expressed in the form of mathematical programs, which are similar to those of linear programming as discussed in the first volume,[1] except for the requirement that the variables must be integers such as 0, 1, 2, 3, ... The reader will already have been convinced as to the practical importance of problems defined by linear programs. In operations research and econometrics we are often aware that the choices are discrete, in other words, that they can only assume definite and not closely contiguous values, that this or that has to be done, a factory has to be built or not built. Consequently, for practical purposes, problems of linear programming with integer solutions are of an even greater importance than the classic problems of linear programming. We shall see that choices for investment and problems for the engineer and even for the plumber can be expressed in this form.

It may well be asked, therefore, why the interest in programming with integers is so recent, dating from some fifteen years only, if it can be so widely applied. Paradoxically, discrete mathematics, which originated with the arithmetic of the Greeks and Arabs, has over recent centuries occupied the position of a poor relation in the field of research. From many points of the scientific spectrum, logic, algebra, operations research, information, humane sciences, and the arts, interest in them has awakened to such a degree that at a

[1] *Note to Reader*: Throughout the present work, Volume 1 refers to A. Kaufmann, "Methods and Models of Operations Research," Prentice-Hall, Englewood Cliffs, New Jersey, 1963.

recent congress of pure mathematics more than half the discussion was devoted to the subject of discrete mathematics. It is but recently that effective methods have been discovered for solving such problems; easy to formulate, they possess the disadvantage of extensive calculations, containing, as they do, numerous variables and constraints.

In this chapter we shall give practical cases that can be expressed as problems with integer variables. Brief statements about the main properties will be given, and methods will be outlined. In the second part of this work the reader will, as usual, find the requisite theoretical analyses. In particular, he will find those dealing with the problems of programs with mixed numbers in which some variables must be integers and others may be continuous, as in classic linear programming. We shall observe that the latter type of problem is specially important.

Section 2. **Some Examples of Problems with Integer Solutions**

1. Characteristics of Problems with Integer Solutions

Let us consider the set **S** containing the solutions of a linear program and let $[x] = [x_1, x_2, \dots, x_n]$ be one of the solutions belonging to **S**. If we now impose the constraint that the components of $[x]$ must be natural numbers (integer and nonnegative) we can state that $[x]$ is an integer solution. Thus, in a case where $n = 5$,

$$[x] = [x_1, x_2, x_3, x_4, x_5] = [3, 0, 1, 9, 0]\,, \tag{2.1}$$

$[x]$ will be an integer solution. This will not be the case for

$$[x] = [x_1, x_2, x_3, x_4, x_5] = [3\ ,1.08, 0, 5.7, 1]\,, \tag{2.2}$$

nor for

$$[x] = [-1, 0, -3, 2, 9] \tag{2.3}$$

and

$$[x] = [6, 1, 9/2, 2/3, 0]\,. \tag{2.4}$$

Let us examine a simple example of linear programming of which we will temporarily ignore the economic function to be optimized.

Let

$$\begin{aligned} 6x_1 + 9x_2 &\leqslant 54\,, \\ 7x_1 + 6x_2 &\leqslant 42\,, \\ x_1 &\leqslant 4\,, \\ x_1 &\geqslant 0\,, \\ x_2 &\geqslant 0\,. \end{aligned} \tag{2.5}$$

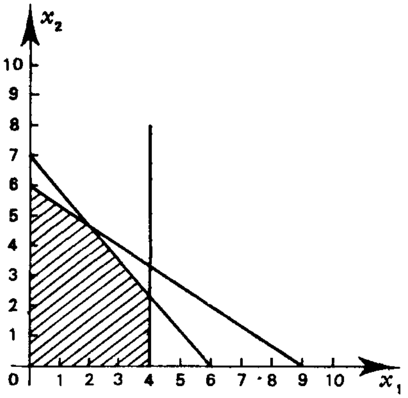


FIG. 2.1

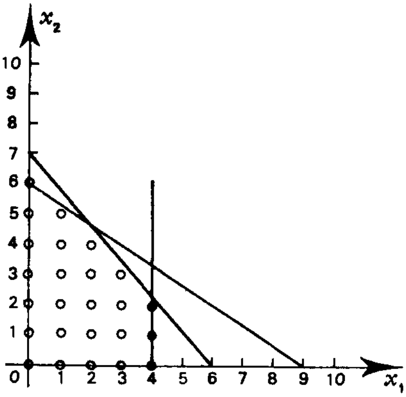


FIG. 2.2

The set **S** of the solutions of (2.5) is represented by the hachured area of Fig. 2.1. Let us now introduce the constraint of only accepting as solutions those of which the components x_1 and x_2 are nonnegative integers: the set Σ of the corresponding solutions is represented in Fig. 2.2.

This subset Σ of **S** consists of

(2.6) $$\Sigma = \{[0, 0], [0, 1], [0, 2], [0, 3], [0, 4], [0, 5], [0, 6], [1, 0], [1, 1], [1, 2], [1, 3], [1, 4], [1, 5], [2, 0], [2, 1], [2, 2], [2, 3], [2, 4], [3, 0], [3, 1], [3, 2], [3, 3], [4, 0], [4, 1], [4, 2]\}\,.$$

Here the number of integer solutions is finite; in other cases it might be infinite.

Let us now suppose that the economic function of the linear program (2.5) is

(2.7) $$[\text{MAX}]z = 7x_1 + 5x_2\,.$$

From Fig. 2.3 it can be seen that the maximal solution of the linear program (2.5), (2.7) is

(2.8) $$[x_1, x_2] = [4, 7/3]\,.$$

This is not an integer solution, but let us nevertheless calculate the corresponding value of z:

(2.9) $$\begin{aligned} z &= (7).(4)+(5).(7/3) \\ &= 39\tfrac{2}{3} = 39.66\ldots \end{aligned}$$

Let us now impose the constraint that the solution of this program is to be integer. With the very simple problem that we are considering, it is sufficient to determine which will be the first point (or points) representing an integer value encountered after entering the polygon of solutions when the straight

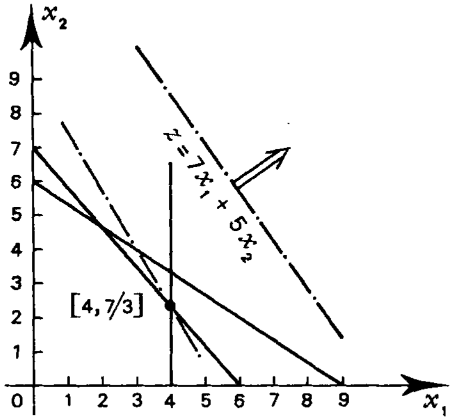


FIG. 2.3

line $7x_1 + 5x_2 = z$, has undergone a parallel displacement. It can be seen by inspecting Figs. 2.2 and 2.3 that this point will be

(2.10) $$[x_1, x_2] = [4, 2]\,,$$

for which we have

(2.11) $$z = (7).(4) + (5).(2) = 38\,.$$

The next point with integer values that we encounter is

(2.12) $$[x_1, x_2] = [3, 3]\,,$$

and for this we obtain

(2.13) $$z = (7).(3) + (5).(3) = 36\,.$$

It is advisable to clarify at once for the reader that the maximal solution with integer values is not always obtained by taking the maximal solution of the program for continuous values and by then suppressing the decimal portion of it. In this context, the reader should study the linear program represented in Fig. 2.4. The maximal solution of this program is [2.8; 4.3] and the maximal solution for integer values is not [2.4] or [3.4] but [3.3], as can be verified by sliding the straight line representing the function z parallel to itself. The same remark applies when we consider a minimal solution with integer values. This is not always obtained from the minimal solution for the corresponding program with continuous values. For example, if [3.17; 2.92] is the minimal solution of a given program, it is perfectly possible that neither [3.3] nor [4.3] is a minimal solution for integer values.

In addition, when the number of variables in the program exceeds two, it may prove very difficult to determine the solutions with integer values without enumerating and verifying all the solutions by means of the constraints. Such a process, useful as it may be for certain particular cases, is not generally

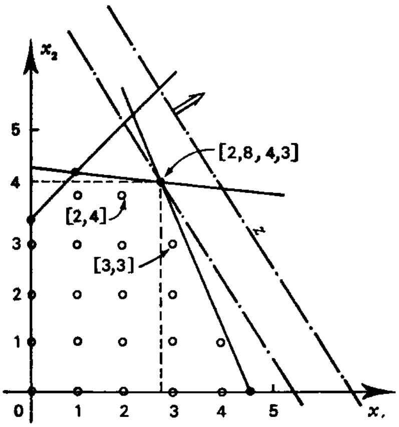


FIG. 2.4

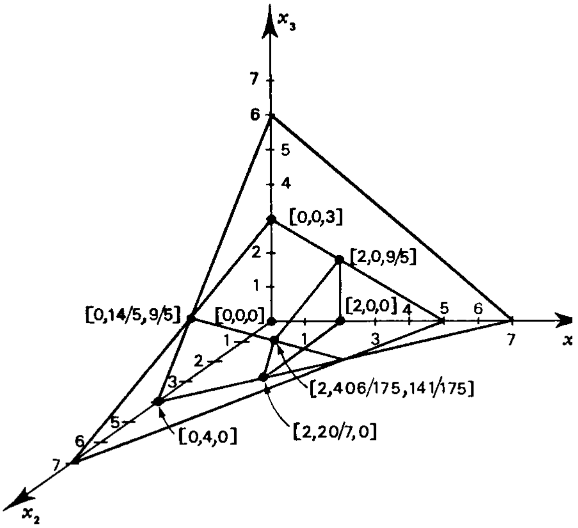


FIG. 2.5

practical because of the large number of integer solutions to be considered. Even in a program with three variables and three constraints (Fig. 2.5),

$$\begin{aligned} &\frac{x_1}{7} + \frac{x_2}{4} + \frac{x_3}{6} \leqslant 1, \\ (2.14) \qquad &\frac{x_1}{5} + \frac{x_2}{7} + \frac{x_3}{3} \leqslant 1, \\ &x_1 \leqslant 2, \\ &\qquad x_1 \geqslant 0, \qquad x_2 \geqslant 0, \qquad x_3 \geqslant 0, \end{aligned}$$

it is by no means easy to discover the integer solutions; to obtain the set that contains them, it is necessary to verify some thirty points.

Except for very simple problems, we are therefore obliged to make use of special algorithms for programs with integer values. The various principles underlying them will be very briefly discussed in the present chapter, and their fuller explanation and proofs will be given in the second part.

Let us, however, first consider some very simple examples.

2. Some Preliminary Examples

A Problem Dealing with the Transportation of School Children[1]

In a village A there is a school attended by some hundred children, 72 of whom live a certain distance away, whence the need to arrange their trans-

[1] This problem is given by Mlle. Edith Heurgon in her thesis, "Programming with integer numbers. Arborescent method of Robert Faure and Yves Malgrange." Faculté des Sciences de Paris, 1967. We have slightly modified the terms to satisfy the requirements of the present work.

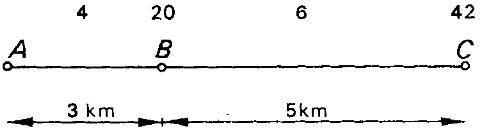


FIG. 2.6

portation by bus. There are two main collection points B and C (B being situated between A and C) (Fig. 2.6). The number of pupils to be collected is as follows: 42 at C, six between C and B, 20 at B, and four between B and A. The firm that can provide the transport owns two types of bus: one with 35 seats and another with 50 seats. The prices charged by the firm are as follows for each journey and for each kind of bus:

	Type of Bus	
	35 seats	*50 seats*
BA	39 F	50.50 F
CA	54 F	68 F
CB	45 F	57.50 F.

We must not be surprised that the proposed charges are not proportional to the distances, since the fixed costs of such an operation generally exceed the variable ones.

The problem is to decide which type of bus should be used on each of the sections in order to minimize the total outlay.

Let us use the following symbols for the variables representing the number of buses to be considered in each case:

	Buses	
	35 seats	*50 seats*
BA	x	x'
CA	y	y'
CB	z	z'

The linear program with integer numbers is easily obtained:

$$\text{(2.15)}\quad \begin{aligned} &[\text{MIN}]f = 39x + 54y + 45z + 50.5x' + 68y' + 57.5z', \\ &35y + 35z + 50y' + 50z' \geqslant 48\,, \\ &35x + 35y + 50x' + 50y' \geqslant 72\,, \\ &x \geqslant 0, \quad y \geqslant 0, \quad z \geqslant 0, \quad x' \geqslant 0, \quad y' \geqslant 0, \quad z' = 0\,. \end{aligned}$$

The first line of the program (2.15) expresses the economic function, the total cost. The second line represents the constraint imposed by the different possibilities that the buses must provide when they start their collection of pupils at C, bring them to B and finally to A. The third line represents the buses that finish at A.

Resolved into continuous variables, the linear program (2.15) provides an optimal solution of

$$\text{(2.16)} \qquad x=0,\quad y=0,\quad z=0,\quad x'=12/25,\quad y'=24/25,\quad z'=0,\quad \min f=89.52.$$

Resolved into integer variables by means of one of the algorithms described in the second part, or by enumeration (which is easy in this case), we then obtain as the minimal solution

$$\text{(2.17)} \qquad x=1,\quad y=0,\quad z=0,\quad x'=0,\quad y'=1,\quad z'=0,\quad \min f=107\,.$$

It will be observed that this solution cannot be obtained by rounding off the solution of (2.16) to the integer immediately below or above it.

The Problem of the Knapsack. A Problem of Investment

A hiker wishes to carry a certain number of articles $X_1, X_2, \ldots, X_n$ in his knapsack. He knows the weight $P_1, P_2, \ldots, P_n$ of each of the articles, as well as their respective volumes[1] $V_1, V_2, \ldots, V_n$. He is unable to carry a total load in excess of P, and his knapsack cannot contain a volume greater than V. The hiker allots values $k_1, k_2, \ldots, k_n$ to each of the articles according to its intrinsic utility. Which objects should he take with him to maximize their total utility?

This problem will be represented by the following linear program with integer values, in which x_1 is the number of the articles X_1 to be carried:

$$\text{(2.18)} \qquad \begin{aligned} &[\text{MAX}]\, z = k_1x_1 + k_2x_2 + \ldots + k_nx_n\,,\\ &P_1x_1 + P_2x_2 + \ldots + P_nx_n \leqslant P\,,\\ &V_1x_1 + V_2x_2 + \ldots + V_nx_n \leqslant V\,,\\ &x_1 \geqslant 0,\quad x_2 \geqslant 0, \ldots, x_n \geqslant 0\,. \end{aligned}$$

A variation of this problem plays an interesting part in a number of algorithms. Let us suppose that our aim is to maximize V and to take P as a constraint (which would not make much sense for the bearer of the knapsack, but makes sense for other concepts). We should then write

$$\text{(2.19)} \qquad \begin{aligned} &[\text{MAX}]\, V = V_1x_1 + V_2x_2 + \ldots + V_nx_n\,,\\ &P_1x_1 + P_2x_2 + \ldots + P_nx_n \leqslant P\,,\\ &x_1 \geqslant 0,\quad x_2 \geqslant 0, \ldots, x_n \geqslant 0\,. \end{aligned}$$

A concrete and practical problem can be envisaged in the form of (2.19).

[1] It would be strictly more fitting to speak of cumbersomeness rather than of volume. The introduction of volumes (unless the articles are soft ones) is clearly open to criticism, and we must ask indulgence for the somewhat theoretical nature of the term.

A capital sum K is available and can be used to construct units of production in different localities L_1, L_2, L_3, and L_4, the installation costs C_1, C_2, C_3, and C_4 varying according to the locality selected. Let us use B_1, B_2, B_3, and B_4 to represent the unit profits derived from investments in the corresponding localities. The problem is which localities to choose and how many units of production to build in each of them in order to maximize the total profit.

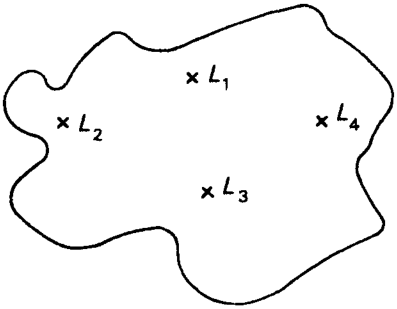


FIG. 2.7

Taking as variables x_1, x_2, x_3, x_4 to represent the number of units to be built in the various localities, we obtain as a model one in all respects similar to (2.19).

$$\begin{aligned} &[\mathrm{MAX}]z = B_1 x_1 + B_2 x_2 + B_3 x_3 + B_4 x_4\,, \\ (2.20)\qquad &C_1 x_1 + C_2 x_2 + C_3 x_3 + C_4 x_4 \leqslant K\,, \\ &x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad x_3 \geqslant 0, \quad x_4 \geqslant 0\,. \end{aligned}$$

The reader will have learned in Volume 2[1] (Section 12, page 86) how to resolve this problem by means of dynamic programming. Some problems with integer values can, indeed, be resolved by this method, but, in cases where there are a greater number of constraints, the method cannot easily be employed and may even have to be discarded from the outset, since the problem cannot be reduced, after it has been transformed, into a sequential form.

3. Another Well-Known Problem

In Volume 1 (page 64) and in Volume 2 (page 265) we gave a problem known in mathematical parlance as a *problem of assignment* but which is equally a

[1] *Note to Reader*: Throughout the present work, Volume 2 refers to A. Kaufmann, "Graphs, Dynamic Programming, and Finite Games," Academic Press, New York, 1967.

linear program with integer values. Here these values are bivalent; that is to say that, in such problems, they can only assume the values of 0 or 1.

Let us recall this problem.[1] We have to consider n workmen $X_1, X_2, \ldots, X_n$ and n positions of employment $Y_1, Y_2, \ldots, Y_n$. To each assignment (X_i, Y_j) a cost is attached (Fig. 2.8):

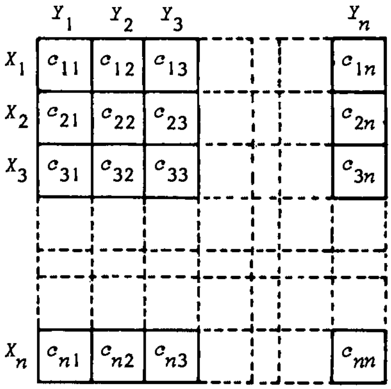
$$c_{ij} \geqslant 0, \qquad i, j = 1, 2, \ldots, n. \tag{2.21}$$

Some of the c_{ij} may be infinite (which means that the corresponding assignment is impossible).

We are required to assign the n workmen to n positions in such a manner that each workman will have one and only one position and that the total cost of the assignments will be minimal. This gives the following program:

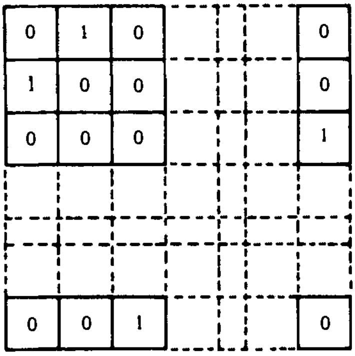
$$\begin{aligned} &[\text{MIN}]\ z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}, \\ &\sum_{i=1}^{n} x_{ij} = 1, \qquad j = 1, 2, \ldots, n, \\ &\sum_{j=1}^{n} x_{ij} = 1, \qquad i = 1, 2, \ldots, n, \\ &x_{ij}^2 = x_{ij}, \qquad i, j = 1, 2, \ldots, n. \end{aligned} \tag{2.22}$$

The relation $x_{ij}^2 = x_{ij}$ imposes the constraint on each variable x_{ij} that it cannot be equal to a number other than 0 or 1. An assignment is represented by a table (Fig. 2.9) containing a single and only a single 1 in each line and also in each column. Various special methods exist for the solution of such problems, as can be discovered from our references [K74]–[K76].



	Y_1	Y_2	Y_3				Y_n
X_1	c_{11}	c_{12}	c_{13}				c_{1n}
X_2	c_{21}	c_{22}	c_{23}				c_{2n}
X_3	c_{31}	c_{32}	c_{33}				c_{3n}
X_n	c_{n1}	c_{n2}	c_{n3}				c_{nn}

FIG. 2.8



0	1	0				0
1	0	0				0
0	0	0				1
0	0	1				0

FIG. 2.9

[1] This problem is given by M. R. de Grove, Revue Française de Recherche Opérationnelle, No. 39, pp. 171–183.

Section 3. Boole's Binary Algebra

1. The Binary States of a System

Situations in which decisions appear as alternatives are the most frequent among those by which each of us is confronted. Indeed, it can be shown that every decision, in a system in which the number of states is finite, can be reduced to alternatives in a more or less complex set of variables. Boole's binary algebra enables us to deal with problems of this kind; it has come to play a fundamental[1] part in all the sciences and especially in operations research, information theory, and language theory. In the course of the preceding volumes the reader has been provided with such information about this algebra as was required in the context. We now propose to enter more deeply into this subject.

The concept of an alternative is, therefore, one of the most frequent with which we have to deal in our reasoning and in our actions:

all or nothing,
red or white,
open or closed,
exists or does not exist,
0 or 1,
true or false,
dead or alive, and so on.

To be sure, these alternatives correspond to models that we use, usually for convenience, in our reasoning.[2] In nature, it is not only black or white but colors, which are to be found, but to convey these colors we nevertheless make use of directions of all or nothing (color television), which can be reduced to such alternatives. A tap may be half-closed (or half-open, if one prefers); by means of an appropriate binary symbolism we can exactly describe the three states: open, half-closed, and closed. There may be situations in which we cannot state whether a thing is true or false, and we then add a third situation (and eventually others). Dead or alive may not be accurately observed or determined (since certain types of coma may be variously interpreted). But

[1] Because a number of problems derived from the humane sciences cannot, or can only with great difficulty, be reduced to a logical treatment by Boole's algebra, there is a growing interest in methods which permit the introduction of shades, of fuzzying of propositions and relations. Such is the aim of the theory of fuzzy sets enunciated by L. A. Zadeh. Note, as an example, A. Kaufmann, "Introduction to the Theory of Fuzzy Subsets," Volume 1, Academic Press, New York, 1975.

[2] It is interesting to note that an inverse tendency is now appearing.

we shall be more disposed to classify the finite states with the aid of more or less complex binary concepts, rather than try to determine whether the variables of the system that we have examined are intrinsically bivalent or not. We know that the concept of a variable, whether it is in a phenomenon of nature or in a phenomenon of organization, is an arbitrary one; it is part of the model that our imagination has constructed.

Let us, accordingly, study a system **S** containing a single element or component S_1, which is free to assume two and only two states E_1 and $\bar{E}_1$. We shall then introduce a variable x_1:

$$\begin{aligned} x_1 &= 1 \quad \text{if } S_1 \text{ is in the state } E_1, \\ &= 0 \quad \text{if } S_1 \text{ is in the state } \bar{E}_1. \end{aligned} \tag{3.1}$$

The choice of the values 1 and 0 for x_1 to represent the state of the system is arbitrary; the variable could equally well be defined in the following manner:

$$\begin{aligned} x_1 &= 0 \quad \text{if } S_1 \text{ is in the state } E_1, \\ &= 1 \quad \text{if } S_1 \text{ is in the state } \bar{E}_1. \end{aligned} \tag{3.2}$$

Let us now consider a system **S** having two elements or components S_1 and S_2, and let the two possible states of S_1 be represented by E_1 and $\bar{E}_1$, and those of S_2, in like manner, by E_2 and $\bar{E}_2$. We now introduce the variables x_1 and x_2, such that

$$\begin{aligned} x_1 &= 1 \quad \text{if } S_1 \text{ is in the state } E_1, \\ &= 0 \quad \text{if } S_1 \text{ is in the state } \bar{E}_1, \\ x_2 &= 1 \quad \text{if } S_2 \text{ is in the state } E_2, \\ &= 0 \quad \text{if } S_2 \text{ is in the state } \bar{E}_2. \end{aligned} \tag{3.3}$$

However, it is possible to employ a more general representation:

$$\begin{aligned} [x_1, x_2] &= [1, 1] \quad \text{if } \mathbf{S} \text{ is in the state } [E_1, E_2], \\ &= [1, 0] \quad \text{if } \mathbf{S} \text{ is in the state } [E_1, \bar{E}_2], \\ &= [0, 1] \quad \text{if } \mathbf{S} \text{ is in the state } [\bar{E}_1, E_2], \\ &= [0, 0] \quad \text{if } \mathbf{S} \text{ is in the state } [\bar{E}_1, \bar{E}_2]. \end{aligned} \tag{3.4}$$

And in a more general manner, if **S** contains n elements or components $S_1, S_2, \ldots, S_n$, able, respectively, to assume the states E_1 or $\bar{E}_1$, E_2 or

$\bar{E}_2, \ldots, E_n$ or $\bar{E}_n$, we can introduce the following values:

(3.5)

$$\begin{aligned}
[x_1, x_2, \ldots, x_n] &= [1, 1, \ldots, 1], \quad \text{if } \mathbf{S} \text{ is in the state } [E_1, E_2, \ldots, E_n],\\
&= [1, 1, \ldots, 0], \quad \text{if } \mathbf{S} \text{ is in the state } [E_1, E_2, \ldots, \bar{E}_n],\\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\\
&= [1, 0, \ldots, 0], \quad \text{if } \mathbf{S} \text{ is in the state } [E_1, \bar{E}_2, \ldots, \bar{E}_n],\\
&= [0, 0, \ldots, 0], \quad \text{if } \mathbf{S} \text{ is in the state } [\bar{E}_1, \bar{E}_2, \ldots, \bar{E}_n].
\end{aligned}$$

It is always possible to use binary variables to represent states with a system **S** of which the elements, finite in number, can assume a larger number of states than two. Thus, let us consider a system **S** containing a single element S_1 capable of assuming states A, B, and C. We can arbitrarily define A, B, C by a variable composed of several binary variables and write

$$\begin{aligned}
[x_1^{(1)}, x_1^{(2)}] &= [1, 1] \quad \text{if } S_1 \text{ is in state } A,\\
&= [1, 0] \quad \text{if } S_1 \text{ is in state } B,\\
&= [0, 1] \quad \text{if } S_1 \text{ is in state } C,\\
&= [0, 0] \quad \text{impossible.}
\end{aligned} \tag{3.6}$$

Or again, to avoid various difficulties of logic, we can write

$$\begin{aligned}
[x_1^{(1)}, x_1^{(2)}] &= [1, 1] \quad \text{if } S_1 \text{ is in state } A,\\
&= [1, 0] \quad \text{if } S_1 \text{ is in state } B,\\
&\left.\begin{aligned} &= [0, 1]\\ &= [0, 0]\end{aligned}\right\} \text{if } S_1 \text{ is in state } C.
\end{aligned} \tag{3.7}$$

In the latter case, two values of $(x_1^{(1)}, x_1^{(2)})$ instead of one represent a state of S_1, and this may be the cause of other difficulties. We shall, nevertheless, show that an integer can always be represented by a binary number.

2. Binary Enumeration

Let us remind ourselves of the significance of decimal expression; for example,

$$2518 = 2.10^3+5.10^2+1.10^1+8.10^0. \tag{3.8}$$

We know that it is possible to employ other bases than 10, for instance, 7:

$$305 = 6.7^2+1.7^1+4.7^0; \tag{3.9}$$

or also 12 (by associating two new digits with those of the base 10, for example,

$\alpha = 10, \beta = 11$):

$$3\alpha.92 = 3.12^3+\alpha.12^2+9.12^1+2.12^0. \tag{3.10}$$

The use of 2 as a base plays a fundamental role in the new mathematics and in numerous applications of them. Thus,

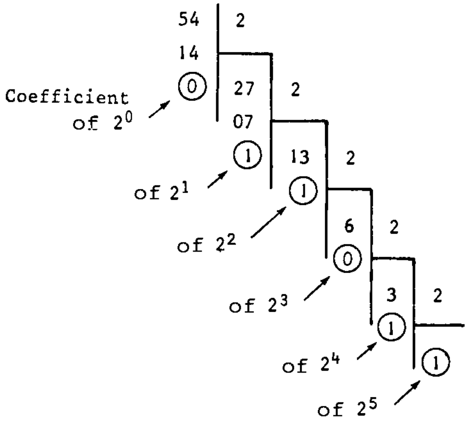
$$1\,1\,0\,1\,1\,0 = 1.2^5+1.2^4+0.2^3+1.2^2+1.2^1+0.2^0. \tag{3.11}$$

It is easy to convert a number given with 2 as a base and express it with base 10. All that is needed is to carry out the expansion. Thus,

$$\underset{\text{base 2}}{1\,1\,0\,1\,1\,0} = 32+16+0+4+2+0 = \underset{\text{base 10.}}{54} \tag{3.12}$$

To effect the inverse conversion, we consider the remainders of the divisions by 2 of the successive quotients, in the manner shown in the following example:

(3.13)



The four operations of common arithmetic can be used for numbers with 2 as a base. Thus, the calculation $1\,1\,0\,1+1\,0\,1\,1\,1\,1$ is expressed as follows:

$$\begin{array}{rr} 1\,1\,0\,1 & (13) \\ +\,1\,0\,1\,1\,1\,1 & +\,(47) \\ \hline 1\,1\,1\,1\,0\,0 & (60) \end{array} \tag{3.14}$$

A subtraction is performed in the following manner: for example, to calculate $1\,0\,0\,1\,1\,0\,1-1\,0\,1\,1\,0$,

$$\begin{array}{rr} 1\,0\,0\,1\,1\,0\,1 & (77) \\ -\quad 1\,0\,1\,1\,0 & -\,(22) \\ \hline 1\,1\,0\,1\,1\,1 & (55) \end{array} \tag{3.15}$$

Multiplication and division are rather more complicated, but these complications mainly arise from our methods of calculation acquired when using 10 as a base.

3. Operations with Bivalent Variables[1]

More than a hundred years ago George Boole (1815–1864) showed how logical propositions could be expressed in algebraic form. Hence, it is possible, thanks to Boole's algebra, to determine the truth or falsity of a proposition by a series of comparatively simple operations.

Let us suppose that we have two propositions connected by the conjunction *and*, with a and b, for example, representing the simple elements of which sodium chloride is composed, namely chloride *and* sodium. Neither chloride nor sodium alone is sufficient; both are needed. In like manner, a man *and* a woman (ignoring parthenogenesis) can have a child. The proposition *a and b* is expressed as $a.b$.

Proceeding further, we use the type of proposition *and/or* when the presence of a single component is sufficient to prove the truth of that proposition. For example, to settle an account we can use either a check or a postal order (indeed, both could be used at the same time, part of the account being settled by one method of payment, the remainder by the second method). A further example is heating by gas *and/or* electricity. In both these cases the proposition *a and/or b* is written as $a+b$.

On the other hand, we use the proposition of the type *or* when the presence of a component excludes that of the other. For instance, to write a letter, I use either a pen or a pencil (just try writing a letter with both!). This disjunctive *or* is written as $a \oplus b$.

Lastly, there is the negation or complementary proposition. If a represents the proposition, "man has set foot on the moon," $\bar{a}$ will represent the proposition, "man has not set foot on the moon." By means of the negation we can immediately verify that the disjunctive *or*, symbolized by $\oplus$, provides the equivalence: $a \oplus b = a.\bar{b}+\bar{a}.b$: a pen and no pencil and/or a pencil and no pen.[2]

Another means of demonstrating the practical use of Boolean algebra is to consider the connections that enable an electric current to pass or not pass through a circuit. In Figs. 3.1–3.3 different arrangements of electrical connections are shown. In Fig. 3.1 the current is free to pass if and only if switches A and B are closed ($a = 1$ and $b = 1$). In Fig. 3.2 it passes if one of the switches

[1] As in Volumes 1 and 2, we shall refrain in Part 1 of this book from using any mathematical concepts other than the four most common operations, and this restriction will even include the theory of sets in its elementary form. In Part 2, all these concepts will be considered from a more advanced mathematical standpoint.

[2] In this case, as will immediately be apparent, the *and/or* is reduced to *or*.

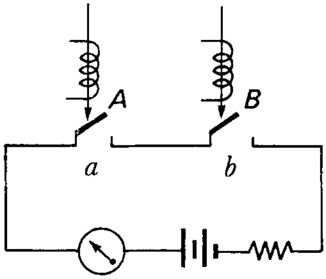


FIG. 3.1

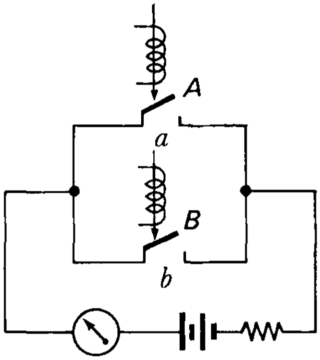


FIG. 3.2

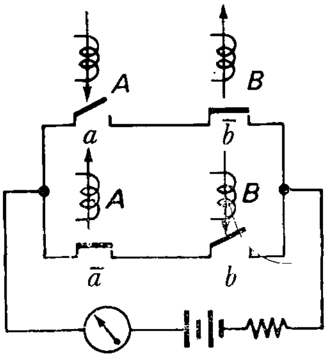


FIG. 3.3

is closed ($a = 1$ and/or $b = 1$). In Fig. 3.3 it passes if one and only one of the switches is closed (the closure of switch A at the top ensures the opening of switch A at the bottom, the same applying reciprocally to B but with the simultaneous situation reversed).

Henceforward, we shall designate by the terms "binary variable," "bivalent variable," or even "Boolean variable" a variable x that can only assume the values 0 or 1. But these variables can equally express any alternative such as black or white, true or false, dead or alive, and so on. For this purpose, it will suffice to establish an arbitrary connection between the pair 0 and 1 and whichever pair is being considered for an alternative.

4. Boolean Functions

Let us recall the concept of a system introduced earlier in this section and consider a system **S** with n components $S_1, S_2, \ldots, S_n$. Let us represent the state of each component by a binary variable x_i, $i = 1, 2, \ldots, n$. Let us further suppose that the system **S** can, in turn, assume one of the two states represented by the binary variable $y = 0$ or 1. We say that the state of system **S** is a Boolean function of the n binary variables $x_1, x_2, \ldots, x_n$.

To begin with, let us consider the number of separate Boolean functions that can exist for n Boolean variables.

For $n = 1$, namely, $y = f(x_1)$, there are $2^{(2^1)} = 4$ different Boolean functions, which are given below (this being a trivial case):

(3.16)

x_1	$f_0(x_1)$
0	0
1	0

$f_0(x_1) = 0$

x_1	$f_1(x_1)$
0	0
1	1

$f_1(x_1) = x_1$

x_1	$f_2(x_1)$
0	1
1	0

$f_2(x_1) = \bar{x}_1$

x_2	$f_3(x_1)$
0	1
1	1

$f_3(x_1) = 1$

For $n = 2$, namely, $y = f(x_1, x_2)$, the number is already appreciably greater: $2^{(2^n)} = 16$, and is shown in table (3.17). All these functions are expressed from the three basic functions (it being possible to choose others): $f_2(x_1)$ (see (3.16)), $f_1(x_1, x_2)$ and $f_7(x_1, x_2)$ (see (3.17)). To avoid any omission or repetition in the enumeration of the 16 functions given in (3.17), the numbers from 0 to 15 have been shown vertically in binary form, starting from the bottom, and the digits have formed the column $f_1(x_1, x_2)$ for $i = 0, 1, 2, \ldots, 15$.

The number of separate Boolean functions with three variables x_1, x_2, x_3 is $2^{(2^3)} = 2^8 = 256$; with n variables $x_1, x_2, \ldots, x_n$ it is $2^{(2^n)}$. The number of possible Boolean functions very soon assumes vast proportions.

From the above, it is possible to form some interesting conclusions. In the first place, if we consider all the systems with n binary components that can assume a binary value, each system contains 2^n possible states and there are neither more nor less than $2^{(2^n)}$ systems of such a kind.

As a demonstration of this, we give a table showing the smaller powers of 2^n and 2^{2n}.

This table (Fig. 3.4) proves that the number of possible systems with binary components becomes infinitely greater when n exceeds 6 or 7 than the number of states that these systems are capable of assuming. For systems formed of tertiary or quaternary instead of binary components, the problem becomes infinitely more difficult.

It is now possible to enunciate the following general property that is not as unimportant as it might appear.

The number of separate systems with n components, each of which is capable of assuming a number of states equal to or greater than 2, is infinitely

(3.17)

0

x_1	x_2	$f_0(x_1,x_2)$
0	0	0
0	1	0
1	0	0
1	1	0

$f_0(x_1,x_2) = 0$

1

x_1	x_2	$f_1(x_1,x_2)$
0	0	0
0	1	0
1	0	0
1	1	1

$f_1(x_1,x_2) = x_1.x_2$

2

x_1	x_2	$f_2(x_1,x_2)$
0	0	0
0	1	0
1	0	1
1	1	0

$f_2(x_1,x_2) = x_1.\bar{x}_2$

3

x_1	x_2	$f_3(x_1,x_2)$
0	0	0
0	1	0
1	0	1
1	1	1

$f_3(x_1,x_2) = x_1$

4

x_1	x_2	$f_4(x_1,x_2)$
0	0	0
0	1	1
1	0	0
1	1	0

$f_4(x_1,x_2) = \bar{x}_1.x_2$

5

x_1	x_2	$f_5(x_1,x_2)$
0	0	0
0	1	1
1	0	0
1	1	1

$f_5(x_1,x_2) = x_2$

6

x_1	x_2	$f_6(x_1,x_2)$
0	0	0
0	1	1
1	0	1
1	1	0

$f_6(x_1,x_2) = \bar{x}_1 x_2 \dot{+} x_1 \bar{x}_2$

7

x_1	x_2	$f_7(x_1,x_2)$
0	0	0
0	1	1
1	0	1
1	1	1

$f_7(x_1,x_2) = x_1 \dot{+} x_2$

8

x_1	x_2	$f_8(x_1,x_2)$
0	0	1
0	1	0
1	0	0
1	1	0

$f_8(x_1,x_2) = \bar{x}_1 . \bar{x}_2$

9

x_1	x_2	$f_9(x_1,x_2)$
0	0	1
0	1	0
1	0	0
1	1	1

$f_9(x_1,x_2) = \bar{x}_1 . \bar{x}_2 \dotplus x_1 x_2$

10

x_1	x_2	$f_{10}(x_1,x_2)$
0	0	1
0	1	0
1	0	1
1	1	0

$f_{10}(x_1,x_2) = \bar{x}_2$

11

x_1	x_2	$f_{11}(x_1,x_2)$
0	0	1
0	1	0
1	0	1
1	1	1

$f_{11}(x_1,x_2) = x_1 \dotplus \bar{x}_2$

(3.17)
(continued)

12

x_1	x_2	$f_{12}(x_1,x_2)$
0	0	1
0	1	1
1	0	0
1	1	0

$f_{12}(x_1,x_2) = \bar{x}_1$

13

x_1	x_2	$f_{13}(x_1,x_2)$
0	0	1
0	1	1
1	0	0
1	1	1

$f_{13}(x_1,x_2) = \bar{x}_1 \dotplus x_2$

14

x_1	x_2	$f_{14}(x_1,x_2)$
0	0	1
0	1	1
1	0	1
1	1	0

$f_{14}(x_1,x_2) = \bar{x}_1 \dotplus \bar{x}_2$

15

x_1	x_2	$f_{15}(x_1,x_2)$
0	0	1
0	1	1
1	0	1
1	1	1

$f_{15}(x_1,x_2) = 1$

n	Number of separate states of a system with n binary components	Number of separate systems with n binary components
	2^n	$2^{(2^n)}$
1	$2^1 = 2$	$2^{(2^1)} = 2^2 = 4$
2	$2^2 = 4$	$2^{(2^2)} = 2^4 = 16$
3	$2^3 = 8$	$2^{(2^3)} = 2^8 = 256$
4	$2^4 = 16$	$2^{(2^4)} = 2^{16} = 65\,536$
5	$2^5 = 32$	$2^{(2^5)} = 2^{32} = 0.42950\text{x}10^{10}$
6	$2^6 = 64$	$2^{(2^6)} = 2^{64} = 0.18447\text{x}10^{20}$
7	$2^7 = 128$	$2^{(2^7)} = 2^{128} =$ a number with 39 digits!

FIG. 3.4

larger than the possible number of states that each of these systems can assume.[1]

This principle will be of implicit importance in the theory of programs with integer values solved by Boolean methods.

5. Important Properties of Boole's Binary Algebra

Let us take as fundamental or basic operations of Boole's binary algebra, the following operations:

(3.17a) complement or negation: $\bar{a}$ (function $f_2(x_1)$);

(3.17b) Boolean addition or union: $a \dotplus b$ (function $f_7(x_1, x_2)$);

(3.17c) multiplication or intersection: $a \,.\, b$ (function $f_1(x_1, x_2)$).

It is possible to express all the Boolean functions by means of only two

[1] This also explains why so many researchers are at present interested in machines capable of adapting the information that is fed into them. The coordinator has a fixed or almost fixed structure; a machine capable of *artificial intelligence* would need to possess the quality of *self-structuring* to a highly diversified degree. When we think, we not only change the states of our neurons but the entire shape of our cerebral system.

operations or even of only one differing from the three operations just given, but their expression then becomes much more complicated. As is now the usual and even standard practice, we shall most frequently make use of the operations $(-)$, $(+)$, and $(.)$.

Let us now examine the properties of this binary algebra.

A primary property is derived from the numbers themselves: 0 and 1. Each is equal to its respective square.

$$0^2 = 0 \tag{3.18}$$

and

$$1^2 = 1. \tag{3.19}$$

The equation

$$a^2 - a = 0 \qquad \text{or} \qquad a^2 = a \tag{3.20}$$

has as its solutions: $a = 0$ or $a = 1$.

It is easy to show that, whatever more or less complex operations are applied to the binary variables $x_i = 0$ or 1, selected from the elementary operations specified in (3.16) and (3.17) and associated in whatever manner, we always obtain binary functions.

Let us examine other elementary properties, first recalling the results obtained from the three basic operations:

$$0+0 = 0, \qquad 0.0 = 0, \qquad \bar{0} = 1-0 = 1, \tag{3.21}$$

$$0+1 = 1, \qquad 0.1 = 0, \tag{3.22}$$

$$1+0 = 1, \qquad 1.0 = 0, \qquad \bar{1} = 1-1 = 0, \tag{3.23}$$

$$1+1 = 1, \qquad 1.1 = 1.$$

Let us now see which are the principal properties or formulas of Boole's binary algebra, which the reader will wish to prove with the help of Eqs. (3.21)–(3.23).

(3.24) $a.b = b.a$, hence multiplication is commutative,

(3.25) $(a.b).c = a.(b.c)$, it is also associative,

(3.26) $a.a = a$,

(3.27) $a.\bar{a} = 0$,

(3.28) $a.0 = 0$,

(3.29) $a.1 = a$.

(3.30) $a+b = b+a$, Boolean addition is therefore commutative,

(3.31) $(a+b)+c = a+(b+c)$, it is also associative, (3.32) $a \dotplus a = a,$

(3.33) $a \dotplus \bar{a} = 1,$

(3.34) $a \dotplus 0 = a,$

(3.35) $a \dotplus 1 = 1.$

(3.36) $a.(b \dotplus c) = a.b \dotplus a.c,$

(3.37) $a \dotplus (b.c) = (a \dotplus b).(a \dotplus c).$

The two last properties (3.36) and (3.37) are known, respectively, as distributivity of Boolean multiplication in relation to Boolean addition and distributivity of Boolean addition in relation to Boolean multiplication. The latter property differs from the corresponding property encountered in common algebra.

It is true that in common algebra we have the identical formula

$$a.(b+c) = a.b+a.c,$$

but we do not have the same

$$a+(b.c) = (a+b).(a+c).$$

We also find

(3.38) $\overline{(\bar{a})} = a,$

which is evident from the given definition.

The following two important properties form *De Morgan's theorem*:

(3.39) $\overline{a\text{–}b} = \bar{a} \dotplus \bar{b},$

(3.40) $\overline{a \dotplus b} = \bar{a}\text{–}\bar{b}.$

Their proof is very simple: all that is required is to utilize the definitions of the operations ($\dotplus$), (.), and (—). Using this method, the proof of (3.39) is given in Fig. 3.5.

Two other very useful properties can be proved just as easily.

(3.41) $\bar{a} \dotplus a.b \;= \bar{a} \dotplus b,$

(3.42) $\bar{a}.(a \dotplus b) = \bar{a}.b.$

	a	b	$a.b$	$\overline{a.b}$	a	b	$\bar{a}$	$\bar{b}$	$\bar{a} \dotplus \bar{b}$
0	0	0	0	1	0	0	1	1	1
1	0	1	0	1	0	1	1	0	1
2	1	0	0	1	1	0	0	1	1
3	1	1	1	0	1	1	0	0	0

FIG. 3.5

It is interesting to note how certain simplifications can be performed, for which one example will suffice:

(3.43) $$\begin{aligned} & a.b.(a+b).(\bar{a}+\bar{c}) \\ &= a.b.(a.\bar{a}+a.\bar{c}+\bar{a}.b+b.\bar{c}) \quad \text{from (3.36)} \\ &= a.b.\bar{c}, \quad \text{since } a.b.(a+b)=a.b \text{ and } a.\bar{a}=0. \end{aligned}$$

Finally, a frequently used property is "absorption":

(3.44) $$a+a.b=a \quad \text{and} \quad a.(a+b)=a.$$

6. The Plumber or the Tile Removal Problem

We shall now present a practical problem (evocative, even if it may appear rather naïve) of programming with bivalent variables, which, as we should recall, involves the use of the values 0 and 1 only. For this purpose we shall apply an algorithm that will demonstrate the properties of Boole's algebra. The derivation of this problem is from the printed circuits of electronic appliances [K48], and the somewhat naïve form in which it is presented here retains all its essential characteristics. The algorithm used is given in our reference [K50].

Let us, then, consider a plumber who has to fit a number of valves in a series of pipes running beneath a floor covered with heavy square tiles (Fig. 3.6). He can fit the valves anywhere he wishes in the pipes, but only one valve must be fitted to each pipe. In order to minimize the effort required, he endeavors to find the least number of tiles that must be moved so that there will be one valve fitted to each pipe.

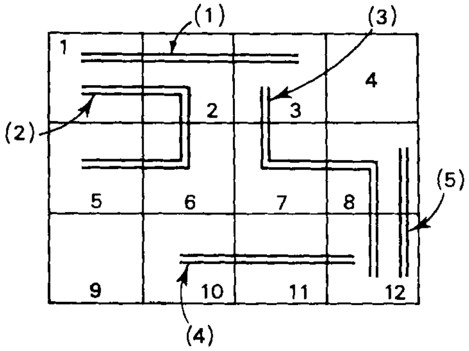


FIG. 3.6

Let us number the tiles from 1 to 12 and the pipes from (1) to (5) (Fig. 3.6). Let us postulate:

(3.45) $$\begin{aligned} x_i &= 0 \quad \text{if, in the selected solution, tile } i \text{ is not removed,} \\ &= 1 \quad \text{if, in the solution, tile } i \text{ is removed,} \\ i &= 1, 2, \ldots, 12. \end{aligned}$$

For each pipe there will be a corresponding constraint showing that at least one tile must be removed to uncover it. Thus, for pipe (1) we shall have

$$x_1+x_2+x_3 \geqslant 1 \tag{3.46}$$

since, for every solution, at least one of the tiles that cover it must be removed, that is to say $x_1 = 1$ and/or $x_2 = 1$ and/or $x_3 = 1$. Hence, we shall have the function to be minimized together with the five constraints that constitute the linear program in bivalent variables:

$$\begin{aligned}
&[\text{MIN}]\ z = x_1+x_2+x_3+x_4+x_5+x_6+x_7+x_8+x_9+x_{10}+x_{11}+x_{12}, \\
&x_1+x_2+x_3 \geqslant 1 \qquad \text{(pipe 1)}, \\
&x_1+x_2+x_5+x_6 \geqslant 1 \qquad \text{(pipe 2)}, \\
&x_3+x_7+x_8+x_{12} \geqslant 1 \qquad \text{(pipe 3)}, \\
&x_{10}+x_{11}+x_{12} \geqslant 1 \qquad \text{(pipe 4)}, \\
&x_8+x_{12} \geqslant 1 \qquad \text{(pipe 5)}, \\
&x_i = 0 \text{ or } 1 \qquad i = 1, 2, \ldots, 12.
\end{aligned} \tag{3.47}$$

By considering the Boolean condition $x_i = 1, 2, \ldots, 12$, instead of $x_i = 0$ or 1, we should be confronted with an ordinary linear program that we could solve, for example, with the help of one of the algorithms described in the first volume. It is clear that, if the solution obtained by this means does not include any number other than 0 or 1, our problem has been solved. It is shown in [K48] and [K50] that this is a good method[1] for solving this problem which is one of a class of problems referred to as *the covering of a set* (see our reference [K76]).

Nonetheless, in solving this problem, we shall make use of a method differing from ordinary linear programs and one that is typically Boolean,

$$\begin{aligned}
&(x_1 \dotplus x_2 \dotplus x_3)\cdot(x_1 \dotplus x_2 \dotplus x_5 \dotplus x_6).(x_3 \dotplus x_7 \dotplus x_8 \dotplus x_{12}) \\
&\qquad .(x_{10} \dotplus x_{11} \dotplus x_{12}).(x_8 \dotplus x_{12}) = 1,
\end{aligned} \tag{3.48}$$

in its imposed condition of having a minimum of variables $x_i = 1$ (since every value $x_i = 1$ implies the removal of a tile). Equation (3.48) means that each term within the bracket must be equal to 1, since each pipe must have one valve.

We shall now simplify (3.48) by making use of various properties given in this section.

Let us first notice that $(x_8 \dotplus x_{12})$ is included in $(x_3 \dotplus x_7 \dotplus x_8 \dotplus x_{12})$ and that, in accordance with (3.44), we have

$$(x_3 \dotplus x_7 \dotplus x_8 \dotplus x_{12}).(x_8 \dotplus x_{12}) = (x_8 \dotplus x_{12}). \tag{3.49}$$

[1] It is useful to familiarize the reader with the basic concept of limits used in mathematical programming. The conditions $x_i \geqslant 0$, $i = 1\ 2, \ldots, 12$, constitute less of a constraint than the conditions $x_i = 0$ or 1, $i = 1, 2, \ldots, 12$, to the extent that the common linear program associated with (3.47) produces a smaller or equal minimum, that is to say a lower, nonstrict limit for the minimal solution of (3.47).

Thus we can suppress $(x_3 \dotplus x_7 \dotplus x_8 \dotplus x_{12})$ in (3.48), and we then have

$$(3.50) \qquad (x_1 \dotplus x_2 \dotplus x_3) . (x_1 \dotplus x_2 \dotplus x_5 \dotplus x_6) . (x_{10} \dotplus x_{11} \dotplus x_{12}) . (x_8 \dotplus x_{12}) = 1 .$$

Let us expand (3.50) by considering the first term and by then applying the property of (3.44) a second time. It follows successively that

$$(3.51) \qquad \begin{aligned} &(x_1 \dotplus x_2) . (x_1 \dotplus x_2 \dotplus x_5 \dotplus x_6) . (x_{10} \dotplus x_{11} \dotplus x_{12}) . (x_8 \dotplus x_{12}) \\ &\quad \dotplus x_3 . (x_1 \dotplus x_2 \dotplus x_5 \dotplus x_6)\, (x_{10} \dotplus x_{11} \dotplus x_{12}) . (x_8 \dotplus x_{12}) = 1 . \end{aligned}$$

$$(3.52) \qquad \begin{aligned} &(x_1 \dotplus x_2) . (x_{10} \dotplus x_{11} \dotplus x_{12}) . (x_8 \dotplus x_{12}) \\ &\quad \dotplus x_3 . (x_1 \dotplus x_2 \dotplus x_5 \dotplus x_6)\, (x_{10} \dotplus x_{11} \dotplus x_{12}) . (x_8 \dotplus x_{12}) = 1 . \end{aligned}$$

We shall proceed in the same manner until there is no common term left among the terms within the brackets, and we then obtain

$$(3.53) \qquad \begin{aligned} &(x_1 \dotplus x_2) . (x_{10} \dotplus x_{11} \dotplus x_{12}) . x_8 \dotplus (x_1 \dotplus x_2) . x_{12} \\ &\qquad \dotplus x_3 . (x_1 \dotplus x_2 \dotplus x_5 \dotplus x_6)\, (x_{10} \dotplus x_{11} \dotplus x_{12}) . x_8 \\ &\qquad \dotplus x_3 (x_1 \dotplus x_2 \dotplus x_5 \dotplus x_6) . x_{12} = 1 . \end{aligned}$$

Finally, after the simplifications have been completed, we have

$$(3.54) \qquad \begin{aligned} &x_1 . x_{12} \dotplus x_2 . x_{12} \dotplus x_1 . x_{10} . x_8 \dotplus x_1 . x_{11} . x_8 \dotplus x_2 . x_{10} . x_8 \dotplus x_2 . x_{11} . x_8 \\ &\quad \dotplus x_3 . x_5 . x_{12} \dotplus x_3 . x_6 . x_{12} \dotplus x_3 . x_5 . x_{10} . x_8 \dotplus x_3 . x_6 . x_{10} . x_8 \\ &\quad \dotplus x_3 . x_5 . x_{11} . x_8 \dotplus x_3 . x_6 . x_{11} . x_8 = 1 . \end{aligned}$$

We obtain a solution involving a minimal number of tiles by equating one or other of the two monomials of the lowest degree of (3.54) with 1, namely, for

(3.55)

(a) $x_1 = 1,\ x_{12} = 1, \quad x_i = 0, \qquad i = 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,$

or

(b) $x_2 = 1,\ x_{12} = 1; \quad x_i = 0, \qquad i = 1, 3, 4, 5, 6, 7, 8, 9, 10, 11 .$

We now have two solutions involving the removal of not more than two tiles, and we can verify that any other solution would involve a greater number. Thus (3.55) provides two optimal solutions starting with the Boolean polynomial (3.54). Unfortunately, the polynomial form of (3.54) obtained from (3.50) is difficult to program with existing machines, easy though it is to calculate mathematically. Nevertheless, if the number of tiles and of pipes were substantially greater, it would be necessary to make use of an algorithm and program the results. We could, for example, replace the general condition $x_i = 0$ or 1, $i = 1, 2, \ldots, 12$, by the condition $0 \leqslant x_i \leqslant 1$ and employ a linear program, although this would not necessarily provide an optimal solution in integers. We could, however, round off in the result every $x_i < 1$; this would not, as a rule, produce an optimal solution, but the problem could then be easily solved. Thus, a solution that is optimal for the linear program might require, for 100 tiles and 30 pipes, only a few seconds calculation on a computer of the third generation. However, we shall give some special algorithms for these problems with integer solutions.

7. Some Remarks on the Subject of Problems with Integer Values Solved by Ordinary Linear Programming

We note first that it is possible, in certain cases, to obtain a solution for these problems by using the solution or solutions derived from the corresponding linear program. This is an obvious procedure.

Let us consider an example:

$$\text{(3.56)} \qquad \begin{aligned} &[\text{MAX}]\ z = 3x_1+3x_2,\\ &11x_1+4x_2 \leqslant 44,\\ &3x_1+5x_2 \leqslant 30,\\ &x_1 \text{ and } x_2 \text{ nonnegative integers.} \end{aligned}$$

If we eliminate the condition requiring integer values, that is, if we consider the same program with x_1 and x_2 only as nonnegative, we obtain as an optimal solution

$$\text{(3.57)} \qquad x_1 = 2\ 14/43, \quad x_2 = 4\ 26/43, \quad z = 894/43 = 20\ 34/43.$$

By rounding off x_1 and x_2 to the integer values immediately below $x_1 = 2$ and $x_2 = 4$, we obtain a point $[x_1, x_2] = [2, 4]$, which satisfies the constraints of (3.56) and gives $z = 18$. From all the evidence, this point gives a *lower limit* for z, whereas $z = 20\ 34/43$ gives an *upper limit*, since it corresponds to a program of maximization with less constraint. We can be certain that the solution of (3.56) will be such that

$$\text{(3.58)}^{1} \qquad 18 \leqslant z \leqslant 20\,,$$

since the costs of x_1 and x_2 are integers and because, by accepting an error less than or equal to 10%, we have obtained a solution of the given problem. It is often possible and acceptable to find the solution of programs with integer values by such means.

[1] By putting the sign $\leqslant$ in front of 20 34/43 we agree to accept another solution giving the same value 20.

In general, if the values obtained for the variables are "moderately" large, we shall not commit too serious an error if we use the common linear program and round off to the integer values (immediately below for a maximization and immediately above for a minimization): this is, of course, on condition that the procedure produces a possible solution and that the coefficients of the economic function are positive.

Hence, programs with integer values are especially difficult to solve when the variables have small optimal values, in other words, when we are dealing with values closely approximating to the first whole numbers.

Let us also note that, if the solution of the linear program happened to be integer, we should have obtained the optimal integer solution of the program. This particularly applies to the problems of assignment and of transportation given in Volume 1, for which we did not, nevertheless, use the method of the simplex or of one of its variants but more specialized methods such as the Hungarian method for the problem of assignment and that of the *stepping-stone* for the problem of transportation. This is because the matrix of the constraints, in such problems, possesses a special property, *total unimodularity.*[1]

As a result, we shall not, in the present volume, further consider problems of this nature, but shall instead turn our attention to problems that are less specialized and also more frequently encountered in practice in operations research.

Section 4. **Methods of Solving Programs with Integer Values by Enumeration**

1. The Principle of Finding Solutions by Enumeration

For a problem with integer numbers, the solutions are usually denumerable and finite. For instance, in a problem in which the variables are x_1, x_2, x_3, x_4, and in which

$$\begin{aligned} &0 \leqslant x_1 \leqslant 3, \\ &0 \leqslant x_2 \leqslant 1, \\ &0 \leqslant x_3 \leqslant 4, \\ &0 \leqslant x_4 \leqslant 5, \end{aligned} \tag{4.1}$$

the total number of solutions is equal to $4 \times 2 \times 5 \times 6 = 240$. It is therefore possible to contemplate the enumeration of all the solutions and, in each case,

[1] The determinants of matrices with values of 0 and 1, extracted from the matrices of the constraints, always possess values equal to -1, 0, or 1.

to evaluate whether it belongs to the domain defined by the constraints and, if the answer is affirmative, to calculate the corresponding value of the economic function.

After enumerating all the solutions, we should retain from among those that satisfy all the constraints, the solution or solutions that optimize the economic function.

We shall, however, discover that partial and a priori knowledge concerning the optimal solution or solutions will obviate the enumeration of all the solutions. Thus, if it is required to minimize $3x_1+2x_2+10x_3+15x_4$ and it is known that the minimum is less than 8, there is no need to consider points such that $x_3 > 0$ and/or $x_4 > 0$, which would automatically give a higher value than 8. In the methods that we shall demonstrate, enumeration will not be complete but only *implicit.*

In another connection, we may consider that combinatorial analysis (what is referred to by the terms *combinatory* or *combinatorial*) is also concerned with all the problems of classification and rearrangement of subsets and, in particular, with optimization.

Let us begin with some very elementary problems that have few variables and few constraints.

Let us consider the following constraints and let us try to discover all the solutions with integer values that satisfy them.

$$\begin{aligned} &11x_1+4x_2 \leqslant 44, \\ &3x_1+5x_2 \leqslant 30, \\ &x_1 \leqslant 3, \\ &x_1 \geqslant 0, \quad x_2 \geqslant 0. \end{aligned} \tag{4.2}$$

It is sufficient to draw Fig. 4.1 in order to obtain the 21 solutions (shown by heavy dots on the diagram). As will be seen, enumèration is simple and immediately productive in a case such as this.

Let us now turn to another and scarcely more complicated case, where we have three variables and three constraints.

Let

$$\begin{aligned} &18x_1+18x_2+14x_3 \leqslant 63, \\ &72x_1+112x_2+126x_3 \leqslant 252, \\ &2x_3 \leqslant 3, \\ &x_1 \geqslant 0, x_2 \geqslant 0, x_3 \geqslant 0. \end{aligned} \tag{4.3}$$

It is still possible to show by a diagram the domain defined by the six planes

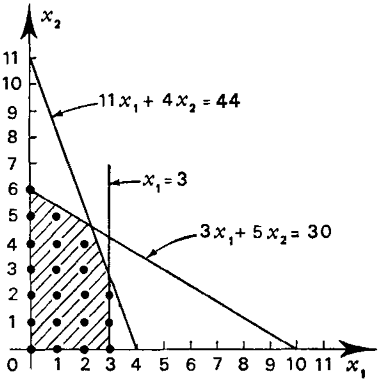


FIG. 4.1

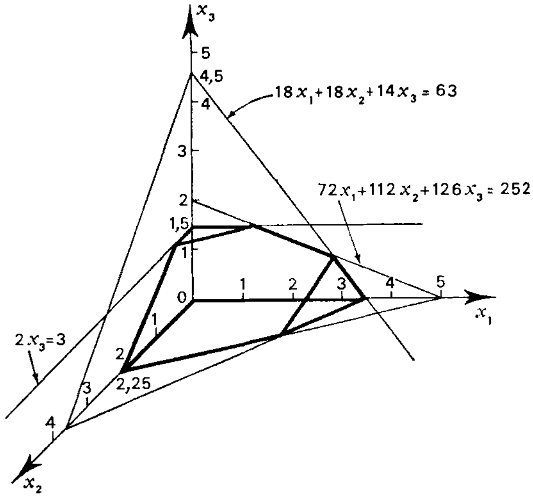


FIG. 4.2

that can be considered if we employ the equals sign in these inequalities, as has been done in Fig. 4.2. But it is very difficult, because of the perspective, to determine whether some points with integer coordinates belong or do not belong to the domain (it could be done by using plane geometry, but the process would be needlessly lengthy). Let us therefore use enumeration to investigate all the possible solutions; to do so, let us begin by limiting the values that are acceptable for the variables. It is clear, from an examination of Fig. 4.2, that we should have $x_1 < 3.5$, $x_2 < 2.25$, and $x_3 < 1.5$. Hence, we will consider which are the points $[x_1, x_2, x_3]$; $x_1 = 0, 1, 2, 3$; $x_2 = 0, 1, 2$, and $x_3 = 0, 1$ that satisfy the three constraints. This has been done in the table shown in Fig. 4.3. In column (1) we have enumerated all the points x_1, x_2, x_3 with $x_1 = 0, 1, 2, 3$, $x_2 = 0, 1, 2$, and $x_3 = 0, 1$. The order in which the enumeration has been performed is termed lexicographical, as we will shortly explain. Columns (2), (3), and (4) contain the results of the evaluation of the first members of the relations (4.2). In these columns the symbol $\overset{\circ}{\circ}$ signifies *compared with.* In column (5) we have introduced the symbol $\in$ if the point belongs to the given domain and the symbol $\notin$ if the point does not belong to it. The 10 points that constitute solutions are shown by asterisks to the left of column (1).

It would have been possible to restrict the enumeration by making use of the fact that it is impossible to have both $x_1 = 3$ and $x_2 = 0$ and at the same time to have $x_1 \geqslant 1$, $x_2 \geqslant 1$, $x_3 \geqslant 1$, or other constraints of this kind. Such conclusions are clearly very useful in reducing the scope of the enumeration.

However, if there were more than three variables and several constraints, such enumeration *by hand* would be virtually impossible in practice and, except in some special cases, might prove beyond the capacity of even the most powerful computers, unless one were prepared to display a very great degree of patience and to accept huge costs for the calculation.

To enumerate solutions for problems of a combinatorial nature it is appropriate to use procedures without omission or redundancy, of which we shall now examine a few examples.

	(1)	(2)	(3)	(4)	(5)
	$[x_1, x_2, x_3]$	$18x_1$ $+ 18x_2$ $+ 14x_3$ $\overset{\circ}{\circ}\ 63$	$72x_1$ $+ 112x_2$ $+ 126x_3$ $\overset{\circ}{\circ}\ 252$	$2x_3$ $\overset{\circ}{\circ}\ 3$	$\in$ or $\notin$
*	[0,0,0]	0	0	0	$\in$
*	[0,0,1]	14	126	2	$\in$
*	[0,1,0]	18	112	0	$\in$
*	[0,1,1]	32	238	2	$\in$
*	[0,2,0]	36	224	0	$\in$
	[0,2,1]	50	350	2	$\notin$
*	[1,0,0]	18	72	0	$\in$
*	[1,0,1]	32	198	2	$\in$
*	[1,1,0]	36	184	0	$\in$
	[1,1,1]	50	310	2	$\notin$
	[1,2,0]	54	296	0	$\notin$
	[1,2,1]	68	422	2	$\notin$
*	[2,0,0]	36	144	0	$\in$
	[2,0,1]	50	270	2	$\notin$
	[2,1,0]	54	256	0	$\notin$
	[2,1,1]	68	382	2	$\notin$
	[2,2,0]	72	368	0	$\notin$
	[2,2,1]	86	494	2	$\notin$
*	[3,0,0]	54	216	0	$\in$
	[3,0,1]	68	342	2	$\notin$
	[3,1,0]	72	328	0	$\notin$
	[3,1,1]	86	454	2	$\notin$
	[3,2,0]	90	440	0	$\notin$
	[3,2,1]	104	566	2	$\notin$

FIG. 4.3

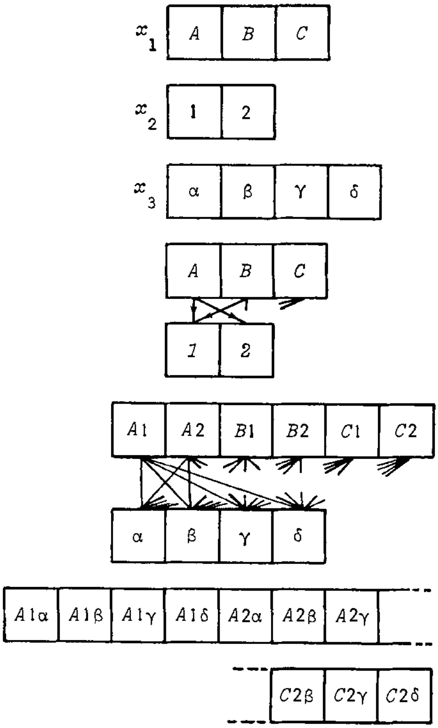


FIG. 4.4

2. Enumeration without Omission or Redundancy. Lexicographical Procedure

The currently most accepted procedure is to construct a lexicographical order similar, for instance, to that of a dictionary (as the adjective implies) or to that of the number plates of automobiles. The procedure is very simple, and one example will suffice to explain it. Let us suppose there are three variables, numerical or otherwise, such that $x_1 = A, B, C$; $x_2 = 1, 2$; $x_3 = \alpha, \beta, \gamma, \delta$. The first step is to allot an order of enumeration to the variables in relation to each other. Let us suppose that x_1 is placed the furthest to the left followed by x_2 and x_3. We then associate the values of x_1 with those of x_2 in such a manner as to arrange the first of x_1 with the first of x_2, the first of x_1 with the second of x_2 and so on; this will give us $A1$, $A2$, $B1$, $B2$, $C1$, $C2$. The result will then be arranged with the values of x_3, which will give us $A1\alpha$, $A1\beta$, $A1\gamma$, $A1\delta$, $A2\alpha$, ..., $C2\beta$, $C2\gamma$, $C2\delta$ (Fig. 4.4).

Two further examples are shown in Figs. 4.5 and 4.6. For Fig. 4.5 we have $x_1 = 0, 1$; $x_2 = 0, 1$; $x_3 = 0, 1$; $x_4 = 0, 1$. For Fig. 4.6 we have $x_1 = 0, 1, 2$; $x_2 = 0, 1$; $x_3 = 0, 1, 2, 3$.

A little further on we shall return to the use of this procedure when constraints intervene, whether these are numerical relations or otherwise.

We must not conclude these explanations on lexicographical procedures without making it clear that the selected orders are entirely arbitrary.

It is equally important to note that if x_1 can assume n_1 values, x_2 can assume n_2, x_r can assume n_r, so that there are exactly

(4.4) $n_1 \times n_2 \times \dots \times n_r$ grandeurs $[x_1, x_2, \dots, x_r]$.

3. Arborescence

The concept of arborescence was introduced in Volume 2, Section 44.2, and was used in Section 8.1, for automatic textual emendation. Nevertheless, we intend to define it again, having regard to the importance that it will assume further on.

An arborescence, then, is a finite graph in which the following properties can be verified:

a. The graph does not include any circuit.

b. There is one and only one vertex, termed a *root*, that is not the terminal extremity of any arc.

c. All the other vertices are the terminal extremities of a single arc.

Figure 4.7 represents an arborescence, in which vertex R is the root. A vertex which is not the initial extremity of an arc is known as a *hanging vertex*. Thus, vertices $C, P, M, U, F, E, D, Q, T, L, V, N, G, H, K$ are hanging vertices.

In Figs. 4.8 and 4.9 we have shown two different procedures for investigating, in a combinatorial manner and with the help of arborescences, the quantities $[x_1, x_2, x_3, x_4]$ where each of the terms is equal to 0 or 1.

Using the first method (Fig. 4.8), we fix the values of $[x_1, x_2, x_3, x_4]$ beginning with x_1, then x_2, and so on. With this method there is no redundancy; each of the 16 solutions is obtained only once in the arborescence.

On the other hand, in Fig. 4.9, we start from the point [0, 0, 0, 0] and replace each 0 by a 1. We perform the same operation for each 0 in the next stage and proceed in the same manner.

The binary numbers thus obtained have been expressed in decimal form to the right of the brackets. Hence it may be observed that this method of enumeration without omission results in redundancies in contradiction to the first method.

4. Hamming's Distance

This is a very simple and useful concept. Let us, for example, consider two quantities, each containing seven variables, $x_1, \dots, x_7$, that in the first have the respective values 0, 1, 0, 1, 1, 0, 1 and in the second 1, 1, 0, 1, 0, 0, 0. Let us

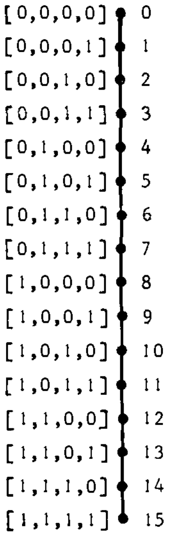


FIG. 4.5

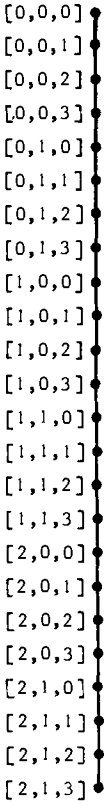


FIG. 4.6

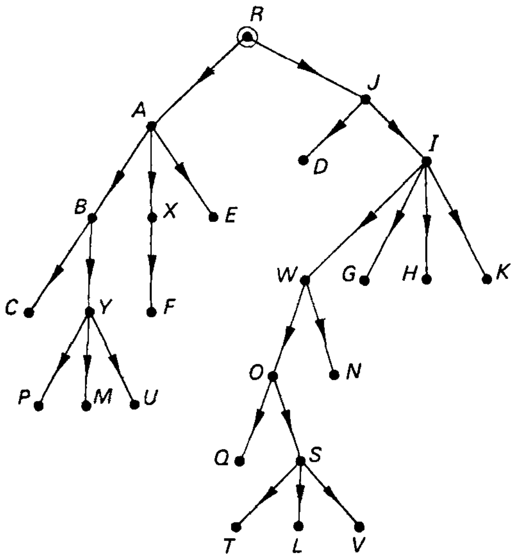


FIG. 4.7

place them one below the other.

(4.5)

x_1	x_2	x_3	x_4	x_5	x_6	x_7
0	1	0	1	1	0	1
1	1	0	1	0	0	0
-1	0	0	0	1	0	1

$$|-1| + |1| + |1| = 1+1+1 = 3.$$

Let us calculate the difference between each element in the first line and the corresponding element in the second line and write the results under the second line. If we now calculate the *absolute values* of these results we find that their sum is equal to 3. We say that Hamming's distance between the two quantities or vectors is 3.

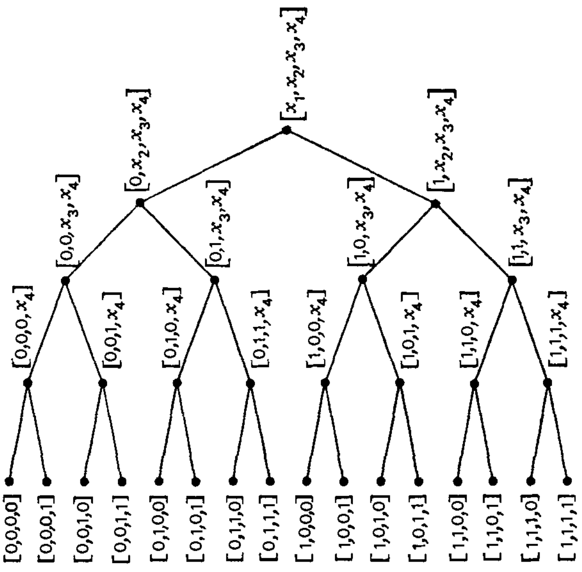


FIG. 4.8

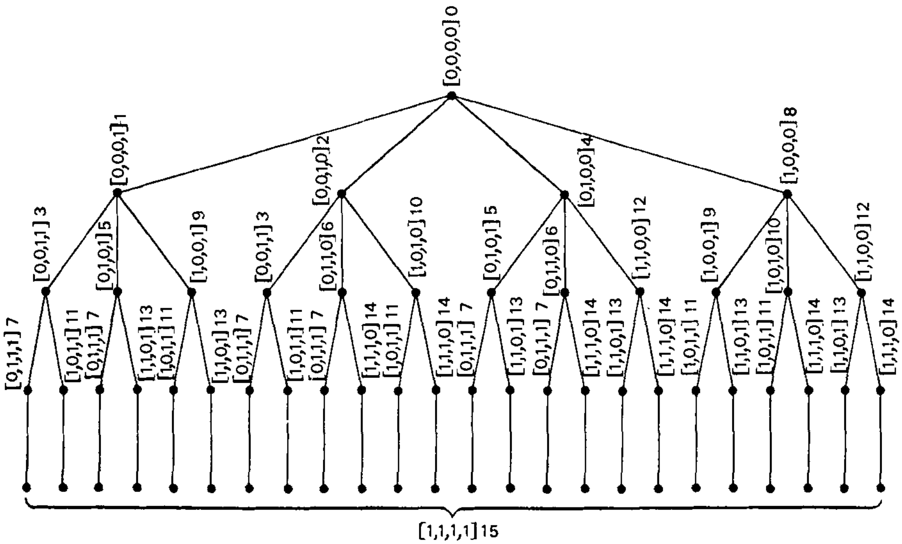


FIG. 4.9

In a more generalized way, let us consider two vectors $(x'_1, x'_2, \ldots, x'_n)$ and $(x_1, x_2, \ldots, x_n)$; we term *Hamming's generalized distance*[1] between these two

[1] We are concerned here with an extension of the concept introduced by Hamming and used, in particular, in the theory of codes. According to Hamming, the distances defined in this manner can only be applied to vectors with components formed by bivalent variables.

vectors the scalar

$$\delta = |x'_1 - x_1| + |x'_2 - x_2| + \ldots + |x'_n - x_n| . \tag{4.6}$$

If we now examine the arborescence shown in Fig. 4.9, we see that the vertices are situated at levels that are spaced at Hemming's distances of 1.

5. Lattice

We are concerned here with a mathematical concept that is too complicated to be presented in its theoretical aspects in this first part, although the reader will find them fully explained in Part 2. We shall, however, give a very elementary and much less generalized explanation that will enable the reader to understand those elements of the concept that are developed in the following sections.

For this purpose let us consider a vector[1] with components that can assume the following values: $x_1 = 0, 1, 2$; $x_2 = 0, 1$; $x_3 = 0, 1, 2, 3$. We shall classify them according to the following concept of level. Let us begin with the vector [0, 0, 0] and let us allot it the arbitrary level 0. Now, let us place on the following level 1 all the possible vectors for which Hamming's distance from the preceding level is 1 (Fig. 4.10), namely the vectors [0, 0, 1], [0, 1, 0], and

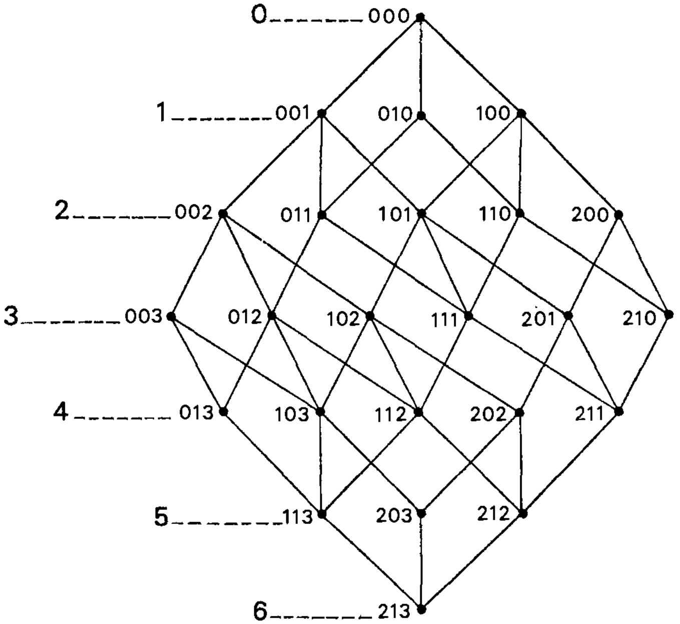


FIG. 4.10

[1] The term vector is used here in the sense of a quantity with several components, or n-tuple.

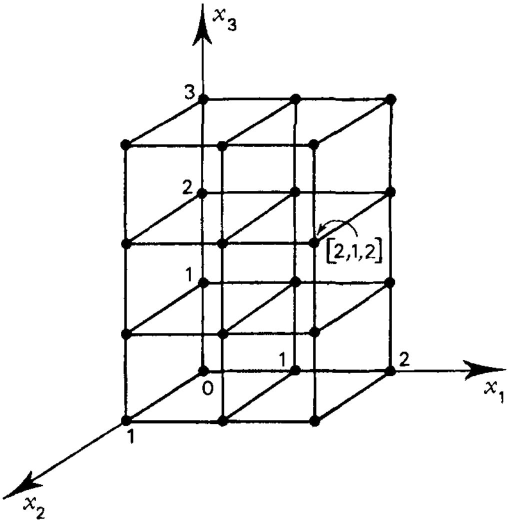


FIG. 4.11. Representation in Cartesian coordinates of the lattice shown in Fig. 4.10.

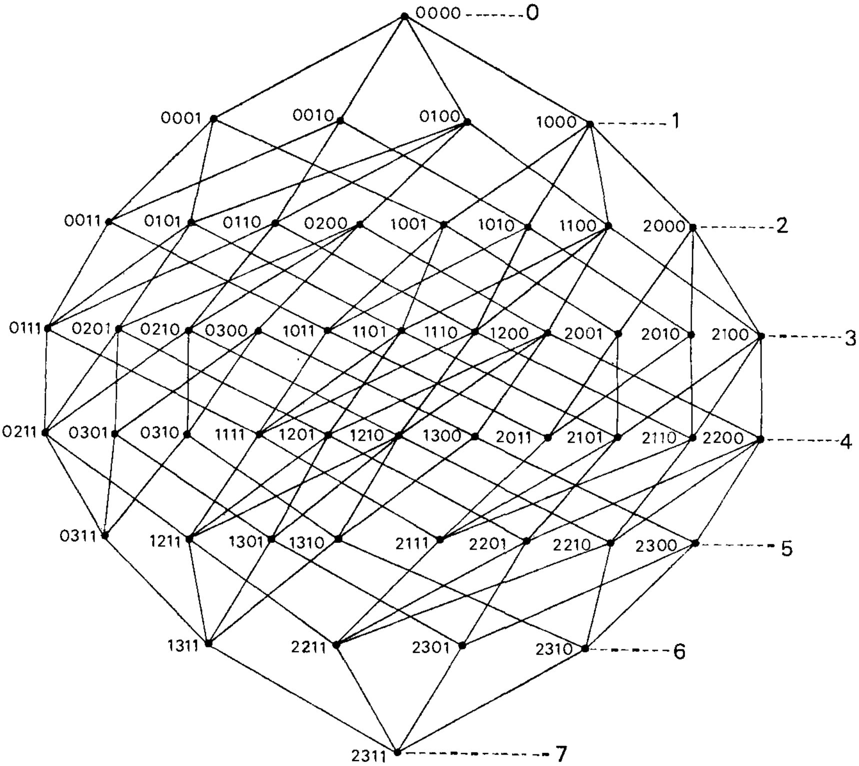


FIG. 4.12

[1, 0, 0]. Let us connect the vertex (0, 0, 0) with each of the other vertices of level 1 by a line. Let us now place on level 2 all the vectors for which Hamming's distance is 1 in relation to one of the vectors of level 1; these are the vectors [0, 0, 2], [0, 1, 1], [1, 0, 1], [1, 1, 0], and [2, 0, 0].

Connect the vertices of level 1 to the vertices of level 2 when they possess a Hamming's distance of 1. Thus we shall join [0, 0, 1] to [0, 0, 2] and to [0, 1, 1]; [0, 1, 0] to [0, 1, 1] and to [1, 1, 0], and so forth. We shall then continue the same procedure until all the vertices have been exhausted. By this means we produce a mathematical structure, termed a *vectorial lattice*, that is shown in its entirety by Fig. 4.10. But, in the case of 1, 2, or 3 components, it is possible to use a different representation of a vectorial lattice such, for example, as that shown in Cartesian coordinates in Fig. 4.11, representing the same example as that given in Fig. 4.10. With more than three components, as is well-known, representation by means of Cartesian coordinates is no longer possible.

In Fig. 4.12 we have shown a vectorial lattice with four components: $x_1 = 0, 1, 2$; $x_2 = 0, 1, 2, 3$; $x_3 = 0, 1$; and $x_4 = 0, 1$. Figures 4.13 and 4.14 represent vectorial lattices in which the components are bivalent variables. Such a lattice is termed a *Boolean lattice*. It will be seen that their representation produces a cube in the case of Fig. 4.13 and a hypercube in that of Fig. 4.14.

These vectorial lattices show the structure of vectors with integer components.[1]

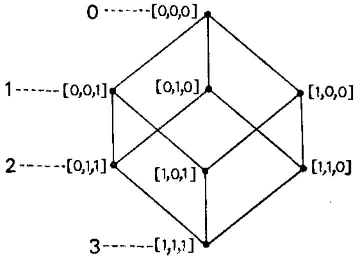


FIG. 4.13

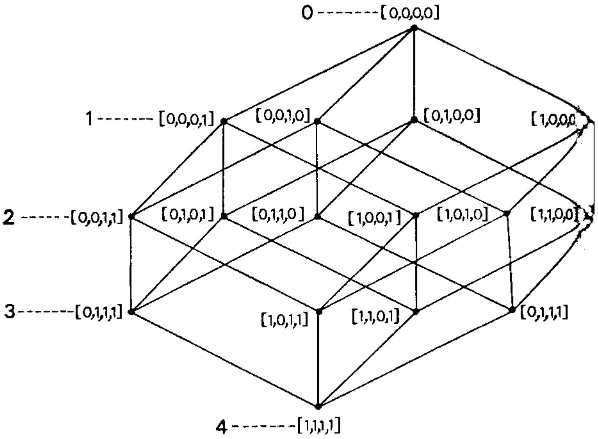


FIG. 4.14

[1] And usually with a finite number of discrete values.

A path that leads from a vertex A to a vertex B, while passing from one level to a level of a higher number is called a *chain* of the lattice, and is not to be confused with the chain of a graph. This term can also be used if the path leads from a higher to a lower level. Thus in Fig. 4.12

$$[0, 0, 1, 0], [0, 1, 1, 0], [1, 1, 1, 0], [1, 2, 1, 0], [2, 2, 1, 0]$$

is a chain.

We shall apply the term *progression* to a path that passes through vertices separated by a Hemming's distance of 1 without reference to the levels. Thus

$$[1, 2, 0, 1], [1, 1, 0, 1], [1, 0, 0, 1], [2, 0, 0, 1], [2, 1, 0, 1], [2, 2, 0, 1], [2, 3, 0, 1]$$

is a progression. A progression that never passes twice through the same vertex is known as an *elementary progression.*

All these concepts will be utilized further on and, in particular, in the algorithm for the solution of integer programs by direct search that will be explained shortly.

It is convenient when dealing with lattices to select a particular element as the point of origin; this is usually the point of which all the components are 0, for instance $[0, 0, 0]$ in Fig. 4.13. We define as the *level of an element* of a lattice, Hamming's distance of this element from the point of origin. For example, the elements $[0, 1, 1]$, $[1, 0, 1]$, $[1, 1, 0]$ of the lattice in Fig. 4.13 are at level 2.

6. Algorithm for Solving Programs with Bivalent Variables by Direct Search

At the beginning of the present section we outlined the principle of algorithms used in direct search. All that was required, for instance, in the example given in Figs. 4.2 and 4.3 was to find out whether a point $[x_1, x_2, x_3]$ belonged or did not belong to the set of constraints. We shall now introduce the economic function and confine ourselves to the case where the variables are bivalent, remembering that any program with integer values can be changed to a program with the binary values of 0 and 1 by a modification of the constraints and variables, a procedure that may sometimes result in their number being substantially increased.

Let us first examine by means of an example the procedure needed to diminish the lexicographical enumeration.

For this purpose let us take the program with bivalent variables[1]:

$$\begin{aligned} &[\mathrm{MAX}]\ z = 3x_1-2x_2+5x_3\,,\\ &(1)\ x_1+2x_2-x_3 \leqslant 2\,,\\ (4.7)\qquad &(2)\ x_1+4x_2+x_3 \leqslant 4\,,\\ &(3)\ x_1+x_2 \leqslant 3\,,\\ &(4)\ 4x_2+x_3 \leqslant 6\,,\\ &\quad x_1\,, x_2\,, x_3 = 0 \text{ or } 1\,. \end{aligned}$$

Let us suppose that we know a solution of (4.7). It is, for example, clear that $[x_1, x_2, x_3] = [1, 0, 0]$ is one solution, and in this case the economic function assumes the value $z = 3$.

We may affirm that every optimal solution will give a value for z greater than or equal to 3 (we say that 3 is a lower limit of the optimal value). We can therefore introduce a supplementary constraint

$$(4.8)\qquad (0)\quad 3x_1-2x_2+5x_3 \geqslant 3\,.$$

We thus obtain a system with five constraints to be satisfied. The supplementary constraint (4.8) will be known as the *filtering constraint*.

With the lexicographical method of enumeration we should have to calculate the left-hand members of four constraints for $2^3 = 8$ solutions, which would require 32 operations. This number can be reduced by using the filter constraint (4.8). We shall perform our operations in accordance with the following table (Fig. 4.15) in which the columns are numbered like the constraints. In calculating the values assumed by the left-hand members of the five constraints in their numerical order, we find that once a constraint is not satisfied it is unnecessary to perform the calculations for the others, thereby reducing the number of operations.

The results of the calculations are shown in Fig. 4.15, and it will be observed that instead of $5 \times 8 = 40$ calculations for x_1, x_2, and x_3, there are only 24. In other examples the reduction can be much greater.

We shall now explain by means of example (4.7) how this procedure can be used in a sequential manner. In drawing up table 4.15 we observe almost at the outset that there is a solution $[0, 0, 1]$ for which $z = 5$. We shall accordingly replace (4.8) by

$$(4.9)\qquad (0')\quad 3x_1-2x_2+5x_3 \geqslant 5\,,$$

[1] The constraints (3) and (4) in Eq. (4.7) are verified a priori if $x_i \leqslant 1$, $i = 1, 2, 3$. They were introduced with the instructional purpose of explaining a method intended for calculation by a computer, which does not usually effect an a priori simplification of this kind.

which will provide a more constricting condition and consequently an improved filter, continuing in the same manner if required.

Point	Constraints					Satisfies (4.7) and (4.8)	Value of z
	(0)	(1)	(2)	(3)	(4)		
[0,0,0]	0					no	
[0,0,1]	5	-1	1	0	1	yes	5
[0,1,0]	-2					no	
[0,1,1]	3	1	5			no	
[1,0,0]	3	1	1	'1	0	yes	3
[1,0,1]	8	0	2	1	1	yes	8
[1,1,0]	1					no	
[1,1,1]	6	2	6			no	

FIG. 4.15

Several other methods of a more or less sophisticated nature are available for reducing the number of operations, of which a few will be shown. But, before doing so, we wish to give an explanation.

Note

Let us suppose that an economic function is in the form

$$z = a_1 x_1 + a_2 x_2 + a_3 x_3 + a_4 x_4, \tag{4.10}$$

in which eacha_i is a real number that is either positive, negative, or null. Let us now arrange a_i in a nondecreasing set of values, for example, such as

$$a_1 \leqslant a_4 \leqslant a_2 \leqslant a_3.$$

(If some of the a terms were equal, their respective positions in the order would no longer be of importance.) It is now clear that if we perform the lexicographical enumeration in the order considered, namely,

$$\begin{aligned} [x_3, x_2, x_4, x_1] &= [0, 0, 0, 0] \\ &= [0, 0, 0, 1] \\ &= [0, 0, 1, 0] \\ &\dots\dots\dots\dots, \end{aligned} \tag{4.11}$$

we shall obtain the corresponding values of z, which will disclose almost the same order

$$0,\ a_1,\ a_4,\ a_4+a_1,\ a_2,\ a_2+a_1,\ a_2+a_4,\ a_2+a_4+a_1,\ \dots. \tag{4.12}$$

This will obviously not be a total order[1] (we could, for instance, have $a_4+a_1 > a_2$), but this total order will often be observed by many of the terms.

To show the advantages of this procedure let us turn to example (4.7), placing the variables in the order x_2, x_1, x_3, since $-2 < 3 < 5$. By incorporating (4.8) in the constraints and by arranging the coefficients of the economic function in nondecreasing order we then have

$$(4.13) \qquad \begin{array}{ll} (0) & -2x_2+3x_1+5x_3 \geqslant 3, \\ (1) & 2x_2+x_1-x_3 \leqslant 2, \\ (2) & 4x_2+x_1+x_3 \leqslant 4, \\ (3) & x_2+x_1 \leqslant 3, \\ (4) & 4x_2+x_3 \leqslant 6. \end{array}$$

Let us now form the table of enumeration (Fig. 4.16).

As will be observed, the values for the solutions greater than or equal to 3 have been obtained earlier and, by using a sequential method, the solution would usually be found even sooner.

Point $[x_2, x_1, x_3]$	Constraints					Satisfies (4.13)	Value of z
	(0)	(1)	(2)	(3)	(4)		
[0,0,0]	0					no	
[0,0,1]	5	-1	1	0	1	yes	5
[0,1,0]	3	1	1	1	0	yes	3
[0,1,1]	8	0	2	1	1	yes	8
[1,0,0]	-2					no	
[1,0,1]	3	1	5			no	
[1,1,0]	1					no	
[1,1,1]	6	2	6			no	

FIG. 4.16

7. Balas's Enumeration[2] Procedure [K25]

Let us examine Fig. 4.16 in which the variables x_i are arranged in the nondecreasing order of their coefficients in the economic function as x_2, x_1, x_3 in order to define the point that each of them occupies in that order.

[1] A total order is one in which all the elements can be arranged in relation to each other like real numbers:

$$A<B<C<\ldots<L<M<\ldots .$$

Mathematicians classify this as a strict total order.

[2] Balas's procedure is a method of enumeration that employs a filter constraint. Historically it is very important, since it demonstrated that algorithms that are effective for programs with integers could make use of implicit enumeration.

When we drew up this table we first calculated the constraint (0) of (4.13) for the point [0, 0, 0] and we then proceeded to the point lexicographically above it [0, 0, 1]. For the latter, all the constraints are satisfied, which provides a *yes* with the value $z = 5$. In this table we proceeded without taking account of this first result. In the procedure laid down by Balas, as soon as such a result is obtained the table is no longer used, and the corresponding new constraint is introduced.

Let us now continue with this modification in mind and observe how we link up our calculations in Balas's method, which can be summarized as follows:

a. Reclassify the variables to define the point in such a manner that these are in the nondecreasing order of their coefficients in the economic function (we are speaking of a maximum, and for a minimum the order would be inverse).

b. Form a table similar to 4.16 but cease the enumeration performed on it as soon as a point has been found that satisfies all the constraints, including the filter. Alternatively, cease when the enumeration is completed, which means that either the optimal solution has been found or that a solution does not exist.

c. If a better solution has been found, and if the enumeration has not been completed, the new filtering constraint (0′) is included in place of the old constraint (0). We then recommence from the lexicographical point above the one obtained in the preceding table, a point that corresponded with a value equal to or greater than the best solution discovered up to that stage. This procedure is then continued until the enumeration is completed.

Point $[x_2, x_1, x_3]$	Constraints					Satisfies (4.13)	Value of z
	(0)	(1)	(2)	(3)	(4)		
[0,0,0]	0					no	
[0,0,1]	5	−1	1	0	1	yes	5

FIG. 4.17

We have just obtained a solution corresponding to the value $z = 5$; hence we have a new filter constraint (0′):

$$(4.14) \qquad (0') \quad -2x_2+3x_1+5x_3 \geqslant 5\,.$$

Now let us construct the new table shown in Fig. 4.18. We enumerate another solution [0, 1, 0] and then find a possible solution [0, 1, 1] corresponding to an improved value of z, that is, $z = 8$.

Hence we now have a new filter constraint

$$(4.15) \qquad (0'') \quad -2x_2+3x_1+5x_3 \geqslant 8\,.$$

By the previous method of calculation we obtain the table of Fig. 4.19,

Point $[x_2,x_1,x_3]$	Constraints					Satisfies (4.13) and (4.14)	Value of z
	(0')	(1)	(2)	(3)	(4)		
[0,1,0]	3					no	
[0,1,1]	8	0	2	1	1	yes	8

FIG. 4.18

finding that no further improvement in z is possible and that $z = 8$ is the maximal value.

Let us now compare the number of additions and comparisons carried out by the method of complete enumeration, without using the filter constraint, with the number when Balas's procedure is used (Figs. 4.17–4.19).

Point $[x_2,x_1,x_3]$	Constraints					Satisfies (4.13) and (4.15)	Value of z
	(0")	(1)	(2)	(3)	(4)		
[1,0,0]	-2					no	
[1,0,1]	3					no	
[1,1,0]	1					no	
[1,1,1]	6					no	

FIG. 4.19

In the first case there are $8 \times 5 = 40$ calculations of linear functions; in the second case the number is 24.

With so few variables the gain is only a moderate one, but as their number increases the proportionate advantage of Balas's method is equally accentuated for the additions and for the comparisons, there being always at least as many of the former as of the latter.

8. The Procedure of Lemke and Spielberg

With this procedure [K59] we can still further reduce the number of additions and comparisons by introducing supplementary criteria of exclusion that permit the a priori exclusion of points that we should have been obliged to calculate with Balas's method.

In addition, in accordance with these two authors and also with Balas,[1] we can further apply a criterion that is one of preferential branching rather than of exclusion, but this only provides the hope, rather than the certainty, of reducing the number of operations in a heuristic manner.

In order to illustrate the procedure we shall use a didactic example. Let us

[1] Lemke and Spielberg as well as Balas made use, at about the same period, of these additional criteria that make it possible to restrict the number of operations to be performed.

examine a program with bivalent values:

$$\begin{array}{lll} & & [\mathrm{MIN}]z = 3x_1+7x_2-x_3+x_4\,, \\ & (1) & 2x_1-x_2+x_3-x_4 \geqslant 1, \\ (4.16) & (2) & x_1-x_2-6x_3+4x_4 \geqslant 8\,, \\ & (3) & 5x_1+3x_2+x_4 \geqslant 5\,, \\ & & x_1, x_2, x_3, x_4 = 0 \text{ or } 1. \end{array}$$

To be able to use the procedure of Lemke and Spielberg, the coefficients of the variables in the function of value must all be nonnegative. To effect this we transform them as follows: let us suppose

$$x_3' = 1-x_3\,.$$

In addition, so as to be able to apply this procedure, and also for reasons of convenience, we shall transform the inequalities bearing the sign *greater than* or *equal to* into inequalities carrying the sign *less than* or *equal to.* To obtain this result for (4.16) it will suffice to multiply the two members of the inequalities (1), (2), and (3) by -1. Finally, by introducing nonnegative deviation variables, we obtain z_1, z_2, and z_3, the new program that replaces (4.16) and that is its equivalent[1]:

$$\begin{array}{lll} & & [\mathrm{MIN}]z = -1+3x_1+7x_2+x_3'+x_4\,, \\ & (1) & -2x_1+x_2+x_3'+x_4+z_1 = 0\,, \\ (4.17) & (2) & -x_1+x_2+6x_3'-4x_4+z_2 = -2\,, \\ & (3) & -5x_1-3x_2-x_4+z_3 = -5\,, \\ & & x_1, x_2, x_3, x_4 = 0 \text{ or } 1, \quad z_1, z_2, z_3 \geqslant 0\,. \end{array}$$

We observe that z, z_1, z_2, z_3 can only assume integer values, since the coefficients of the program in z and in (1), (2), and (3) are integers and also because we have imposed the constraint that the variables x_i must be integers.

The sequential investigation will start, as in Balas's procedure, from the point where all the variables are null, namely the point $[x_1 = 0, x_2 = 0, x_3' = 0, x_4 = 0]$. But we shall no longer follow a lexicographical method such as that in (4.17)–(4.19) to pass from one point to another.

To begin with, let us define what we term a *step forward* and a *step backward* in the following explanations.

Let us suppose that in the course of the investigation we have arrived at

[1] This is an instructional example intended to explain a method. By simple observation it can be seen that (2) implies $x_4 = 1$ and that (3) implies $x_3 = 1$. The method shown here does not take those a priori evaluations into account.

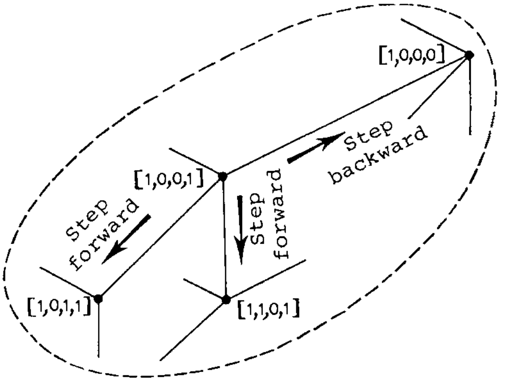


FIG. 4.20

the point $[x_1 = 1, x_2 = 0, x'_3 = 0, x_4 = 1]$ starting from the point $[x_1 = 1, x_2 = 0, x'_3 = 0, x_4 = 0]$. To take a step forward from the point [1, 0, 0, 1] is to go to one of the points of the higher level, at a Hamming's distance of 1, that is, to one of the points [1, 0, 1, 1] or [1, 1, 0, 1]. To take a step backward from the same point is to return to the preceding point, in this case [1, 0, 0, 0] (see Fig. 4.20). A step backward is required in the investigation when we have been able to decide by one of the criteria that will be defined shortly, that no solution can be obtained either by one or by several steps forward or that no better solution than the one already obtained can be found by such steps forward. In the contrary case one step forward is required.

Let us observe that as soon as a solution is obtained (for instance $[x_1 = 1, x_2 = 0, x'_3 = 0, x_4 = 1]$ in the case of (4.17)), a step forward would increase the value of the function z', that is, to $z = 3$, since in the transformed program (4.17) all the coefficients of the function z are positive. Hence, as soon as a solution has been found a step backward is required.

Lemke and Spielberg employ three criteria or tests to reduce the enumeration and we shall now proceed to explain them.

Projected Exclusion Test

"When we have found a solution we seek only solutions that will increase the value of z by at least 1."

Before proceeding further let us observe that in Balas's procedure we could, by contrast, discover several different solutions providing the same value for z.

To perform this projected exclusion test, we shall consider the four constraints, the first of which is obtained from the function of z to be minimized, by stating that we shall look only for points for which $z < 3$, namely,

$$(4.18) \qquad -1+3x_1+7x_2+x'_3+x_4 < 3,$$

and constraints (1), (2), and (3) of (4.19).

Let us transform (4.18) by adding a nonnegative deviation variable z_0 and by observing that <3 can be replaced by $\leqslant 2$, since only integer values are being considered.

Summing up, the points $[x_1, x_2, x_3', x_4]$ that may be solutions have to satisfy the four constraints:

$$
\begin{array}{lll}
 & (0) & 3x_1+7x_2+x_3'+x_4+z_0 = 3, \\
 & (1) & -2x_1+x_2+x_3'+x_4+z_1 = 0, \\
(4.19) & (2) & -x_1+x_2+6x_3'-4x_4+z_2 = -2, \\
 & (3) & -5x_1-3x_2-x_4+z_3 = -5. \\
 & & x_1, x_2, x_3', x_4 = 0 \text{ or } 1, \quad z_0, z_1, z_2, z_3 \geqslant 0.
\end{array}
$$

Starting with this example, we shall now explain the projected exclusion test. If, at one point, we have $z_0 \geqslant 0$, we can take a step forward provided that at least one of the variables is equal to 0. But, for each variable the coefficient of which strictly exceeds z_0, the step forward obtained by increasing the value of this variable from 0 to 1 is excluded. Indeed, this step forward would result in z_0 becoming strictly negative.

Let us illustrate this situation in our example by means of Fig. 4.21. For the point [1, 0, 0, 0] we have $z_0 = 0$. If we return from the point [1, 0, 0, 1], the two possible steps forward that remain are those that lead to one of the points [1, 1, 0, 0] or [1, 0, 1, 0]. But the coefficients of x_2 in (4.19), line (0), is 7, which is strictly greater than $z_0 = 0$. Hence we exclude the step toward [1, 1, 0, 0]. Similarly, by this same test we can exclude the step toward [1, 0, 0, 1]. Accordingly, since we have already come from [1, 0, 0, 1] and since it is impossible to go toward [1, 0, 1, 0] and [1, 1, 0, 0], we are obliged to take a step backward from [1, 0, 0, 0], returning thereby to the point [0, 0, 0, 0].

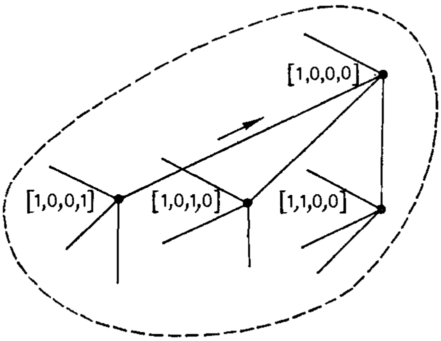


FIG. 4.21

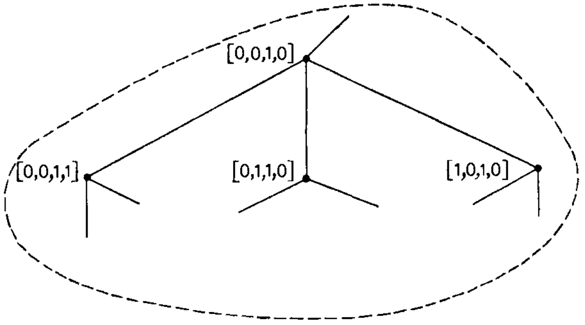


FIG. 4.22

Infeasibility Test

Let us suppose we are at the point [0, 0, 1, 0]. This point is not a solution, since it provides the value $z_2 = -8$ in (4.19). There is now a choice of three possible paths forward. But let us observe that in constraint (2) of Eq. (4.19) the sum of the coefficients of the null variables (in this case x_1, x_2, and x_4) of which the coefficients are negative (that is, the coefficients -1 and -4 of x_1 and x_4) is equal to -5. But -5 is greater than -8 and it will not be possible by taking a step forward from [0, 0, 1, 0] to find a solution. In fact, the highest value of z_2 to be obtained by steps forward from [0, 0, 1, 0] will be $z_2 = -8-(-5) = -3$, a negative number; and none of these points therefore provides a solution. Accordingly we exclude the points [0, 0, 1, 1], [0, 1, 1, 0], and [1, 0, 1, 0] by this *infeasibility test.* Since no step forward will provide a solution, a step backward is required and we return to [0, 0, 0, 0].

Preferred Variable Test

The two preceding tests were of a formal nature that provided a certain indication that no solution could be found by forward steps. By contrast, the preferred variable test provides a means of selecting, from several possible choices, the step forward that seems likely to prove best. It is, therefore, a *heuristic* test and will be used if the infeasibility test has not excluded all the steps forward. In this case, the possible forward steps are those not excluded by the projected exclusion test. We shall now explain the preferred variable test with the aid of example (4.17). Let us suppose that we are starting the sequential procedure at point [0, 0, 0, 0] to solve program (4.17). The infeasibility test does not yield anything (except as showing the absence of a solution, since [0, 0, 0, 0] is not a solution, and that any steps forward would not provide one). As no solution has yet been found the projected exclusion test does not apply, and four steps forward toward [0, 0, 0, 1], [0, 0, 1, 0], [0, 1, 0, 0], and [1, 0, 0, 0] are possible. At the point [0, 0, 0, 0] we have $z_1 = 0$, $z_2 = -2$, and $z_3 = -5$ and we now evaluate the total deviation from the sum of the deviation variables z_i that are negative, giving us $-2-5 = -7$.

Preference from among the possible steps forward will be given to the one (unique or not) that has the greatest tendency to reduce this deviation. To carry out this preferred variable test we shall calculate, in each line of (4.17) for which z_i is negative, the sum of the coefficients of the variables that correspond to possible steps forward, in this case the coefficients of lines (2) and (3). The preferred variable will be the one giving the minimum.

$$\begin{array}{ll}\text{For } x_1 \text{ we have} & -1-5 = -6, \\ \text{For } x_2 \text{ we have} & 1-3 = -2, \\ \text{For } x_3' \text{ we have} & 6+0 = 6, \\ \text{For } x_4 \text{ we have} & -4-1 = -5.\end{array} \tag{4.20}$$

Hence we shall begin by a step forward toward [1, 0, 0, 0], then toward [0, 0, 0, 1] if the projected exclusion test does not exclude it; next, toward [0, 1, 0, 0]. Finally, a step forward will be taken with the last preference toward [0, 0, 1, 0] (see Fig. 4.23). With such heuristic tests there is unfortunately no mathematical certainty that their use in a particular program will reduce the number of additions and comparisons; all that can be claimed is that they generally reduce the number of such operations. We are now able to apply the projected exclusion test that excludes a considerable number of steps forward and can proceed to solve the program, thanks to the procedure of Lemke and Spielberg. We denote by $\check{z}$ the best value of the economic function discovered in our sequential investigation and, at the start, ignore whether or not a solution exists. Since $\check{z}$ is an upper limit of the minimum of z, accordingly it possesses as great a value as we desire in order to begin our calculations.

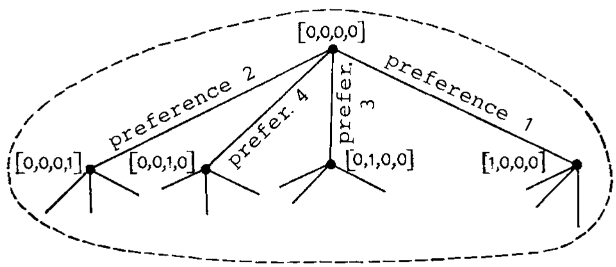


FIG. 4.23

Solving Program (4.17) *by Lemke's and Spielberg's Procedure*

We start from the point [0, 0, 0, 0] that is not a solution, since z_2 and z_3 are negative. Let us therefore apply the infeasibility test to constraints (2) and (3) in Eq. (4.17).

The sum of the negative coefficients of the null variables in (2) is equal to $-1-4 = -5$, which is less than $z_2 = -2$. Hence the infeasibility test for line

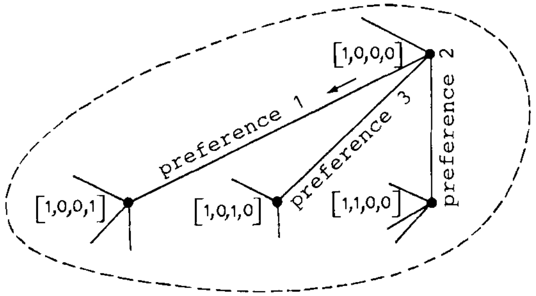


FIG. 4.24

(2) authorizes steps forward. Similarly for line (3) we find $-5-3-1 = -9$, which is less than $z_3 = -5$. Here, too, the infeasibility test yields nothing, and the projected exclusion test does not apply, since we have not yet obtained a solution. Four forward steps are possible. Calculating the preferred variable in (4.20) in the same way as before, we select the step forward toward [1, 0, 0, 0].

This point is not a solution, since $z_2 = -1$, so we now apply the infeasibility test to line (2) of Eq. (4.17). Of the null variables, x_4 alone has a negative coefficient, -4. Hence, with $-4 < -1$, the infeasibility test authorizes steps forward. As we have not yet obtained a solution, the exclusion test cannot be used, and we employ the preferred variable test to choose which of the three possible steps forward toward [1, 0, 0, 1], [1, 0, 1, 0], or [1, 1, 0, 0] should be taken. We now take the coefficients of the null variables in line (2) (in this case a single equation, whereas for [0, 0, 0, 0] lines (2) and (3) of (4.17) would give z_2 and $z_3 < 0$).

(4.21)
For x_2 we have 1;
for x_3' we have 6;
for x_4 we have -4.

The step forward to [1, 0, 0, 1] is the preferred one. The second preference leads to [1, 1, 0, 0] and the third to [1, 0, 1, 0] (see Fig. 4.24). The point [1, 0, 0, 0] is a first solution with a value for the economic function of $z = 3$. We do not make a further step forward from [1, 1, 0, 0], since the economic function could not diminish, the coefficients being nonnegative.

Accordingly we take a step backward, returning to [1, 0, 0, 0]. In addition, as in (4.18), we introduce a new filter constraint,

$$-1+3x_1+7x_2+x_3'+x_4 < 3. \tag{4.22}$$

It shows that we are only looking for those points that improve the economic function. Constraint (4.22) can be expressed as

$$(0) \quad 3x_1+7x_2+x_3'+x_4+z_0 = 3, \tag{4.23}$$

as we observed previously.

Back at the point [1, 0, 0, 0] after a step backward from [1, 0, 0, 1], which had first preference, we must still investigate the points that can be reached by steps forward to [1, 0, 1, 0] or [1, 1, 0, 0]. Since a solution has been found we can now employ the projected exclusion test. We have $z_0 = 0$, and the coefficients of x_2 and x'_3 in the economic function z, respectively, 7 and 1, are greater than z_0. The exclusion test therefore excludes the above steps. Since no step forward is possible we accordingly return [0, 0, 0, 0]. From this last point it remains for us to test our three remaining preferences (see Fig. 4.23). We have $z_0 = 3$. Hence the step forward to [0, 0, 0, 1], which was our second preference, is not excluded, since the coefficient of x_4 in the economic function is 1, which is less than $z_0 = 3$. We therefore take a step forward to this point, which is not, however, a solution since $z_3 = -4$, a negative number. The possible steps forward from [0, 0, 0, 1] are shown in Fig. 4.25; we have $z_0 = 2$. The step forward to [1, 0, 0, 1] is excluded since the coefficient of x_1 in function z is $3 > z_0$. Similarly, the step toward [0, 1, 0, 1] is excluded since $7 > z_0$. The step forward to [0, 0, 1, 1], on the other hand, is not excluded since $1 \leqslant z_0 = 2$. Point [0, 0, 1, 1] is not a solution since $z_1 = -2, z_2 = -4, z_3 = -4$.

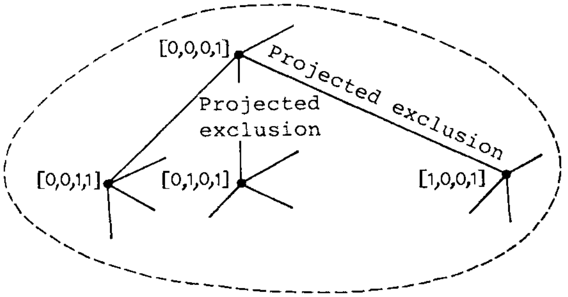


FIG. 4.25

We now employ the infeasibility test on the first three lines of (4.17) to discover whether one or more steps forward can provide a solution. In line (2) the sum of the negative coefficients of the null variables x_1, x_2 is -1, a number that is strictly greater than $z_2 = -4$. Hence a solution cannot be found by one or more steps forward from this point. We therefore take a step backward to [0, 0, 0, 1] (see Fig. 4.25). From this point any forward step is excluded and a step backward returns us to [0, 0, 0, 0]. At this point we still have to employ the third and fourth preferences. Here $z_0 = 3$. The projected exclusion test excludes a forward step for the third preference toward [0, 1, 0, 0], since the coefficient of x_2 in the z function is $7 > 3$ (see Fig. 4.23). On the other hand, the step forward for the fourth preference to [0, 0, 1, 0] is not excluded by this test. This point, however, is not a solution since $z_1 = -1$, $z_2 = -8$, $z_3 = -5$. We now perform the infeasibility test for the first three lines of (4.17). In (2) the sum of the negative coefficients of the null variables is $-5 > -8$. Since a step forward is not possible, we return to the point of origin, and since no other step forward is possible from this point, the procedure is concluded. Hence the optimum for program (4.17) is the best solution obtained, namely,

$$(4.24) \qquad x_1 = 1, \quad x_2 = 0, \quad x'_3 = 0, \quad x_4 = 1, \quad \text{with a value} \quad z = 3.$$

This point provides the optimal solution for the initial program (4.16):

$$(4.25) \qquad x_1 = 1, \quad x_2 = 0, \quad x_3 = 1, \quad x_4 = 1, \quad \text{with a value} \quad z = 3.$$

In Fig. 4.26 we have shown the movements carried out in accordance with the Lemke–Spielberg procedure and have indicated the tests that authorize the exclusion of a step forward. Using this procedure, which is more complicated than that of Balas, we have investigated six points out of a possible 16. With more extensive programs, the saving is considerably greater and the economy of the procedure correspondingly more important.

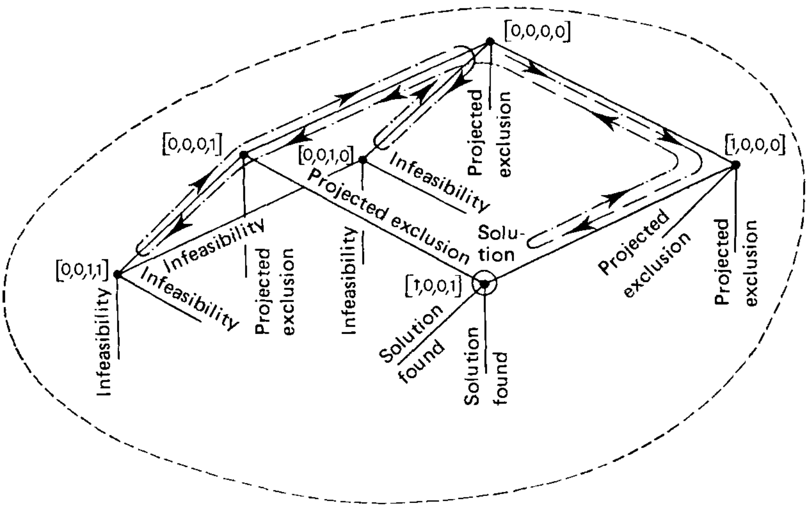


FIG. 4.26

Practical Arrangement of the Calculations for the Lemke–Spielberg Procedure

We begin by transforming program (4.16) to give it the form of program (4.17), which we resolve by the Lemke–Spielberg procedure. The solution of program (4.16) is obtained from that of program (4.17) by making $x_3 = 1 - x'_3$. We construct a table (Fig. 4.27) for convenience of calculation. In line (0) of the table we write the progressively decreasing values of $\check{z}$, the best value of the economic function discovered in the sequential investigation. In this example only one value appears, since only one solution was obtained. In the top left-hand corner we write -1 since the z function is expressed as

$$(4.26) \qquad z = -1 + 3x_1 + 7x_2 + x'_3 + x_4.$$

In line (1) of the table we insert the coefficients 3, 7, 1, 1 of the economic function, and then the successive values of z_0 obtained in the sequential investigation. In lines (2), (3), and (4) we insert the coefficients of lines (1), (2), and (3) of program (4.17). In these lines we also insert the values assumed by z_1, z_2, and z_3 for the different points in the sequential investigation. Thus, when we investigate point [0, 0, 0, 1] in line (6′), we find in the corresponding column the values of z_0, z_1, z_2, z_3 for this point, namely, -1, -2, -4, -4. In the first column of lines (0′)–(10′), which represents the end of the investigation, we give a list of the points investigated in the sequential procedure; this list gives the movements shown in Fig. 4.26. In the further columns of lines (0′)–(10′) we show the results of the tests performed in the following order:

Is the point a solution giving $z_1, z_2, z_3 \geqslant 0$?
Does the infeasibility test exclude every step forward?

For this purpose we calculate the sum of the coefficients in constraint i of the free variables (not marked by ☒), considering only those that are negative. If this sum is greater than $z_i < 0$, the point is not a solution and cannot become one by taking any step forward. If neither of these tests provides a result we perform the projected exclusion test on the free nonnull variables.

(0)		x_1	x_2	x'_3	x_4	z			3								
(1)	-1	3	7	1	1	z_0			-1	0	-3	-2	-1	-2	-3	-2	-3
(2)	Values of z_i	-2	1	1	1	z_1	0	2	1	2	0	-1	-2	-1	0	-1	0
(3)	List	-1	-1	6	-4	z_2	-2	-1	3	-1	-2	2	-4	2	-2	-8	-2
(4)	of points investigated	-5	-3	0	-1	z_3	-5	0	1	0	-5	-4	-4	-4	-5	0	-5
(0')	[0,0,0,0]	-⑥	-4	6	-5												
(1)	[1,0,0,0]		-1	6	-④												
(2')	[1,0,0,1]	Solution found															
(3')	[1,0,0,0]		PET	PET	☒												
(4')	[0,0,0,0]	☒	PET	6	-⑤												
(5')	[0,0,0,1]	PET	PET	⑥													
(6')	[0,0,1,1]	Infeasibility															
(7')	[0,0,0,1]	PET	PET	☒													
(8')	[0,0,0,0]	☒	PET	⑥	☒												
(9')	[0,0,1,0]	Infeasibility															
(10')	[0,0,0,0]	☒	PET	☒	☒												

FIG. 4.27

If the test excludes the corresponding step forward it is indicated by PET in the column corresponding to the variable. Thus, in line (5′) the steps toward [1, 0, 0, 1] and [0, 1, 0, 1] are excluded by the projected exclusion test.

In the case of the null variables that are not excluded, we insert in this line the sum of the coefficients of lines (2), (3), or (4) for which z_1, z_2, or z_3 are negative. This enables us to perform the preferred variable test by taking the minimum of this sum, this being shown in a circle. Line (4′) corresponds to the point of origin. We have returned to this point by a step backward from the preceding point in the list, namely, [1, 0, 0, 0], and the step forward to [1, 0, 0, 0] is now excluded, as indicated by the sign ☒. We have $z_2 = -2$ and $z_3 = -5$; the sum of the coefficients of x_3 in lines (3) and (4) is 6 and that of x_4 is -5. Since -5 is smaller than 6 we prefer the variable x_4 and circle -5.

With each step forward we fill in a new line and a new column. In the column we calculate the z_i values; in the line we insert the result of the tests. For each step backward we also fill in a new line and a new column. Since the point already appears in the list we look for it in order to insert the same values of z_1, z_2, z_3 in the new column. However z_0 must be recalculated because in the meanwhile a solution may have been found. We carry into the new line the signs PET or ☒ of the preceding line in which the same point appears. A further sign ☒ signifying *already investigated* is inserted in the column of the variable that has been reduced to zero by the step backward.

The calculations are concluded when we return to the point of origin and when any forward step is excluded because in that line the columns of the four variables bear the sign PET or ☒.

To be sure, we are dealing here with a purely instructive example accompanied by the lengthy and painstaking explanations that were requisite. Indeed, in a case of simple enumeration the procedure would have proved a great deal shorter. But these procedures are intended for programs with integers in which a comparatively large number of variables is encountered and for which treatment by a computer is needed.

Section 5. **Some More Complicated Examples of Problems with Integer Values**

1. A Simple Example to Show the Pitfalls of Rounding Off

Before considering concrete problems we wish again to emphasize the fact that the optimal solution of a linear program with integer solutions can be very different from the optimal solution of the same program when integer solutions are not mandatory. Let us consider the following linear program for which integer solutions are required:

$$(5.1)\quad \begin{array}{l} [\mathrm{MAX}]\ z = 4x_1+x_2, \\ (1)\quad -3/7x_1+x_2 \leqslant 1, \\ (2)\quad x_1-x_2 \leqslant 1/3, \\ (3)\quad x_1,\ x_2 \geqslant 0 \text{ and integers.} \end{array}$$

A geometrical solution is shown in Fig. 5.1. If we ignore the constraint that x_1 and x_2 must be integers, we find as a solution $x_1 = \frac{7}{3}$ and $x_2 = 2$ with $z = 11\frac{1}{3}$. It is evident from the figure (it would be sufficient to calculate the value of z for each point marked with a ∘ and such that $0 \leqslant x_1 \leqslant 2$, $0 \leqslant x_2 \leqslant 2$), that the maximum for z, given the constraint that x_1 and x_2 are to be integers, occurs with $x_1 = 1$ and $x_2 = 1$, namely, $z = 5$. Hence the optimum for the associated linear program gives $11\frac{1}{3}/5 = 2\frac{4}{15}$ times (or 226%) the result of the program with integer values. Equally, if we round off each x_1 to the nearest whole number above its value we find $x_1 = 3$ and $x_2 = 2$, which no longer satisfies the constraints and would give $z = 14$ or 280% of the correct result.

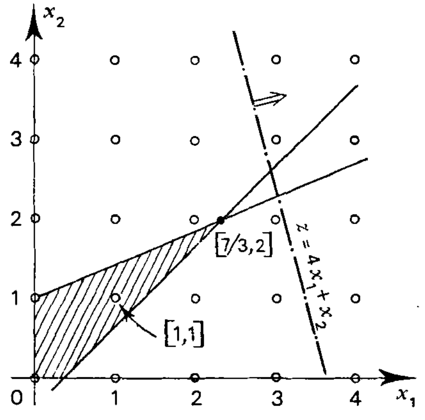


FIG. 5.1

2. Some Concrete Examples. An Interesting Agricultural Problem[1]

This example will be of special interest to the reader because of the detailed method by which the constraints and the economic function are constructed.

[1] This example is an adaptation and a variation of the one given by P. L. Hammer, and S. Rudeanu, "Boolean Methods in Operations Research and Related Areas," Springer Publ., New York, 1970.

We are again dealing with a problem in which the solutions are constituted by the values 0 or 1 of the variables; but this problem assumes the form of a nonlinear program.

With the exception of a few details, the practical programs for the selection of crops appear in the form of the fairly general model described below.

An agricultural estate contains m lots, $L_1, L_2, \ldots, L_m$. In these lots n types of crop can be grown, $C_1, C_2, \ldots, C_n$, with $m > n$, but in each lot only one crop can be grown. When a crop C_j is grown in a lot L_i, a total expense of d_{ij} is incurred, but supplementary work may be carried out for which the cost is C_{ij} for crop C_j in lot L_i. A fertilizer, but only one, may also be used, selected from r types of product, $F_1, F_2, \ldots, F_r$, to fertilize lot L_i. The supplementary cost of using product F_k in lot L_i, in which C_j is grown, is b_{ijk}. Finally, each lot may or may not be irrigated. The cost of the irrigation of lot L_i will be a_i and does not vary according to the crop.

We can now define the quantities α_{ijk}^{00}, α_{ijk}^{01}, α_{ijk}^{10}, and α_{ijk}^{11} in the following manner: α_{ijk}^{00} is the average harvest when crop C_j is grown on lot L_i using fertilizer F_k, but without additional work or irrigation; α_{ijk}^{01} represents the same case with irrigation added; α_{ijk}^{01} represents the first case with additional work; finally α_{ijk}^{11} represents the first case with additional work and irrigation.

With π_j representing the total average harvest from crop C_j, we set a production target p_j for each crop and impose the restriction

$$\pi_j \geqslant p_j, \qquad j = 1, 2, \ldots, n. \tag{5.2}$$

Let us define the following bivalent variables:

$$\begin{aligned} x_{ij} &= 1, \quad \text{if crop } C_j \text{ is grown on lot } L_i, \\ &\qquad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n, \\ &= 0, \quad \text{in the contrary case.} \end{aligned} \tag{5.3}$$

$$\begin{aligned} y_{ik} &= 1, \quad \text{if fertilizer } F_k \text{ is used on lot } L_i, \\ &\qquad i = 1, 2, \ldots, m; \quad k = 1, 2, \ldots, r, \\ &= 0, \quad \text{in the contrary case.} \end{aligned} \tag{5.4}$$

$$\begin{aligned} z_i &= 1, \quad \text{if additional work is needed on lot } L_i, \\ &\qquad i = 1, 2, \ldots, m, \\ &= 0, \quad \text{in the contrary case.} \end{aligned} \tag{5.5}$$

$$\begin{aligned} t_i &= 1, \quad \text{if lot } L_i \text{ is irrigated,} \\ &= 0, \quad \text{in the contrary case.} \end{aligned} \tag{5.6}$$

Let us now consider how to construct the model that is, because of the nature of the problem, somewhat complicated.

In the first place we impose the restriction that in a lot L_i there can only be one crop C_j. Thus for lot L_1 we write

$$(5.7) \qquad x_{11}+x_{12}+\ldots+x_{1n} \leqslant 1\,.$$

The sign *less than* or *equal to* indicates that we cannot cultivate L_1. For lot L_2 we write

$$(5.8) \qquad x_{21}+x_{22}+\ldots+x_{2n} \leqslant 1\,.$$

And similarly for each of the lots. By grouping the results we are finally able to write m equations

$$(5.9) \qquad x_{i1}+x_{i2}+\ldots+x_{in} \leqslant 1\,, \qquad i = 1, 2, \ldots, m\,.$$

Similarly for the variables y_{ik} we write

$$(5.10) \qquad y_{i1}+y_{i2}+\ldots+y_{ir} = 1\,, \qquad i = 1, 2, \ldots, m\,,$$

since we only employ one kind of fertilizer for each lot.

The relation (5.2) shows that the production must be greater than or equal to a target set in advance for each crop. To determine the expression of π_j it is convenient to introduce the following subsidiary variables:

$$(5.11) \qquad \bar{y}_{ik} = 1-y_{ik}\,,$$

$$(5.12) \qquad \bar{z}_i = 1-z_i\,,$$

$$(5.13) \qquad \bar{t}_i = 1-t_i\,.$$

The condition (5.2) can now be expressed as

$$(5.14) \qquad \pi_j = \sum_{i=1}^{m} x_{ij} \sum_{k=1}^{r} y_{ik}(\alpha_{ijk}^{11}.z_i t_i + \alpha_{ijk}^{10}.z_i \bar{t}_i + \alpha_{ijk}^{01}.\bar{z}_i t_i + \alpha_{ijk}^{00}.\bar{z}_i \bar{t}_i) \geqslant p_j\,, \qquad j = 1, 2, \ldots, n\,.$$

Using W to represent the total cost of production and taking this total cost as the economic function to be optimized, we have

$$(5.15) \qquad W = \underbrace{\sum_{i=1}^{m} a_i t_i}_{\text{irrigation}} + \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}(\underbrace{d_{ij}}_{\text{initial cost}} + \underbrace{c_{ij} z_i}_{\text{works}} + \sum_{k=1}^{r} \underbrace{b_{ijk} y_{ik}}_{\text{fertilizer}})\,.$$

Finally the model will be constituted by the following program with bivalent variables

$$(5.16) \qquad [\mathrm{MIN}]\, W = \sum_{i=1}^{m} a_i t_i + \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}(d_{ij} + c_{ij} z_i + \sum_{k=1}^{r} b_{ijk} y_{ik})\,,$$

$$(5.17)\qquad \sum_{i=1}^{m} x_{ij} \sum_{k=1}^{r} y_{ik}(\alpha_{ijk}^{11}.z_i t_i + \alpha_{ijk}^{10}.z_i \bar{t}_i + \alpha_{ijk}^{01}.\bar{z}_i t_i + \alpha_{ijk}^{00}.\bar{z}_i \bar{t}_i) \geqslant p_j\,, \quad j = 1, 2, \ldots, n\,,$$

$$(5.18)\qquad \sum_{j=1}^{n} x_{ij} \leqslant 1, \qquad i = 1, 2, \ldots, m\,,$$

$$(5.19)\qquad \sum_{k=1}^{r} y_{ik} = 1, \qquad i = 1, 2, \ldots, m\,,$$

in which all the variables x_{ij}, y_{ik}, z_i, t_i can only assume the values of 0 and 1, and in which the variables $\bar{y}_{ik}$, $\bar{z}_i$, and $\bar{t}_i$ are defined by (5.11)–(5.13).

We might equally use other criteria to solve the problem. If we allocated a profit v_j to each unit of production C_j we could then optimize the total value V of the crops, namely,

$$(5.20)\qquad V = \sum_{j=1}^{n} v_j.\pi_j\,,$$

where π_j is expressed by (5.14). In this case we should aim at finding a solution such that the total cost W given by (5.15) is lower than or equal to a threshold W_0. We should then obtain the program

$$(5.21)\qquad [\text{MAX}]\ V = \sum_{j=1}^{n} v_j.\left[\sum_{i=1}^{m} x_{ij} \sum_{k=1}^{r} y_{ik}(\alpha_{ijk}^{11}.\,z_i t_i + \alpha_{ijk}^{10}.\,z_i \bar{t}_i + \alpha_{ijk}^{01}.\,\bar{z}_i t_i + \alpha_{ijk}^{00}.\,\bar{z}_i \bar{t}_i)\right]\,,$$

$$(5.22)\qquad \sum_{i=1}^{m} a_i t_i + \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}\left(d_{ij} + c_{ij} z_i + \sum_{k=1}^{r} b_{ijk} y_{ik}\right) \leqslant W_0\,,$$

$$(5.23)\qquad \sum_{j=1}^{n} x_{ij} \leqslant 1, \qquad i = 1, 2, \ldots, m,$$

$$(5.24)\qquad \sum_{k=1}^{r} y_{ik} = 1\,, \qquad i = 1, 2, \ldots, m\,.$$

$$(5.25)\qquad x_{ij}, y_{ik}, z_i, t_i = 0 \text{ or } 1\,, \qquad i = 1, 2, \ldots, m, j = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, r.$$

Other criteria, too, could be used: for instance, minimizing the proportion V/W with constraints (5.17)–(5.19), maximizing this proportion with constraints (5.22)–(5.25), or introducing a constraint $V \geqslant V_0$. To be sure, with different economic functions, the optimal solutions are usually different.

This initial example, easy enough to understand but complicated in its expression, has only been offered to the reader with the aim of demonstrating the method of reasoning and of constructing models for programs with bivalent variables. Should he feel intimidated by these somewhat complicated summations we suggest that he proceed to expand them taking $n = 3$, $m = 2$, and so forth.

3. A Fresh Examination of the Problem of the Traveling Salesman

In Volume 1 (page 72) and in Volume 2 (page 286) we have already considered this problem that is so well known in mathematical treatises on combinatorial problems, particularly those connected with the theory of graphs. The theoretical solution of this celebrated problem depends on the programing of bivalent variables, as we propose to recall. Nevertheless, the concrete solution by the use of computers is obtained by means of a special method termed "branch and bound" by the Anglo-Saxons and "separation and progressive evaluation" by the French (for this method the reader is referred to [K18]).

It is useful, from a methodological standpoint, to explain how this problem of optimization can be transformed into a program of bivalent variables.

For this purpose, let us consider a complete symmetrical graph of type n, namely, a graph with n vertices, in which each pair of vertices (X_i, X_j) is connected by an arc having a value $C_{ij} \geqslant 0$. The problem of the traveling salesman may now be enunciated in the following manner:

To find the Hamiltonian circuit (or circuits) such that the total value of the arcs composing it is minimal.

This problem can be transformed into a program with bivalent variables of different states.[1]

Let us arbitrarily select a vertex of the circuit, for instance X_1, and let us introduce the following variables with three indices:

(5.26) $X_{ijr} = 1$, if the rth arc ($r = 1, 2, \ldots, n$) of the circuit starting from X is the arc (X_i, X_j),

$= 0$, in the contrary case.

With this convention we still have:

(5.27) $$x_{ij1} = 0, \qquad i \neq 1,$$

and

(5.28) $$x_{ijn} = 0, \qquad j \neq 1.$$

Thus in Fig. 5.2, which represents a complete graph containing six vertices,

[1] This concept is due to G. B. Dantzig, see [K8].

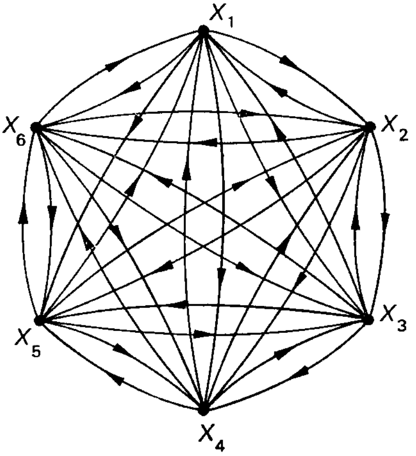


FIG. 5.2

let us consider the circuit $(X_1, X_3, X_4, X_5, X_2, X_6, X_1)$. For this circuit we have

	Arc	*Value of r*
	(X_1, X_3)	1
	(X_3, X_4)	2
(5.29)	(X_4, X_5)	3
	$(X_5\ X_2)$	4
	$(X_2\ X_6)$	5
	(X_6, X_1)	6

Hence we have, for instance, $x_{1,3,1} = 1$, $x_{3,4,2} = 1$, ..., $x_{6,1,6} = 1$, all the other X_{ijr} being null.

In our example there are $6 \times 6 \times 6 = 216$ variables x_{ijr}; a solution will include six variables equal to 1 and 210 equal to 0. For n vertices there would be n^3 variables of which n would be equal to 1 and $n^3 - n = (n-1)n(n+1)$ would be equal to 0.

To express the program with bivalent variables, let us introduce the constraints that ensure that the solution obtained will be a Hamiltonian circuit.

Let us indicate that if we arrive at vertex $X_j \neq X_1$ after traversing r arcs from the vertex of origin, we leave vertex X_j when we traverse the $(r+1)$th arc. This can be expressed as

$$(5.30) \qquad \sum_{i=1}^{n} x_{ijr} = \sum_{k=1}^{n} x_{jk,r+1}, \qquad \begin{array}{l} r = 1, 2, \ldots, n-1, \\ j = 2, 3, \ldots, n. \end{array}$$

Given that this constraint does not apply for $j = 1$, we leave vertex X_1 by the first arc numbered $r = 1$ and we return to it by the arc numbered $r = n$. It can easily be seen that a solution containing two unconnected circuits would not satisfy this constraint.

Finally, let us clearly specify that we leave vertex X_i by one and only one arc:

$$(5.31) \qquad \sum_{j=1}^{n} \sum_{r=1}^{n} x_{ijr} = 1, \qquad i = 1, 2, \ldots, n.$$

Let us now define the economic function:

$$(5.32) \qquad [\text{MIN}]\ z = \sum_{r=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} c_{ij} . x_{ijr} .$$

Finally the program can be written

$$(5.33) \qquad [\text{MIN}]\ z = \sum_{r=1}^{n} \sum_{j=1}^{n} \sum_{i=1}^{n} c_{ij} . x_{ijr},$$

$$(5.34) \qquad \sum_{i=1}^{n} x_{ijr} = \sum_{k=1}^{n} x_{jk,\, r+1}, \qquad \begin{array}{l} r = 1, 2, \ldots, n-1, \\ j = 2, 3, \ldots, n. \end{array}$$

$$(5.35) \qquad \sum_{j=1}^{n} \sum_{r=1}^{n} x_{ijr} = 1, \qquad i = 1, 2, \ldots, n,$$

$$(5.36) \qquad x_{ijr} = 0 \text{ or } 1.$$

In this form, first formulated by M. M. Flood in 1956, the program in bivalent variables for the problem of the traveling salesman includes n^3 variables and $2n^3$ constraints. Thus, for the limited case shown in Fig. 5.2, where there are only six vertices, it would be necessary to solve a program with $6 \times 6 \times 6 = 216$ variables and 72 constraints. The enumeration for the $5.4.3.2 = 120$ circuits would be appreciably quicker. Indeed, the formulation shown in (5.32)–(5.36) is of purely academic interest.[1]

In 1960 another model for this problem containing fewer constraints and variables than the preceding model was given by A. W. Tucker, and we now propose to examine it.

To do so, let us consider a complete symmetrical graph with $n+1$ vertices $X_0, X_1, X_2, \ldots, X_n$ and try to discover the Hamiltonian circuit with a total minimal value, with its origin at X_0. Let x_{ij} be a bivalent variable such that its value is 1 if the circuit passes along arc (X_i, X_j) and 0 in the contrary case, $i, j = 0, 1, 2, \ldots, n$. The program is now developed as follows, the quantities

[1] In Section 23 (page 378) we show Trubin's method, which is very effective when applied to a problem of this kind.

u_i being defined later:

$$\text{(5.37)} \qquad [\text{MIN}]\ z = \sum_{i=0}^{n} \sum_{j=0}^{n} c_{ij} . x_{ij} ,$$

$$\text{(5.38)} \qquad \sum_{i=0}^{n} x_{ij} = 1, \qquad j = 1, 2, \ldots, n,$$

$$\text{(5.39)} \qquad \sum_{j=0}^{n} x_{ij} = 1, \qquad i = 1, 2, \ldots, n,$$

$$\text{(5.40)} \qquad u_i - u_j + n x_{ij} \leqslant n-1 \qquad (1 \leqslant i \neq j \leqslant n),$$

$$\text{(5.41)} \qquad x_{ij} = 0 \text{ or } 1, \qquad i, j = 1, 2, \ldots, n,$$

$$u_i \in \mathbf{R} \text{ (real numbers)} \qquad i = 1, 2, \ldots, n.$$

Any solution that satisfies (5.38) and (5.39) and that is a Hamiltonian circuit will satisfy (5.40), and reciprocally. When (5.40) is not satisfied the solution contains at least two elementary circuits with a number of arcs $k \leqslant n$. In fact, if we add up all the inequalities (5.40) corresponding to $x_{ij} = 1$ for the arcs (X_i, X_j) that belong to an elementary and non-Hamiltonian circuit and that do not pass through vertex X_0, we annul the differences $u_i - u_j$ and obtain $nk \leqslant (n-1)k$; this is impossible, whence the contradiction between the hypothesis and the conclusion. We need only demonstrate that, for each Hamiltonian circuit starting from X_0, it is possible to find a value u_i that satisfies (5.40). Let us choose $u_i = r$ if vertex X_i is the terminal extremity of the rth arc when we traverse the path that runs from X_0 to X_i, $r = 1, 2, \ldots, n$. It is evident that $u_i - u_j \leqslant n-1$ is satisfied for every arc (X_i, X_j). Hence the conditions are satisfied for all the $x_{ij} = 0$ and for $x_{ij} = 1$, and we have

$$\text{(5.42)} \qquad u_i - u_j + n x_{ij} = r - (r+1) + n = n-1 .$$

Using this model, for n vertices of the graph we have n^2 bivalent variables and $2(n-1)+(n-1)^2 = n^2-1$ constraints instead of n^3 and $2n^2$, respectively. Hence in the small example of Fig. 5.1 there are 36 bivalent variables and 35 constraints, which is an appreciable reduction. Nevertheless the number of constraints and of variables quickly increases with n^2 as n grows larger.

At the present time the branch and bound method is favored for the problem of the traveling salesman. But the method of Gomory, which will be explained in Section 19, also produces good results for this problem.[1]

4. Planning the Work of Teachers and Law Courts

This type of problem occurs in every school where there are a number of classrooms and a number of teachers for the pupils. It has numerous variations, one of which has been selected as an example.

[1] See, for example, G. T. Martin, Solving Traveling Salesman Problem by Integer Linear Programming, Control Data Corp., New York, May 1966.

Let there be M groups of pupils, each group constituting a class, P teachers, and S classrooms. Let us also introduce the following bivalent variables:

$$(5.43) \qquad \begin{aligned} x_{ijkt} &= 1, && \text{when teacher } j \text{ takes a class with group } i \text{ in classroom } k \text{ on date } t, \\ &= 0, && \text{in the contrary case.} \end{aligned}$$

$$i = 1, 2, \ldots, M; j = 1, 2, \ldots, P; k = 1, 2, \ldots, S; t = 1, 2, \ldots .$$

The work week is presumed to last for only q days ($1 \leqslant q \leqslant 6$) from Monday to Saturday. The daily schedule contains a maximum of h hours study. Each period may last one hour or two hours, the unit being an hour. Various constraints exist with respect to the use of classrooms for a group i and a teacher j; for instance, a physical training instructor cannot, in principle, teach in a room lacking the equipment that he needs. For a group i and a teacher j we provide a vector,

$$(5.44) \qquad [O_{ij}] = [O_{ij1}, O_{ij2}, \ldots, O_{ijS}],$$

of which the elements O_{ijk} have a value of 1 if room k can be used by group i for the lesson of teacher j and have a value of 0 in the contrary case. In addition, for every teacher j, a vector exists,

$$(5.45) \qquad [d_j] = [d_{j1}, d_{j2}, \ldots, d_{j,qh}],$$

of which the elements d_{jt} have a value of 1 if teacher j is available at hour t and a value of 0 in the contrary case. In addition, let us specify the significance of qh as the index of $d_{j,qh}$ in (5.45).

What is meant by hour t is the position of an hour among the working hours of the week. Hence a week contains qh periods of one hour resulting from the h periods of the work day.

Let us now introduce various constraints representing the actual conditions under which a school operates.

At hour t a teacher who is available can only teach a single group of pupils or not teach one:

$$(5.46) \qquad \sum_{i=1}^{M} \sum_{k=1}^{S} x_{ijkt} \leqslant d_{jt}, \qquad \begin{aligned} j &= 1, 2, \ldots, P, \\ t &= 1, 2, \ldots, qh. \end{aligned}$$

At hour t a group i can only attend one course:

$$(5.47) \qquad \sum_{j=1}^{P} \sum_{i=1}^{S} x_{ijkt} \leqslant 1, \qquad \begin{aligned} i &= 1, 2, \ldots, M, \\ t &= 1, 2, \ldots, qh. \end{aligned}$$

At a given hour t a classroom k can be used only by a group i attending a course given by teacher j:

$$(5.48) \qquad \sum_{i=1}^{M} \sum_{j=1}^{P} x_{ijkt} \leqslant 1, \qquad \begin{aligned} k &= 1, 2, \ldots, S, \\ t &= 1, 2, \ldots, qh. \end{aligned}$$

With respect to the possible occupation of a room k by a group i with a teacher j, we have

$$(5.49) \qquad \sum_{t=1}^{qh} x_{ijkt} \leqslant qh \,.\, O_{ijk} \,, \qquad \begin{aligned} i &= 1, 2, \dots, M, \\ j &= 1, 2, \dots, P, \\ k &= 1, 2, \dots, S. \end{aligned}$$

Finally we can obtain a matrix $[C_{ij}]$ giving the number of hours for the course of teacher j with group i. We should have

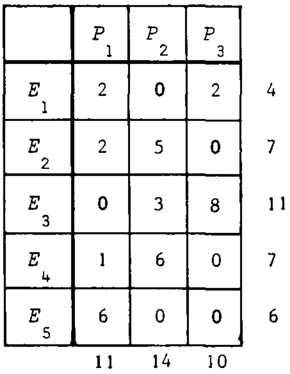
$$(5.50) \qquad \sum_{k=1}^{S} \sum_{t=1}^{qh} x_{ijkt} = C_{ij} \,, \qquad \begin{aligned} i &= 1, 2, \dots, M, \\ j &= 1, 2, \dots, P. \end{aligned}$$

In some schools there is a restriction that a class never spends more than two hours with the same teacher or in studying the same subject. A mathematical constraint would then be added.

Of course many other constraints might be added, for instance, that any teacher is limited to λ hours teaching in one day; equally, that a group of pupils must not study the same subject for more than μ hours in a day (namely, with the same teacher), that certain subjects should be excluded on Mondays, since it is a well-known fact of modern life that the so-called Sunday day of rest is the most exhausting day of the week, especially for the young

Usually the planning of the occupation of the classrooms is not optimized; we are content with one or more solutions that satisfy the constraints. However we can imagine various criteria, each of which would correspond to a particular need: arranging the largest possible number of lessons at certain times, separating exhausting lessons, arranging for the maximum number of lessons to last two hours (which might suit the consensus of teachers) or on the contrary, to last only one hour (if such should be the desire of the teachers). Let it be understood also that optimization cannot take place for several criteria unless, of course, these can be merged, which rarely happens.

Let us now take a numerical example that should serve to remove the



	P_1	P_2	P_3	
E_1	2	0	2	4
E_2	2	5	0	7
E_3	0	3	8	11
E_4	1	6	0	7
E_5	6	0	0	6
	11	14	10	

FIG. 5.3

theoretical aspect somewhat from our exposition; it has been expressly made as simple as possible, and too much importance should not be attached to the numbers in the data, which have, by intention, been fairly widely dispersed.

We imagine that there are five groups of pupils E_1, E_2, E_3, E_4, and E_5 and three teachers P_1, P_2, and P_3. The length of each teacher's course with each group of pupils is given in hours in Fig. 5.2. In our example it is to be understood that there are four hours study in a day and six days of study in a week. Hence, in accordance with the general enunciation given above, we have

$$(5.51) \qquad M = S, \quad P = 3, \quad S = 3, h = 4, \quad q = 6.$$

The obligations with respect to the classrooms are given in Fig. 5.4 where

O_{ijk}	S_1	S_2	S_3
E_1P_1	1	1	0
E_1P_2	0	0	0
E_1P_3	0	1	0
E_2P_1	1	1	0
E_2P_2	1	0	0
E_2P_3	0	0	0
E_3P_1	0	0	0
E_3P_2	1	0	1
E_3P_3	0	1	1
E_4P_1	1	1	0
E_4P_2	1	0	1
E_4P_3	0	0	0
E_5P_1	1	1	0
E_5P_2	0	0	0
E_5P_3	0	0	0

FIG. 5.4

	Monday				Tuesday				Wednesday				Thursday				Friday				Saturday			
$[d_j]$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
P_1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
P_2	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
P_3	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

FIG. 5.5

Number of variables not identically null

i, j \ t	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
1,1	2	2	2	2	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	32
1,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1,3	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	12
2,1	2	2	2	2	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	32
2,2	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	16
2,3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3,1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3,2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	2	2	2	2	32
3,3	0	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	24
4,1	2	2	2	2	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	32
4,2	2	2	2	2	2	2	2	2	2	2	2	2	0	0	0	0	0	0	0	0	2	2	2	2	32
4,3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5,1	2	2	2	2	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	2	2	2	2	32
5,2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5,3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FIG. 5.6

each line corresponds to a vector $[O_{ij}]$. The hourly availability of the teachers is given in Fig. 5.5 in which each line corresponds to a vector $[d_j]$.

A possible solution is shown in Fig. 5.6. In this example it is easy to find a solution given the availability of a place with the classes; this would not be the case if the number of teachers were increased without a proportionate increase in the number of classrooms.

It is instructive in this example to discover the number of variables x_{ijkt} not identically null. There are, to begin with, $5 \times 3 \times 3 \times 24 = 1080$ variables x_{ijkt}. Those that are, a priori, identically null in accordance with Fig. 5.4 number $4 \times 6 \times 29 = 696$. But if we take Fig. 5.5 (the availability of teachers) into account we can then construct Fig. 5.6 that gives the daily number of variables that are not identically null, namely, 244. This can help us to draw up a planning arrangement.

However, concrete problems of this type are, as we may well suspect, much larger in scope, given the number of teachers, of groups, and of classrooms. It then becomes necessary to divide the problem into sections and to employ heuristic[1] methods of a somewhat elaborate nature in the absence of analytic solutions leading to optimization. Figure 5.7 shows a solution that can be improved in relation to a selected criterion and, if it is advisable, in relation to several criteria.

[1] A heuristic procedure is constituted by a set of rules that are intuitive and partial selections, enabling a solution and, in certain cases, a better solution to be obtained.

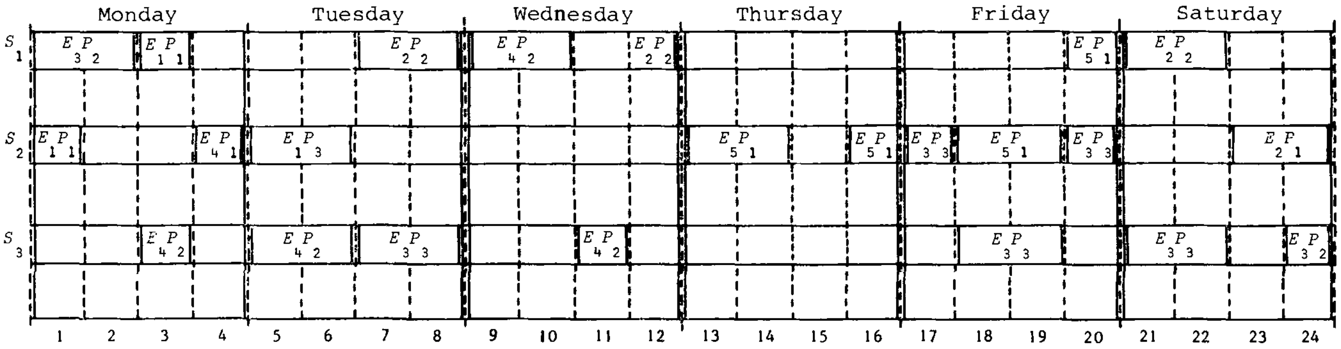

Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
S_1
$E_3 P_2$
$E_1 P_1$
$E_2 P_2$
$E_4 P_2$
$E_2 P_2$
$E_5 P_1$
$E_2 P_2$
S_2
$E_1 P_1$
$E_4 P_1$
$E_1 P_3$
$E_5 P_1$
$E_5 P_1$
$E_3 P_3$
$E_5 P_1$
$E_3 P_3$
$E_2 P_1$
S_3
$E_4 P_2$
$E_4 P_2$
$E_3 P_3$
$E_4 P_2$
$E_3 P_3$
$E_3 P_3$
$E_3 P_2$
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24

FIG. 5.7

5. The Problem of Selecting Masks in an Integrated Circuit

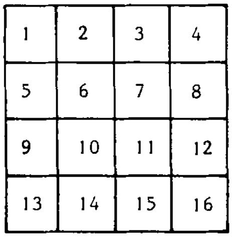
The methods of operations research can be used for problems far removed from econometrics. The example given earlier in this section about the removal of tiles showed that the plumbers' union could more advantageously apply to operational researchers than to the medical profession to prevent back strain! The problem that we now present is concerned with the purely technological problem of choosing the masks used for the production, by a vacuum or diffusion process of deposit, of the integrated circuits used in the most advanced machines for sifting information. The model and the special algorithm for its solution that are given in Part 2 (see [K49] and [K55]) are here being used in a practical context.

An integrated circuit is manufactured in a plate (perhaps square) composed of silicon or germanion with its sides some millimeters in length. This plate is itself divided into several dozen elementary cells which, after deposits of metallic connections and the diffusion of impurities, allowing for the formation of diodes and transistors, have special logical functions. Let us, however, ignore the technological aspects of electronics and concern ourselves with the elementary cells that are situated on these small plates of integrated circuits. Our aim is to minimize the number of defective cells produced by the delicate process of manufacture.

In Fig. 5.8 we show a plate with 16 numbered cells, though in practice there are far more; 8 × 8 = 64 or even 32 × 32 = 1024. These plates may constitute a complete element of memory for a bit or a more complex information entity (octet, word) with its access circuits.

These integrated circuits are manufactured by the use of *masks*, metal plates in which very delicate incisions are cut. These masks act as stencils that enable various deposits or diffusions to flow in the same manner as in the reproduction of a painting, and they are used in rotation. If there is a defect in one of them in the area corresponding to a cell, this cell will in turn be defective and will require further treatment that it is desired to avoid.

The *sum of the defects* is represented by Fig. 5.9, where the integrated circuit contains nine cells and for the manufacture of which three types of mask have to be superimposed. For each type of mask two versions are available,



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

FIG. 5.8. Plate of an integrated circuit with 16 numbered cells.

differing according to the position and number of defects, in the same way that automobiles of the same type possess slight differences. Our aim is to select from each of the three types of mask the version that will produce the least number of defects for the total combination.

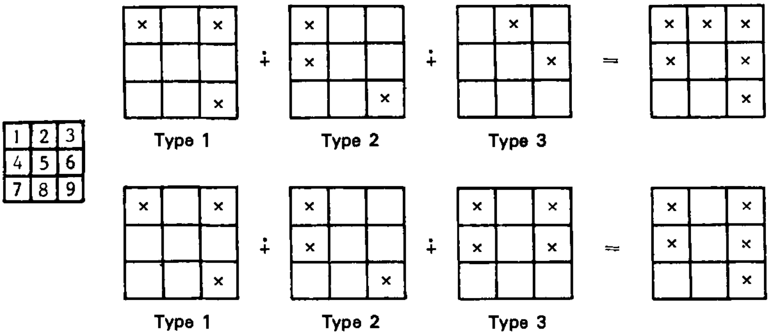


FIG. 5.9. The different number of defects that occur when using two versions of the same type of mask.

Let us examine Fig. 5.9. The first version of the type 3 mask produces six defects, whereas the second version using type 3′ produces only five. Allowing, therefore, for the inefficient return from the process of manufacture, we are obliged to construct a large number of masks (often as many as 50) of the same type. In practice more than ten types of masks are normally used to produce an integrated circuit with some 100 cells. At this level of combinatorial complexity, the manual selection of masks to produce a minimal number of defects is impractical. Mathematical programming enables us to construct a model and to solve this problem.

Let

$$(5.52)\qquad \begin{aligned} x_{ij} &= 1, && \text{if, for the optimal selection, we take version } j \text{ of type } i \text{ mask,} \\ &= 0, && \text{if we do not take it.} \end{aligned}$$

In our example (Fig. 5.9), $i = 1, 2, 3$, and $j = 1, 2$, since, for the purpose of simplification, we suppose that there are only two versions for each type of mask, although in practice there would be far more.

We have to choose a mask of each type and only one version of each type (if possible, the best), which provides the three constraints:

$$(5.53)\qquad \begin{aligned} x_{11}+x_{12} &\geqslant 1, \\ x_{21}+x_{22} &\geqslant 1, \\ x_{31}+x_{32} &\geqslant 1. \end{aligned}$$

The inequalities $\geqslant 1$ have been used to produce a certain homogeneity in the formulas, with the clear understanding that, when minimizing, only one version of each type will be selected.

If the type 1 mask exists in two versions having, respectively, defects in cells 1, 3, and 9 (as in Fig. 5.9) and in cells 2, 3, 4, and 7; if the type 2 mask has two versions with defects, respectively, in cells 1, 4, and 9 and cells 1, 4, 5, and 9; if the type 3 mask exists in two versions defective, respectively, in cells 2 and 6 and in cells 1, 3, 4, and 6 (type 3′ in Fig. 5.9), we can now state that selecting a mask that has a defect in cell p results in a cell p in the integrated circuit being defective. It should be noted that not more than three masks, one of each type, will be chosen.

Let

$$(5.54) \qquad \begin{aligned} w_p &= 1, \quad \text{if there is a defect in cell } p \text{ of the circuit,} \\ &= 0, \quad \text{if there is no such defect.} \end{aligned}$$

We now have the following relations for this simple example:

$$(5.55) \qquad \begin{aligned} x_{11}+x_{21}+x_{22}+x_{32} &\leqslant 3w_1, \\ x_{12}+x_{31} &\leqslant 3w_2, \\ x_{11}+x_{12}+x_{32} &\leqslant 3w_3, \\ x_{12}+x_{21}+x_{22}+x_{32} &\leqslant 3w_4, \\ x_{22} &\leqslant 3w_5, \\ x_{31}+x_{32} &\leqslant 3w_6, \\ x_{12} &\leqslant 3w_7, \\ 0 &\leqslant 3w_8, \\ x_{11}+x_{21}+x_{22} &\leqslant 3w_9, \end{aligned}$$

$$x_{11}, x_{12}, x_{21}, x_{22}, x_{31}, x_{32}, w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8, w_9 \in \{0, 1\}.$$

Let us explain the first of these relations. By hypothesis, only the masks of type 1 version 1, type 2 versions 1 and 2, and type 3 version 2 have a defect in cell 1. If we select one or more of these masks in an arbitrary manner we shall obtain a first member greater than 0 and less than or equal to 3, since the total number of masks chosen does not exceed 3. As w_1 is equal to 0 or 1 it must have the value 1, thus indicating a defect in cell 1 of the circuit.

We propose to minimize the total number of defects, that is,

$$(5.56) \qquad [\text{MIN}]\; z = \sum_{p=1}^{9} w_p,$$

with constraints (5.53) and (5.55).

This defines a model with bivalent variables. If we chose to ignore the particularities of the constraints, as for example that x_{ij} does not appear in (5.56), we could solve the problem by the use of the appropriate algorithms, but with attendant difficulties that we must immediately stress. Thus, for a problem of this type that includes 128 cells, 50 versions of each type and seven types of mask, there would be 128 bivalent variables w_p, 350 variables x_{ij} and $7+128=135$ constraints if a direct adaptative research algorithm DZLP1 were employed (see [K63] and [K64]), and with this method no solution was obtained after 2 hours 30 minutes use of a large third-generation computer. An attempt was also made with an older algorithm LIP1 using the procedure of Gomory (see Part 2, p. 301) on a very large second-generation computer, but it proved impossible to program the problem owing to lack of memory space.

This example underlines the difficulties encountered in the treatment of certain problems with bivalent values that include a large number of variables and constraints. We should not therefore be surprised at finding important mathematical developments that are sometimes difficult to take account of, as will be seen in certain sections of Part 2.

We are often obliged (and this is equally true for common linear programming), to construct special algorithms designed to take advantage of the particular structure of the problem. Thus, if we consider (5.55), we see that only one integer variable w_i appears in each inequality. This will be utilized in Section 22 to develop a very effective algorithm.

Section 6. **Arborescent and Cut Methods for Solving Programs with Integer Values**

1. Principle of Arborescent Methods

With arborescent methods an implicit enumeration of the solutions is employed. This differs from the Lemke–Spielberg algorithm in so much that with each iteration not only a solution but a subset of solutions must be examined. On this account these methods are often termed *multibranched*[1] or *arborescent.* They belong to a more general category termed *branch and bound,* the theory of which has been studied by B. Roy and P. Bertier.[2]

We shall now make use of a very simple example to explain the general procedure for these arborescent methods.

[1] See K. Spielberg, [K68], Enumerative Methods for Integer and Mixed Integer Programming. IBM Rep. N.Y. Scientific Center, 320-2928, March 1968. See also the works of R. Faure, and Y. Malgrange, who pioneered these methods some ten years ago (see [K10]).

[2] P. Bertier, and B. Roy, "A Solution Procedure for a Class of Problems Raising Combinatorial Character," Operations Research Center, Univ. of California, Berkeley, 1967.

Given

$$
(6.1)\qquad
\begin{aligned}
&(1)\quad [\text{MAX}]\quad z = x_1 + x_2\,,\\
&(2)\quad x_1 + 9/14 \quad x_2 \leqslant 51/14\,,\\
&(3)\quad -2x_1 + x_2 \leqslant 1/3\,,\\
&(4)\quad x_1, x_2 \geqslant 0\,,\\
&(5)\quad x_1, x_2 \text{ integers.}
\end{aligned}
$$

The domain of the solutions that satisfy constraints (2), (3), and (4) of (6.1) is shown in Fig. 6.1. The solutions corresponding to values x_1 and x_2 that satisfy (4) and (5) are shown by heavy dots.

If we solve the program without taking account of (5), the point $x_1 = 3/2$, $x_2 = 10/3$, namely A, for which $z = 29/6$, represents the maximum.

Now let us suppose that the optimal point with integer values is not too far distant from A. The integer values of x_1 nearest to $\frac{3}{2}$ are 1 and 2.

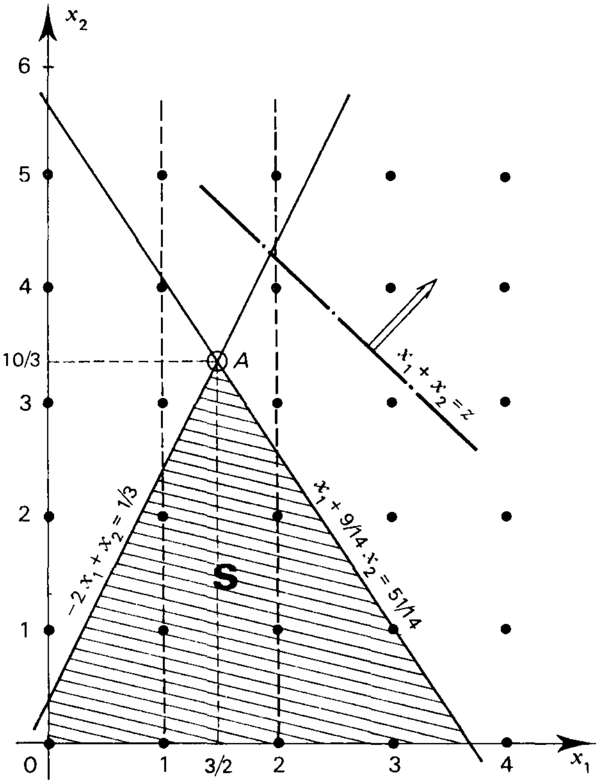


FIG. 6.1

Let us then consider the two following programs with integer values:

$$(6.2)\qquad \begin{array}{ll} (1) & [\mathrm{MAX}]\ z = x_1+x_2, \\ (2) & x_1+9/14\quad x_2 \leqslant 51/14, \\ (3) & -2x_1+x_2 \leqslant 1/3, \\ (4) & 2 \leqslant x_1,\quad 0 \leqslant x_2, \\ (5) & x_1,\, x_2 \text{ integers}, \end{array}$$

and

$$(6.3)\qquad \begin{array}{ll} (1) & [\mathrm{MAX}]\ z = x_1+x_1, \\ (2) & x_1+9/14\, x_2 \leqslant 51/14, \\ (3) & -2x_1+x_2 \leqslant 1/3, \\ (4) & 0 \leqslant x_1 \leqslant 1,\quad 0 \leqslant x_2, \\ (5) & x_1,\, x_2 \text{ integers}. \end{array}$$

By so doing, we have separated the set of solutions of program (6.1) into two subsets of solutions, one obtained by (6.2) and the other by (6.3). These two disjoint subsets have a union that gives all the solutions of (6.1), and in this way no integer solution is lost. With this procedure we have passed from the domain **S** shown in Fig. 6.1, which contains all the integer solutions of (6.1), into the domain $\mathbf{S}_1 \cup \mathbf{S}_2$ in Fig. 6.2, which also contains all the solutions for program (6.1). Now, let us solve programs (6.2) and (6.3) without taking into account their constraints (5), that is to say, as common linear programs.

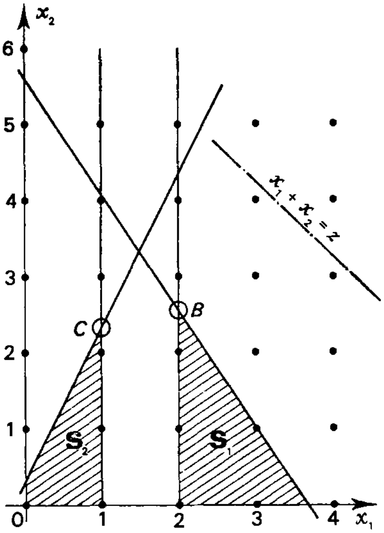


FIG. 6.2

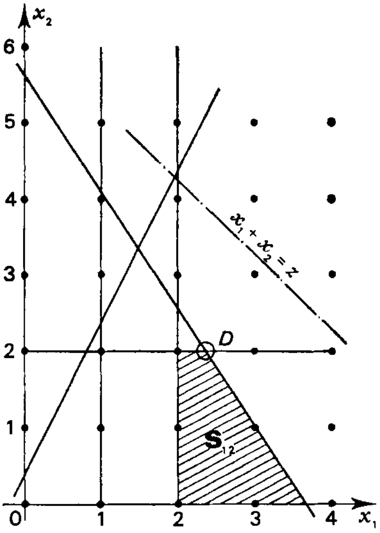


FIG. 6.3

We then obtain the following:

For the linear program (6.2) without (5),

(6.4) $x_1 = 2, \quad x_2 = 23/9, \quad \max z = 41/9$ (point B).

and for the linear program (6.3) without (5),

(6.5) $x_1 = 1, \quad x_2 = 7/3, \quad \max z = 10/3$ (point C).

Neither of these solutions has integer values.

The reader will be able to follow the development of the calculations on Figs. 6.2–6.5. Since point B gives a maximum upper value for z, the separation procedure will be continued from subset $\mathbf{S}_1$ of the solutions.

Let us separate subset $\mathbf{S}_1$ into two disjoint subsets $\mathbf{S}_{11}$ and $\mathbf{S}_{12}$, the union of which gives all the integer solutions for $\mathbf{S}_1$. In (6.4) we see that $x_2 = 23/9$, that is to say $2 \leqslant x_2 \leqslant 3$. Let us then take as supplementary constraints $x_2 \geqslant 3$ on the one hand and $x_2 \leqslant 2$ on the other, which gives

(6.6)
(1) $[\text{MAX}]\ z = x_1 + x_2,$
(2) $x_1 + 9/14\ x_2 \leqslant 51/14,$
(3) $-2x_1 + x_2 \leqslant 1/3,$
(4) $2 \leqslant x_1, \quad 3 \leqslant x_2,$
(5) x_1, x_2 nonnegative integers,

and, on the other hand,

(6.7)
(1) $[\text{MAX}]\ z = x_1 + x_2,$
(2) $x_1 + 9/14 x_2 \leqslant 51/14,$
(3) $-2x_1 + x_2 \leqslant 1/3,$
(4) $2 \leqslant x_1, \quad x_2 \leqslant 2,$
(5) x_1, x_2 nonnegative integers.

It is evident from Fig. 6.3 that program (6.6) is impossible as it does not contain any integer or other solution, since $x_2 \geqslant 3$ is not compatible with $\mathbf{S}_1$.

On the other hand, point D of domain $\mathbf{S}_{12}$ provides an optimal solution for program (6.7) without (5), namely,

(6.8) $x_1 = 33/14, \quad x_2 = 2, \quad \max z = 61/14.$

But this solution is not integer and our procedure must be continued. Comparing the solution at point D with that at point C, we have

(6.9) point C : $\max z = 10/3$; point D : $\max z = 61/14$.

Given that $61/14 > 10/3$ we shall continue with point D as our starting point; otherwise we should have had to return to C and continue from there.

Let us now consider domain $\mathbf{S}_{12}$ and the optimal solution (6.7), which is not integer. Since we have $x_1 = 33/14$ we separate $\mathbf{S}_{12}$ into two disjoint

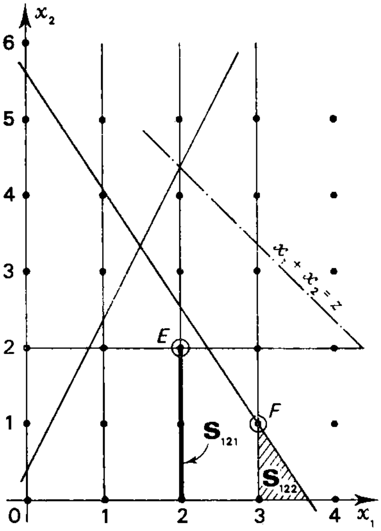


FIG. 6.4

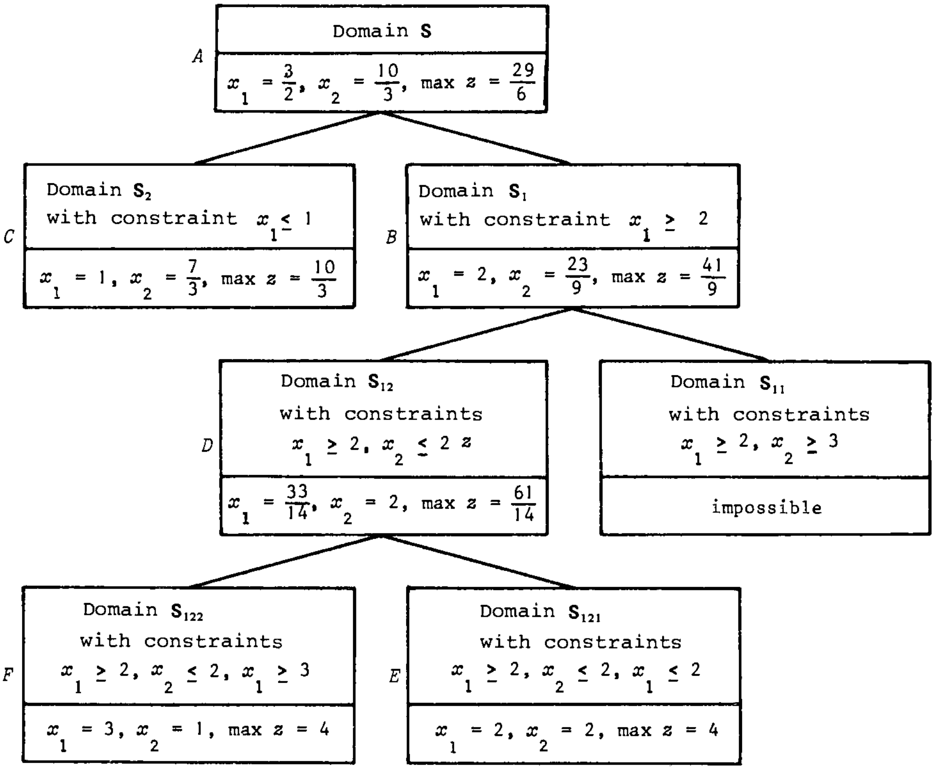

A
Domain $\mathbf{S}$
$x_1 = \frac{3}{2}, x_2 = \frac{10}{3}$, max $z = \frac{29}{6}$
C
Domain $\mathbf{S}_2$
with constraint $x_1 \leq 1$
$x_1 = 1, x_2 = \frac{7}{3}$, max $z = \frac{10}{3}$
B
Domain $\mathbf{S}_1$
with constraint $x_1 \geq 2$
$x_1 = 2, x_2 = \frac{23}{9}$, max $z = \frac{41}{9}$
D
Domain $\mathbf{S}_{12}$
with constraints
$x_1 \geq 2, x_2 \leq 2$ z
$x_1 = \frac{33}{14}, x_2 = 2$, max $z = \frac{61}{14}$
Domain $\mathbf{S}_{11}$
with constraints
$x_1 \geq 2, x_2 \geq 3$
impossible
F
Domain $\mathbf{S}_{122}$
with constraints
$x_1 \geq 2, x_2 \leq 2, x_1 \geq 3$
$x_1 = 3, x_2 = 1$, max $z = 4$
E
Domain $\mathbf{S}_{121}$
with constraints
$x_1 \geq 2, x_2 \leq 2, x_1 \leq 2$
$x_1 = 2, x_2 = 2$, max $z = 4$

FIG. 6.5. *Branch and bound* method of finding the solutions.

domains, the union of which provides all the integer solutions of $\mathbf{S}_{12}$. On the one hand, this gives the constraint $x_1 \leqslant 2$ and, on the other hand $x_1 \geqslant 3$, whence we obtain the following programs with which we shall associate domains $\mathbf{S}_{121}$ and $\mathbf{S}_{122}$:

$$
(6.10)\qquad \begin{array}{ll}
(1) & [\text{MAX}]\ z = x_1+x_2\,, \\
(2) & x_1+9/14\ x_2 \leqslant 51/14\,, \\
(3) & -2x_1+x_2 \leqslant 1/3\,, \\
(4) & 2 \leqslant x_1\,,\quad x_2 \leqslant 2\,,\quad x_1 \leqslant 2\,. \\
(5) & x_1\,, x_2 \ \ \text{nonnegative integers,}
\end{array}
$$

and

$$
(6.11)\qquad \begin{array}{ll}
(1) & [\text{MAX}]\ z = x_1+x_2\,, \\
(2) & x_1+9/14\ x_2 \leqslant 51/14\,, \\
(3) & -2x_1+x_2 = 1/3\,, \\
(4) & 2 \leqslant x_1\,,\quad x_2 \leqslant 2\,,\quad 3 \leqslant x_1\,, \\
(5) & x_1\,, x_2 \ \ \text{nonnegative integers.}
\end{array}
$$

Domains $\mathbf{S}_{121}$ and $\mathbf{S}_{122}$ are shown in Fig. 6.4. For the linear program (6.10) without (5) we find

$$
(6.12)\qquad x_1 = 2\,,\qquad x_2 = 2\,,\qquad \max z = 4\,.
$$

For the linear program (6.11) without (5) we find

$$
(6.12\text{a})\qquad x_1 = 3\,,\qquad x_2 = 1\,,\qquad \max z = 4\,.
$$

Let these be the points E and F on Fig. 6.4. This time we have found an optimal solution with integer values; indeed, we have found two, since points E and F are both suitable. Nevertheless, we must still verify that the maximum value obtained for z (namely, 4) is greater than the possible values obtained from the other hanging vertices of the arborescence in Fig. 6.5; all we need do is to compare it with the result obtained at C, and we then find $4 > 10/3$. Thus the points with integer values E and F are indeed optimal solutions of (6.1).

We have outlined the principle of arborescent methods (also called *multi-branch*)[1] because by evaluating, for example, subset $\mathbf{S}_{122}$ and by observing that the associated linear program had an integer solution, we were able to eliminate the elements of both set $\mathbf{S}_2$ and set $\mathbf{S}_{12}$, apart from the two optimums obtained. Using this method, the discovery of a solution in a subset often

[1] In mathematical works this method is termed *branch and bound*, and it should be noted that it refers to a wide field of optimization. It can prove lengthy and troublesome owing to the need to return to many new branchings.

enables us to eliminate inspection, whence the separation of several other subsets.

The procedure employed is an algorithm of optimization, even though we have used heuristic criteria such as that of separating $\mathbf{S}_1$ before $\mathbf{S}_2$. What is essential is that the convergence must be a property associated with the criterion employed and, as can be verified, such was the case in our procedure. With other procedures, still using the branch and bound method, the choice of a different criterion might have led us to select $\mathbf{S}_2$ before $\mathbf{S}_1$, but this change of order does not matter as long as the convergence toward an optimum is assured.

Let us note that this algorithm is also valid for cases where only some of the variables have integer values.

2. Difficulties of Utilization

The very simple instructional example above does not reveal the full difficulties of the method. It is often observed that, for a given problem, in the subsets obtained by separation the optimum is discovered fairly soon. In the above example this optimum was found in $\mathbf{S}_{12}$ and it was not necessary to return to other branchings. Such an early result is far from always occurring, and to complete the optimization it may prove necessary to perform a large number of branchings and separations. To realize the truth of this we need only consult the table (Fig. 6.6) that relates to the program OPHELIE MIXTE[1] used by the METRA group and shows the characteristics of treatment on one of the most powerful present-day computers.

In this table it will be observed that the number of iterations before stopping (an iteration consists of the separation of a subset and the solution of the linear program associated with it) is frequently greater than the number of the iteration at which the optimum was discovered. To reduce the number (and this method is employed in the majority of standard codes for mixed programming), use is made of the following criterion of stoppage that will be illustrated and explained by an example:

Given

$$\text{(6.13)}\qquad \begin{array}{ll} (1) & [\mathrm{MAX}]\ z = -x_1+x_2, \\ (2) & 2x_1+x_2 \leqslant 6, \\ (3) & -5/2\ x_1+x_2 \leqslant 0, \\ (4) & x_1 \geqslant 0, \quad x_2 \geqslant 0, \\ (5) & x_1, x_2 \text{ integers.} \end{array}$$

[1] Table taken from the article by R. Benayoun, B. Roy, and J. Tergny, Concerning the S.E.P. procedure in the program OPHELIE MIXTE, *METRA Rev.* 4, (1), 1970.

If we solve the associated linear program (Fig. 6.7), we find the optimum for point A: $x_1 = 4/3$, $x_2 = 10/3$, $z = 2$.

Let us propose to limit ourselves to an integer solution for which the deviation from the optimum is less than or equal to 1.5. This means that we shall end the separation procedure as soon as we are certain that the maximum of the linear program, without the integer constraint, does not exceed the best integer solution obtained by more than 1.5.

Hence, if we separate domain $\mathbf{S}$ into two subsets $\mathbf{S}_1$ and $\mathbf{S}_2$, where $\mathbf{S}_1$ is such that $x_1 \geqslant 2$ and $\mathbf{S}_2$ is such that $x_1 \leqslant 1$, we find

$$(6.14)\qquad \begin{array}{ll} (1) & [\mathrm{MAX}]\ z = -x_1+x_2\,, \\ (2) & 2x_1+x_2 \leqslant 6\,, \\ (3) & -5/2\ x_1+x_2 \leqslant 0\,, \\ (4) & 2 \leqslant x_1\,,\quad 0 \leqslant x_2\,, \\ (5) & x_1\,, x_2 \text{ integers,} \end{array}$$

and

$$(6.15)\qquad \begin{array}{ll} (1) & [\mathrm{MAX}]\ z = -x_1+x_2 \\ (2) & 2x_1+x_2 \leqslant 6\,, \\ (3) & -5/2\ x_1+x_2 \leqslant 0\,, \\ (4) & x_1 \leqslant 1\,,\quad 0 \leqslant x_2\,, \\ (5) & x_1\,, x_2 \text{ integers.} \end{array}$$

The optimal solution of (6.14) without (5) is

$$(6.16)\qquad x_1 = x_2 = 2\,, \qquad \max z = 0\,, \qquad \text{point } B\,.$$

And that of (6.15) without (5) is

$$(6.17)\qquad x_1 = 1\,, \qquad x_2 = 5/2\,, \qquad \max z = 3/2\,, \qquad \text{point } C\,.$$

At this stage we have the arborescence of separation of Fig. 6.8.

We have found a solution with integer values in subset $\mathbf{S}_1$. We know that the optimum with integer values in $\mathbf{S}_2$ is less than or equal to 1.5, since this optimum corresponds to a more constrained problem than the associated linear problem that allows a maximum of 1.5. As we have already found an integer solution, we are only interested in a solution greater than 0 (value obtained) plus 1.5 (the exact optimum), namely $0+1.5 = 1.5$. Hence we are not concerned with the points of $\mathbf{S}_2$ and we take as the optimum nearest to 1.5

Type of problem		Oil investments	Oil investments	Allocation of planes in a commercial air company	Chemical investments	Allocation of ships	Oil investments	Oil investments	Agricultural investments
Characteristics of the programs	Number of constraints	24	138	314	326	202	604	1 244	16
	Number of continuous variables	0	104	0	485	260	1 955	3 884	0
	Number of bivalent variables	20	48	0	16	0	25	24	0
	Number of integer variables	0	0	298	0	26	0	0	15
	Number of nonnull coefficients in matrix of constraints	446	631	2 141	11 801	1 294	10 362	20 333	135
Continuous optimization	Number of changes of basis	31	185	508	300	61	1 060		
	Time of solution in seconds	0.65	8.52	28.6	19.2	2.0	162	4	
	Continuous value of optimum	30.23	913.52	181 086	210 745	36 672	19 701	82 976	13
Mixed or integer phases of separation	Number of iterations before stopping	48	291	18	23	358	34	30	42
	Number of changes of basis during mixed phase	198	809	254	487	335	4 582	741	88
	Value of optimum	53	323.2	187.128	208.388	26.136	18.308	79.097	14
	Number of iterations giving optimum	40	118	18	23	250	16	10	20
	Number of integer solutions obtained	1	1	1	1	10	2	4	2
	Time of solution of mixed phase in seconds	13.55	85.2	35.55	67.25	120.75	984	357.62	9.95

solution of the associated linear program

FIG. 6.6

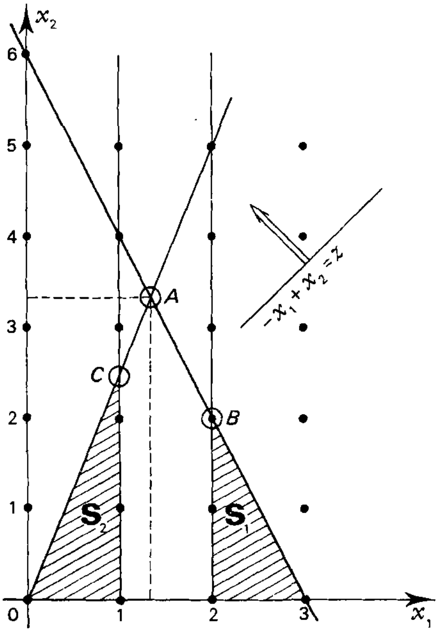
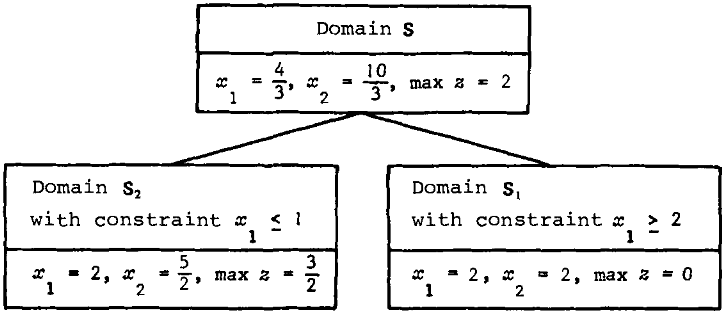


FIG. 6.7



Domain **S**

$x_1 = \frac{4}{3}$, $x_2 = \frac{10}{3}$, max $z = 2$

Domain S_2
with constraint $x_1 \leq 1$

$x_1 = 2$, $x_2 = \frac{5}{2}$, max $z = \frac{3}{2}$

Domain S_1
with constraint $x_1 \geq 2$

$x_1 = 2$, $x_2 = 2$, max $z = 0$

FIG. 6.8

the solution $x_1 = x_2 = 2$, $z = 0$. It can be verified that the true optimum corresponds to $x_1 = 1$, $x_2 = 2$, $z = 1$.

Many other heuristic criteria for stopping are used to choose the subsets to be separated and the separation variables. These criteria provide a fundamental contribution to the efficacy of present-day codes capable of solving problems of large-scale dimensions. Since they use linear programming as a subprogram for the computer, they should be combined with very productive codes of linear programming.

3. Principle of Cut Methods

These methods consist in replacing the conditions $x_i =$ a nonnegative integer, $i = 1, 2, \ldots, n$, where n is the number of variables, by $x_i \geqslant 0$, $i = 1, 2, \ldots, n$. We then add linear constraints termed *cuts* that are only verified if a solution has integer values. Each of these cuts constitutes a necessary condition for a solution to possess integer values.

We shall now explain the principle of these methods starting with a particular method of cut for which R. E. Gomory [K42] is responsible.

Let us consider the following example:

$$(6.18)\qquad \begin{array}{ll} (1) & [\text{MAX}]\ z = x_1 + x_2\,, \\ (2) & -x_1 + x_2 \leqslant 1\,, \\ (3) & 3x_1 + x_2 \leqslant 4\,, \\ (4) & x_1 \geqslant 0\,,\quad x_2 \geqslant 0\,, \\ (5) & x_1, x_2 \text{ integers.} \end{array}$$

If we disregard (5) in (6.18) the maximum occurs at point A for which we have (Fig. 6.9)

$$(6.19)\qquad x_1 = 3/4\,,\qquad x_2 = 7/4\,,\qquad \max z = 10/4\,.$$

If we now add to inequalities (2) and (3) in (6.18) the deviation variables u_1 and u_2, we shall obtain the following equivalent program:

$$(6.20)\qquad \begin{array}{ll} (1) & [\text{MAX}]\ z = x_1 + x_2\,, \\ (2) & -x_1 + x_2 + u_1 = 1\,, \\ (3) & 3x_1 + x_2 + u_2 = 4\,, \\ (4) & x_1, x_2, u_1, u_2 \geqslant 0\,, \\ (5) & x_1, x_2 \text{ integers.} \end{array}$$

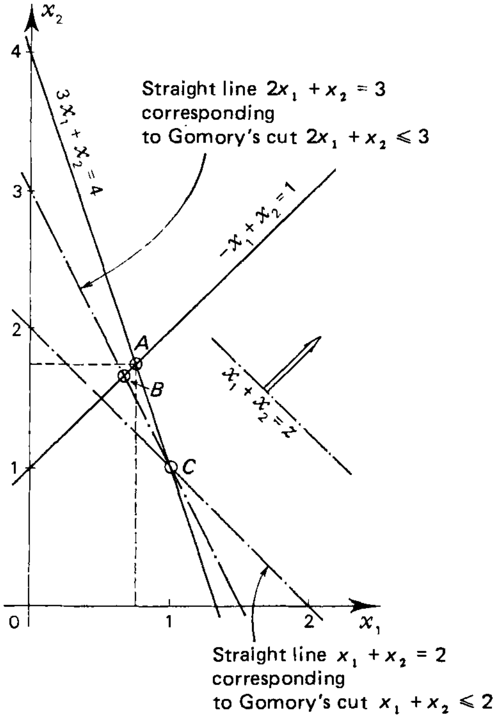
Let us now solve (2) and (3) of (6.20) to express x_1 and x_2 as functions of u_1 and u_2; it follows that

$$(6.21)\qquad x_1 = \frac{3}{4} + \frac{u_1}{4} - \frac{u_2}{4}\,,$$

$$(6.22)\qquad x_2 = \frac{7}{4} - \frac{3u_1}{4} - \frac{u_2}{4}\,.$$

In particular, if $u_1 = u_2 = 0$, we find $x_1 = 3/4$ and $x_2 = 7/4$, namely, point A, the intersection of the straight lines $-x_1 + x_2 = 1$ and $3x_1 + x_2 = 4$.

Let us consider lines (2) and (3) in Eq. (6.20) which have integer coefficients.



Straight line $2x_1+x_2=3$ corresponding to Gomory's cut $2x_1+x_2 \leqslant 3$

Straight line $x_1+x_2=2$ corresponding to Gomory's cut $x_1+x_2 \leqslant 2$

FIG. 6.9

If x_1 and x_2 are integers, since the second members 1 and 4 are integers, it follows that u_1 and u_2 must also be integers. We shall now, starting from (6.21) effect a *Gomory cut.*

x_1 integer gives $3/4+u_1/4-u_2/4$ integer. Since u_2 must be an integer, we must have

$$\frac{3}{4}+\frac{u_1}{4}-\frac{u_2}{4}+u_2 \quad \text{integer.} \tag{6.23}$$

That is,

$$\frac{3}{4}+\frac{u_1}{4}+\frac{3u_2}{4} \quad \text{integer.} \tag{6.24}$$

Since u_1 and u_2 are nonnegative, the first possible integer is 1. We therefore have

$$\frac{3}{4}+\frac{u_1}{4}+\frac{3}{4}u_2 \geqslant 1, \tag{6.25}$$

that is to say,

$$\frac{u_1}{4}+\frac{3}{4}u_2 \geqslant \frac{1}{4}. \tag{6.26}$$

This is the supplementary constraint sought and not verified at point A ($u_1 = 0$, $u_2 = 0$) that is optimal for program (6.20) without constraint (5). Solving u_1 and u_2 as functons of x_1 and x_2 by means of lines (2) and (3) of Eq. (6.20), we obtain another expression of the cut (6.26)

$$\tfrac{1}{4}(1+x_1-x_2)+\tfrac{3}{4}(4-3x_1-x_2) \geqslant \tfrac{1}{4}, \tag{6.27}$$

namely,

$$2x_1+x_2 \leqslant 3 \qquad \text{(Gomory's cut)}. \tag{6.28}$$

In Fig. 6.9 it is apparent that point A does not satisfy constraint (6.28).

Hence, to the relations (6.18) we now add constraint (6.28), and this gives

$$\text{(6.29)} \qquad \begin{array}{ll} (1) & [\text{MAX}]\ z = x_1+x_2, \\ (2) & -x_1+x_2 \leqslant 1, \\ (3) & 3x_1+x_2 \leqslant 4, \\ (6) & 2x_1+x_2 \leqslant 3, \\ (4) & x_1 \geqslant 0, \quad x_2 \geqslant 0, \\ (5) & x_1, x_2 \quad \text{integers}. \end{array}$$

The maximum, if we ignore constraint (5), is reached at point B (Fig. 6.9) for which we have $x_1 = 2/3$, $x_2 = 5/3$, $\max z = 7/3$.

In the same manner as before we introduce the deviation variables u_1, u_2, and u_3 into lines (2), (3), and (6) of Eq. (6.29). Thus (6.29) becomes

$$\text{(6.30)} \qquad \begin{array}{ll} (1) & [\text{MAX}]\ z = x_1+x_2, \\ (2) & -x_1+x_2+u_1 = 1, \\ (3) & 3x_1+x_2+u_2 = 4, \\ (6) & 2x_1+x_2+u_3 = 3, \\ (4) & x_1, x_2, u_1, u_2, u_3 \geqslant 0, \\ (5) & x_1, x_2 \quad \text{integers}. \end{array}$$

Let us express x_1 and x_2 as functions of u_1, u_2, and u_3 beginning at (2) and

(6). By elimination, we obtain

$$x_1 = \frac{2}{3} + \frac{1}{3}u_1 - \frac{1}{3}u_3, \tag{6.31}$$

$$x_2 = \frac{5}{3} - \frac{2}{3}u_1 - \frac{1}{3}u_3. \tag{6.32}$$

Let us note that we might have used (3) and (6).

Beginning at (6.31), let us produce a Gomory cut. x_1 integer results in $2/3+u_1/3-u_3/3$ integer. Since u_3 is integer (integer coefficients in (6) of (6.30) we have

$$\frac{2}{3} + \frac{u_1}{3} - \frac{u_3}{3} + u_3 \quad \text{integer}, \tag{6.33}$$

hence

$$\frac{2}{3} + \frac{u_1}{3} - \frac{u_3}{3} + u_3 \geqslant 1, \tag{6.34}$$

since u_1 and u_3 are nonnegative integers.

By expressing u_1 and u_3 as functions of x_1, x_2 beginning with (2) and (6) of Eq. (6.30), we discover a new Gomory cut,

$$x_1+x_2 \leqslant 2. \tag{6.35}$$

By adding this new constraint to (6.29) it follows that

$$\begin{aligned}
&(1)\quad [\text{MAX}]\ z = x_1+x_2,\\
&(2)\quad -x_1+x_2 \leqslant 1,\\
&(3)\quad 3x_1+x_2 \leqslant 4,\\
&(6)\quad 2x_1+x_2 \leqslant 3,\\
&(7)\quad x_1+x_2 \leqslant 2,\\
&(4)\quad x_1 \geqslant 0, \quad x_2 \geqslant 0,\\
&(5)\quad x_1, x_2 \ \text{integers}.
\end{aligned} \tag{6.36}$$

The maximum is now attained at point C for which $x_1 = 1$, $x_2 = 1$, max $z = 2$, that is to say, a point corresponding to a solution with integer values, which is indeed the optimum for the given program (6.18).

We should observe that, in the course of the different iterations, the denominators of the different fractions 1/4, 1/3, ... that are encountered diminish in the expressions such as (6.21)–(6.34). This is an interesting feature of Gomory's method and makes its convergence possible.

4. Difficulties of Utilization

The theory of cut methods dates from 1958, and the brilliance of the algebraic procedure seemed to hold out great promise. Nevertheless, at the present time no industrial code of programming in integers uses it in its complete form, although some of the most popular arborescent methods utilize the cut as a subsidiary aid. The reason for this neglect is the slowness with which convergence takes place. Starting with the first point obtained as a solution of the associated linear program by ignoring the condition $x_1, x_2, ..., x_n$ integers, we may have to associate a considerable number of cuts to obtain an optimal point, as we shall prove in Part 2, page 327. The reader can already be convinced of this drawback from the example of Fig. 6.9 where two cuts had to be added to obtain an optimal integer point. If one were to take example (6.1), which was solved by arborescent methods, and if one attempted to solve it by the cut method, a large number of cuts would be needed to find a point corresponding to an integer solution.

However we have seen that this method can be very effective for problems involving the *covering of a set*, such as that of the plumber in Section 3, where all the coefficients of the initial matrix are 0, $+1$, or -1.

Cut methods leave room for important theoretical developments for finding more effective cuts that eliminate a larger part of the domain of the constraints and thereby overcome the difficulties of convergence.

Section 7. **Programs with Mixed Numbers**

1. Problems with Mixed Numbers

Both linear and integer programming are subject to a generalization that is of importance in programming with mixed numbers.

Let $\mathbf{S}$ be the set of solutions for a linear program, and let us suppose that every solution $[s] \in \mathbf{S}$ contains $n+p$ variables such that n variables are constrained to take integer nonnegative values only, whereas the other p variables are only constrained to take real nonnegative values:

$$(7.1) \qquad [s] = [x_1, x_2, \ldots, x_n; y_1, y_2, \ldots, y_p],$$

where the $x_i, i = 1, 2, \ldots, n$ are nonnegative integers and the $y_j, j = 1, 2, \ldots, p$ are nonnegative real numbers.

A program in which we impose the condition of only accepting solutions of type (7.1) is a program with mixed numbers.

Hence a solution such as (7.1) may be shown in the following form, illustrated by an example where $n = 3$ and $p = 2$:

$$(7.2) \qquad [s] = [x_1, x_2, x_3; y_1, y_2] = [3, 0, 11; 1.27, 5.98].$$

A solution such as

$$(7.3) \qquad [s'] = [x_1', x_2', x_3'; y_1', y_2'] = [3.2; 0.4; 10.78; 2.08; 6.43]$$

could not be suitable.

Let us first consider a very elementary example, and let the program with mixed numbers be

$$(7.4) \qquad \begin{aligned} &[\text{MIN}]\ z = x_1+y_1+y_2, \\ &6.4x_1+3.2y_2 \leqslant 6, \\ &2x_1+3y_1+3y_2 \geqslant 4, \\ &y_1 \leqslant 3, \\ &x_1 \text{ real}, \quad y_1 \text{ and } y_2 \text{ integers}, \quad x_1, y_1, y_2 \geqslant 0. \end{aligned}$$

Set Σ of all the solutions of program (7.4), without the constraint y_1 and y_2 integers, can be represented in a three-dimensional space (Fig. 7.1) and is formed by all the points inside or on the surface of the convex polyhedron *ABCDEFGH* shown by heavy lines.

If we now impose the condition that y_1 and y_2 are integers, the subset of the mixed solutions $\mathbf{S} \subset \Sigma$ will be given by

$$(7.5) \qquad \begin{aligned} &[\tfrac{1}{2} \leqslant x_1 \leqslant 15/16, \; y_1 = 1, \; y_2 = 0], \\ &[0 \leqslant x_1 \leqslant 7/16, \; y_1 = 1, \; y_2 = 1], \\ &[0 \leqslant x_1 \leqslant 15/16, \; y_1 = 2, \; y_2 = 0], \\ &[0 \leqslant x_1 \leqslant 7/16, \; y_1 = 2, \; y_2 = 1], \\ &[0 \leqslant x_1 \leqslant 15/16, \; y_1 = 3, \; y_2 = 0], \\ &[0 \leqslant x_1 \leqslant 7/16, \; y_1 = 3, \; y_2 = 1], \end{aligned}$$

as the reader can verify in Fig. 7.1.

The optimal solution of the linear program in which we are considering solutions belonging to Σ is, as can be verified by using one of the methods given in Volume 1,

$$(7.6) \qquad [s^*] = [x_1 = 15/16, y_1 = 17/24, y_2 = 0], \qquad z^* = 1.64,$$

that is to say, corresponds to point A in Fig. 7.1.

To obtain the optimal solution or solutions with the supplementary non-negative integer constraints y_1 and y_2, we consider the six subsets of solutions

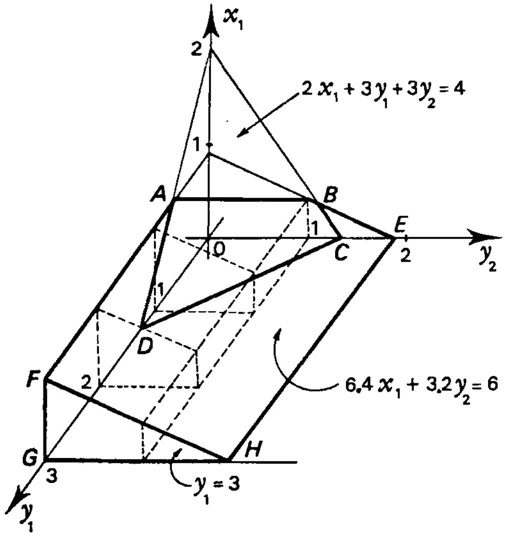


FIG. 7.1

given by (7.5). The domains of z are given opposite each corresponding subset.

$$\begin{aligned}
&[\tfrac{1}{2} \leqslant x_1 \leqslant 15/16,\ y_1 = 1,\ y_2 = 0] : && 3/2 \leqslant z \leqslant 31/16, \\
&[0 \leqslant x_1 \leqslant 7/16,\ \ y_1 = 1,\ y_2 = 1] : && 32/16 \leqslant z \leqslant 39/16, \\
&[0 \leqslant x_1 \leqslant 15/16,\ y_1 = 2,\ y_2 = 0] : && 32/16 \leqslant z \leqslant 47/16, \\
&[0 \leqslant x_1 \leqslant 7/16,\ \ y_1 = 2,\ y_2 = 1] : && 3 \leqslant z \leqslant 55/16, \\
&[0 \leqslant x_1 \leqslant 15/16,\ y_1 = 3,\ y_2 = 0] : && 3 \leqslant z \leqslant 63/16, \\
&[0 \leqslant x_1 \leqslant 7/16,\ \ y_1 = 3,\ y_2 = 1] : && 4 \leqslant z \leqslant 71/16.
\end{aligned} \tag{7.7}$$

From an examination of (7.7) it will be observed that there is one and only one optimal solution corresponding to

$$[s_m^*] = [x_1 = 1/2, y_1 = {}'1, y_2 = 0], \quad z_m^* = 3/2 = 1.5. \tag{7.8}$$

The method employed above can be generalized and used in cases where the number of x_i and y_i variables both remain small; otherwise there are too many domains such as (7.7) to examine in relation to each other. Hence it is possible to define the following procedure.

If we know a priori all the vectors $[y] = [y_1, y_2, \ldots, y_n]$ forming part of the vectors $[s]$ that constitute the solutions for set **S**, then with the constraints of the given program as our starting point, we define the domain relating to the other variables $x_1, x_2, \ldots, x_n$. Each domain thus defined appears in the form of new constraints, and it is then necessary to optimize the economic function z for this new set. This was the procedure used from (7.4) to (7.8).

This method consistutes a basic principle in certain algorithms and heuristic procedures (see [K59]), although other procedures that are more difficult to explain, but which produce a better convergence, are generally regarded as preferable. These will be explained in Sections 21 and 23.

2. An Example: Factory Location

Let us consider the problem of a manufacturer who produces extensively sold goods that are to be distributed from new factories for which he has to decide the location and size. He knows the distribution of his retailers as well as the particular requirements of each.

To simplify the problem let us suppose that only one product is manufactured and also that the problem does not apply to consecutive periods but to a continuous one. The model that we shall construct could be generalized for a sequential problem with several different products, although we should then have to expect a considerable increase in the number of variables and of constraints.

Let $b_1, b_2, \ldots, b_n$ represent the known requirements of the n retailers, and let $a_1, a_2, \ldots, a_m$ be the productive capacities of the m factories of varying size to be located. Each of these m factories has a unit cost of construction f_i, $i = 1, 2, \ldots, m$, and the cost of transporting a unit of merchandise from the ith factory to the jth retailer is c_{ij}. Let x_{ij} be the number of units transported between these two.

$$(7.9) \qquad \begin{aligned} y_i &= 1, \quad \text{if we build the } i\text{th factory,} \\ &= 0, \quad \text{if we do not build it.} \end{aligned}$$

Our aim is to minimize the total costs of construction and distribution while satisfying the requirements of the retailers, namely,

$$(7.10) \qquad [\text{MIN}]\ z = \sum_{i=1}^{m} \left[f_i y_i + \sum_{j=1}^{n} c_{ij} x_{ij} \right],$$

$$(7.11) \qquad \sum_{i=1}^{m} x_{ij} \geqslant b_j, \qquad j = 1, 2, \ldots, n$$

(to satisfy the requirements of the n retailers),

$$(7.12) \qquad \sum_{j=1}^{n} x_{ij} \leqslant a_i y_i, \qquad i = 1, 2, \ldots, m,$$

$$y_i = 0 \text{ or } 1, \qquad i = 1, 2, \ldots, m,$$

$$x_{ij} \geqslant 0, \quad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, n.$$

With regard to (7.12) we must observe that if factory i is not built ($y_i = 0$), no production and hence no delivery is possible from that factory and, if it is built ($y_i = 1$), only its production capacity a_i can be delivered.

If the sites for the factories were arbitrarily fixed, we should be confronted with a transport problem of a classic type (see Volume 1, page 51). If, in addition, the production capacity of the factories were taken as being unlimited, the solution of this problem would be a simple one; each retailer would obtain his supplies from the nearest factory, and this would produce a model sometimes referred to as a *simple siting problem.*

To find an optimal solution for the program in mixed numbers (7.10)–(7.12) we can solve the 2^m linear programs in x_{ij} (transportation problems) for each of the 2^m values of the vector $[y_1, y_2, ..., y_m]$, $i = 1, 2, ..., m$, a fact on which one of Spielberg's algorithms is based. To diminish the number of transportation problems to be solved, he employs expanded methods of exclusion (see Section 4, page 47).

We now show some results that are given in [K67] when using a medium-size computer of the third generation.

m	n	t in minutes	Number of iterations
20	35	2	229
30	80	10	2 129
60	80	> 60	> 10 300

FIG. 7.2

We note that the problems capable of solution by this *first generation* of algorithms (1967) were of modest dimensions.

3. Optimization of Nonlinear Economic Functions Separable for Addition

We shall now show that it is possible to transform a problem of optimization (minimal or maximal) of any function, whatever its constraints, to a program with mixed numbers (PMN) on condition that this function and its constraints are separable in relation to addition.

This is an important fact, since it is now possible to solve PMN programs of very large dimensions on powerful third-generation computers. The formulation given below may certainly sometimes result in a considerable increase in the number of variables but, with the advances that have been made in the coding of these programs and their transmission through a computer, the solution of nonlinear programs of large magnitude can be obtained in a comparatively short time.

It is necessary to suppose that the economic function to be optimized is *separable for addition*, that is to say, that it can be written as a sum of functions, each with one variable, that is,

$$(7.13)\qquad [\text{OPT}]\ z = f_1(x_1) + f_2(x_2) + \ldots + f_k(x_k).$$

Hence function z that follows is separable for addition:

$$(7.14)\qquad f_1(x_1) + f_2(x_2) = \sin x_1/2 + x_2^3/2.$$

The basis of the principle is to transform each nonlinear function to be optimized into another function, defined as *by intervals*, in such a manner that a linear function is obtained in each interval.

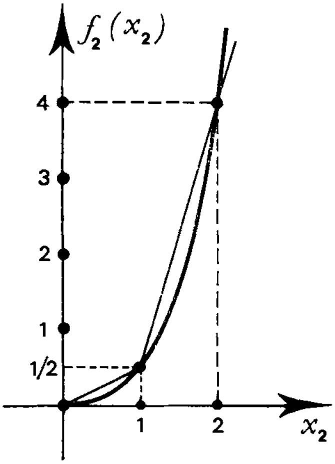


FIG. 7.3

In this manner let us consider the function

$$(7.15)\qquad f_2(x_2) = x_2^3/2, \qquad x_2 \geqslant 0,$$

that is shown in Fig. 7:3.

The linear approximation by intervals consists in breaking up the interval $0 \leqslant x_2 < \infty$ or a smaller interval $0 \leqslant x_2 \leqslant a$, the value of a being such that the interval will cover all the possible values of x_2 in the program.

Let us consider, for example, (7.15) and let us suppose that we are restricted to the case where $0 \leqslant x_2 \leqslant 2$, a hypothesis we are fully justified in making owing to the form of the program in which x_2 appears. Let us state that x_2 is a linear combination of the values at the following points: $x_2 = 0, 1,$ or 2. Let us now suppose $u_{20} = 0$, $u_{21} = 1$, $u_{22} = 2$, and state,

$$(7.16)\qquad x_2 = x_{20}\cdot u_{20} + x_{21}\cdot u_{21} + x_{22}\cdot u_{22}, \qquad x_{20}, x_{21}, x_{22} \geqslant 0.$$

Let us next consider how to define the variables x_{20}, x_{21}, x_{22} in such a manner that (7.16) can replace x_2. We have

$$x_2 = 0.x_{20}+1.x_{21}+2.x_{22}. \tag{7.17}$$

If $0 \leqslant x_2 \leqslant 1$, let us suppose[1] $x_{20}+x_{21} = 1$ and $x_{22} = 0$. To each value of x_2 one and only one value for each of the variables x_{20} and x_{21} can be made to correspond. Thus, for $x_2 = 0.6$, it follows that $x_{20} = 0.4$ and $x_{21} = 0.6$. If $1 \leqslant x_2 \leqslant 2$, let us suppose $x_{20} = 0$ and $x_{21}+x_{22} = 1$. To each value of x_2 one and only one value for each of the variables x_{21} and x_{22} can be made to correspond. Thus, if $x_2 = 1.7$, it follows that $x_{21} = 0.3$ and $x_{22} = 0.7$.

It must be understood that because of the condition $x_{2i}+x_{2,i+1} = 1$, $x_{2j} = 0$, $j \neq i$, $j \neq i+1$ that has been imposed, for each i we shall have $0 \leqslant x_{21} \leqslant 1$.

Lastly, the variable x_2 will be replaced by the three variables x_{20}, x_{21}, x_{22} such that

$$\begin{aligned} & x_2 = 0.x_{20}+1.x_{21}+2.x_{22}, \\ & 0 \leqslant x_2 \leqslant 1 : \quad x_{20} = 1-x_2, \quad x_{21} = x_2, \quad x_{22} = 0, \\ & 1 \leqslant x_2 \leqslant 2 : \quad x_{20} = 0, \quad x_{21} = 2-x_2, \quad x_{22} = x_2-1. \end{aligned} \tag{7.18}$$

With the same instructional intent, we shall now give similar explanations concerning the variable x_1, which intervenes in the function $\sin x_1/2$ of (7.14).

Let us suppose that this variable is only to be taken into account in the interval $0 \leqslant x_1 \leqslant 4\pi$. Let us take the values 0, π, 2π, 3π, and 4π in this interval (see Fig. 7.4). Let us suppose $u_{10} = 0$, $u_{11} = \pi$, $u_{12} = 2\pi$, $u_{13} = 3\pi$, $u_{14} = 4\pi$ and state

$$x_1 = x_{10}.u_{10}+x_{11}.u_{11}+x_{12}.u_{12}+x_{13}.u_{13}+x_{14}.u_{14}, \tag{7.19}$$

which will give

$$\begin{aligned} & x_1 = 0.x_{10}+\pi.x_{11}+2\pi.x_{12}+3\pi.x_{13}+4\pi.x_{14}, \\ & \text{(a)}\ 0 \leqslant x_1 \leqslant \pi : \quad x_{10} = 1-x_1/\pi, \quad x_{11} = x_1/\pi, \\ & \qquad\qquad x_{12} = x_{13} = x_{14} = 0; \\ & \text{(b)}\ \pi \leqslant x_1 \leqslant 2\pi : \quad x_{10} = 0, \quad x_{11} = 2-x_1/\pi, \\ & \qquad\qquad x_{12} = x_1/\pi-1, \quad x_{13} = x_{14} = 0; \\ & \text{(c)}\ 2\pi \leqslant x_1 \leqslant 3\pi : \quad x_{10} = x_{11} = 0, \quad x_{12} = 3-x_1/\pi, \\ & \qquad\qquad x_{13} = x_1/\pi-2, \quad x_{14} = 0; \\ & \text{(d)}\ 3\pi \leqslant x_1 \leqslant 4\pi : \quad x_{10} = x_{11} = x_{12} = 0, \\ & \qquad\qquad x_{13} = 4-x_1/\pi, \quad x_{14} = x_1/\pi-3. \end{aligned} \tag{7.20}$$

[1] In a more general manner we could state that $\alpha x_{20}+\beta x_{21} = 1, \alpha > 0, \beta > 0$, which would be another method of approximation; but for theoretical reasons, we prefer to take $\alpha+\beta = 1$, and this will apply to what follows.

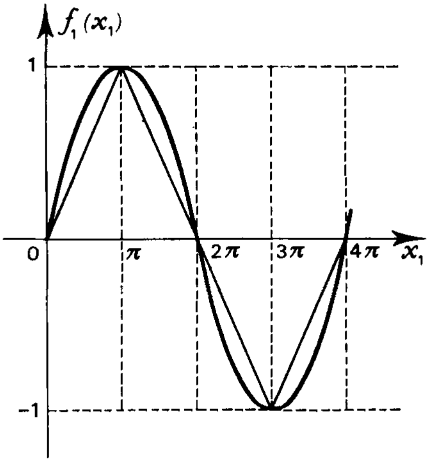


FIG. 7.4

The approximization of functions $f_1(x_1)$ and $f_2(x_2)$ will then be made by means of the expressions

$$(7.21) \qquad \begin{aligned} f_1(x_1) = \sin x_1/2 &= x_{10} f_1(u_{10}) + x_{11} f_1(u_{11}) + x_{12} f_1(u_{12}) \\ &\quad + x_{13} \cdot f_1(u_{13}) + x_{14} f_1(u_{14}) \\ &= x_{10} \,.\, \sin 0 + x_{11} \,.\, \sin \pi/2 + x_{12} \,.\, \sin 2\pi/2 \\ &\quad + x_{13} \,.\, \sin 3\pi/2 + x_{14} \,.\, \sin 4\pi/2 \\ &= x_{11} - x_{13}, \end{aligned}$$

with $x_{10}, x_{11}, x_{12}, x_{13}, x_{14}$ defined by (7.20).

$$(7.22) \qquad \begin{aligned} f_2(x) = x_2^3/2 &= x_{20} f_2(u_{20}) + x_{21} f_2(u_{21}) + x_{22} f_2(u_{31}) \\ &= x_{20} \,.\, 0^3/2 + x_{21} \,.\, 1^3/2 + x_{22} \,.\, 2^3/2 \\ &= x_{21}/2 + 4x_{22}, \end{aligned}$$

with x_{20}, x_{21}, x_{22} defined by (7.18).

These are not linear constraints since, for $0 \leqslant x_2 \leqslant 1$, we have $x_{20} = 1 - x_2$ and, for $1 \leqslant x_2 \leqslant 2$, we have $x_{20} = 0$. We thus pass from the variables x_1 and x_2 to the variables $x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{20}, x_{21}, x_{22}$ for which we shall use a program that is still nonlinear, but which we can present in the form of a linear program containing integer variables.

$$(7.23) \qquad [\text{MIN}]\ z = x_{11} - x_{13} + x_{21}/2 + 4x_{22},$$

where x_{11}, x_{13}, x_{21}, x_{22} assume their values in the intervals defined by (7.18) and (7.20).

Let us now return to example (7.14) and complete the economic function with a constraint in order to study the program. We shall also impose the condition that this constraint must be separable for addition. Let us, for example, take a constraint such as

$$(7.24) \qquad x_1^2 - 2x_2 \leqslant 3,$$

that is not linear but is separable for addition (x^2 and $-2x_2$ give $x_1^2 + (-2x_2)$).

We shall now have the nonlinear program:

$$(7.25) \qquad \begin{aligned} &(1)\ \ [\text{MIN}]\ z = f_1(x_1) + f_2(x_2) \\ &\qquad\qquad\quad\ = \sin x_1/2 + x_2^3/2, \\ &(2)\ \ x_1^2 - 2x_2 \leqslant 3, \\ &(3)\ \ x_1, x_2 \geqslant 0. \end{aligned}$$

In passing to the variables $x_{10}, \ldots, x_{14}$, $x_{20}, \ldots, x_{22}$ we shall have as the approximation of the economic function

$$[\text{MIN}]\ z = x_{11} - x_{13} + \tfrac{1}{2}x_{21} + 4x_{22}.$$

We must add a constraint, since, for every x_1 included in the closed interval $[u_{1i}, u_{1i+1}]$ we have $x_1 = u_{1i} \cdot x_{1i} + u_{1i+1}x_{1i+1}$ with $x_{1i} + x_{1i+1} = 1$, the other $x_{ij} = 0$, $j = i$, $j = i+1$. Hence, whatever the interval in which x_1 is situated, we can state that we have

$$(7.26) \qquad \begin{aligned} &x_{10} + x_{11} + x_{12} + x_{13} + x_{14} = 1 \quad \text{and, in the same way,} \\ &x_{20} + x_{21} + x_{22} = 1. \end{aligned}$$

These equations do not completely express conditions (7.18) and (7.20). Let us therefore introduce subsidiary integer bivalent variables y_{10}, y_{11}, y_{12}, y_{13}, y_{20}, y_{21}.

$$(7.27) \qquad \begin{array}{lll} y_{10} = 1, & \text{if } 0 \leqslant x_1 \leqslant \pi, & \text{that is, if } x_{10} \text{ and/or } x_{11} \text{ are nonnull,} \\ y_{11} = 1, & \text{if } 0 \leqslant x_1 \leqslant \pi, & \text{that is, if } x_{11} \text{ and/or } x_{12} \text{ are nonnull,} \\ y_{12} = 1, & \text{if } \pi \leqslant x_1 \leqslant 2\pi, & \text{that is, if } x_{12} \text{ and/or } x_{13} \text{ are nonnull,} \\ y_{13} = 1, & \text{if } 2\pi \leqslant x_1 \leqslant 3\pi, & \text{that is, if } x_{13} \text{ and/or } x_{14} \text{ are nonnull,} \\ y_{20} = 1, & \text{if } 0 \leqslant x_2 \leqslant 1, & \text{that is, if } x_{20} \text{ and/or } x_{21} \text{ are nonnull,} \\ y_{21} = 1, & \text{if } 1 \leqslant x_2 \leqslant 2, & \text{that is, if } x_{21} \text{ and/or } x_{22} \text{ are nonnull.} \end{array}$$

We observe that conditions (7.27) imply that a single y_{11} and a single y_{21}

are equal to 1. We can therefore replace (7.27) by

$$\begin{array}{llcl}
 & x_{10} \leqslant y_{10}, & & \\
 & x_{11} \leqslant y_{10}+y_{11}, & & x_{20} \leqslant y_{20}, \\
(7.28) & x_{12} \leqslant y_{11}+y_{12}, & \text{and} & x_{21} \leqslant y_{20}+y_{21}, \\
 & x_{13} \leqslant y_{12}+y_{13}, & & x_{22} \leqslant y_{21}, \\
(7.29) & x_{14} \leqslant y_{13}, & & y_{20}+y_{21} = 1. \\
 & y_{10}+y_{11}+y_{12}+y_{13} = 1. & &
\end{array}$$

Now let us make the same type of approximation for constraint (7.24) that we made for $f_1(x_1)$ and $f_2(x_2)$ with the help of (7.21) and (7.22). Let us suppose $\varphi_1(x_1) = x_1^2$ and $\varphi_2(x_2) = -2x_2$ and state,

$$\begin{aligned}
(7.30) \qquad \varphi_1(x_1) = x_1^2 &= x_{10} \cdot \varphi_1(u_{10}) + x_{11} \cdot \varphi_1(u_{11}) + x_{12} \cdot \varphi_1(u_{12}) \\
&\qquad + x_{13} \cdot \varphi_1(u_{13}) + x_{14} \cdot \varphi_1(u_{14}) \\
&= x_{10} \cdot 0 + x_{11} \cdot \pi^{\,2} + x_{12} \cdot (2\pi)^2 \\
&\qquad + x_{13} \cdot (3\pi)^2 + x_{14} \cdot (4\pi)^2 \\
&= \pi^2 . x_{11} + 4\pi^2 . x_{12} + 9\pi^2 . x_{13} + 16\pi^2 . x_{14},
\end{aligned}$$

$$\begin{aligned}
(7.31) \qquad \varphi_2(x_2) = -2x_2 &= x_{20} \cdot \varphi_2(u_{20}) + x_{21} \cdot \varphi_2(u_{21}) + x_{22} \cdot \varphi_2(u_{22}) \\
&= x_{20} \cdot 0 + x_{21} \cdot (-2.1) + x_{22} \cdot (-2.2) \\
&= -2x_{21} - 4x_{22}.
\end{aligned}$$

Constraint (7.24) can thus be expressed as

$$(7.32) \qquad \pi^2 . x_{11} + 4\pi^2 . x_{12} + 9\pi^2 . x_{13} + 16\pi^2 . x_{14} - 2x_{21} - 4x_{22} \leqslant 3,$$

which is a linear constraint.

Finally, the nonlinear program (7.25) can be represented by an approximation that reduces it to the program with mixed numbers.

$$\begin{array}{ll}
(7.33) & [\text{MIN}]\ z = x_{11} - x_{13} + x_{21}/2 + 4x_{22}, \\
 & \pi^2 . x_{11} + 4\pi^2 . x_{12} + 9\pi^2 . x_{13} + 16\pi^2 . x_{14} - 2x_{21} - 4x_{22} \leqslant 3, \\
 & \\
 & x_{10} \leqslant y_{10}, \\
 & x_{11} \leqslant y_{10} + y_{11}, \\
 & x_{12} \leqslant y_{11} + y_{12}, \\
 & x_{13} \leqslant y_{12} + y_{13},
\end{array}$$

$$x_{14} \leqslant y_{13},$$

$$x_{20} \leqslant y_{20},$$

$$x_{21} \leqslant y_{20}+y_{21},$$

$$x_{22} \leqslant y_{21},$$

$$x_{10}+x_{11}+x_{12}+x_{13}+x_{14} = 1,$$

$$y_{10}+y_{11}+y_{12}+y_{13} = 1,$$

$$x_{20}+x_{21}+x_{22} = 1,$$

$$y_{20}+y_{21} = 1.$$

$$x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{20}, x_{21}, x_{22} \geqslant 0,$$

$$y_{10}, y_{11}, y_{12}, y_{13}, y_{20}, y_{21} = 0 \text{ or } 1.$$

We have thus shown by means of an example that a nonlinear program, of which the economic function and the constraints are separable for addition, can be reduced by approximation to a program with mixed numbers. The method used permits us to generalize this property for all nonlinear programs separable for addition.

This formulation for such a class of problem is extremely important now that we possess computers with great capacity and central memory, able to deal with the considerable increase in the number of variables and constraints.

It should be noted that, in cases where the separated functions of which we are optimizing the sum are convex (see Section 14) in the economic function and in all the constraints, it can be shown that it is no longer necessary to introduce constraints such as $x_{ij} \leqslant y_{i,j-1}+y_{i,j}$ or even $x_{i0} \leqslant y_{i0}$, which results in a very considerable simplification.

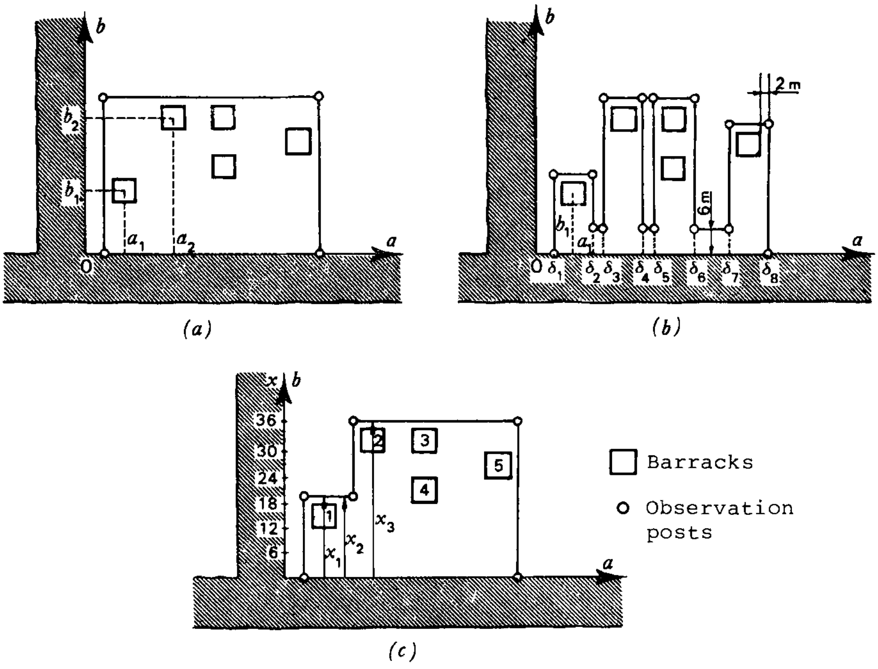
4. The Problem of the Prisoners-of-War Camp

We have already shown that practical and concrete problems frequently take the form of programs in mixed numbers (PMN) rather than in integers. Beginning with the following example we shall show that these PMN can be used in the most diverse cases. The present example appears in a guise that can be variously described as amusing or sinister.

Given a prisoners-of-war camp containing k barracks spread out over an area bounded on the south by a river and on the west by one of its tributaries (Figs. 7.5a, b, and c). These waterways are parallel to the axes Oa and Ob. The commandant of the camp plans to have the camp enclosed by a barbed wire fence, but intends to use only lengths of fencing parallel to the rivers without any slanting portions to connect them. At each angle of the fencing he is obliged by regulations to install an observation post, and alongside the fence bordering the river on the south side he is also compelled to leave a patrol road that must be at least 6 m in width.

Further, the camp must have a fence bordering the river on the south side that will permit easy access if troops are required. Also, the fencing must be located at least 2 m distant from the barracks to render escape more difficult, the barracks being square buildings 6 m × 6 m in size.

The camp has a stock of $2r$ ($r \geqslant 2$) observation posts, since the minimal number required for a rectangular camp is four (Fig. 7.5a). For obvious reasons of security the commandant wishes to minimize the area of the camp without exceeding his stock of observation posts. In Fig. 7.5a we have shown the camp with a minimal number of observation posts used, in Fig. 7.5b the camp with a minimal area, and in Fig. 7.5c the minimal area when only six observation posts are used. The shape represented in Fig. 7.5b requires 16 observation posts.



(a)

(b)

(c)

FIG. 7.5

It is clear from Fig. 7.5 that the fences erected to minimize the camp area, however many observation posts are installed, must run as close to the buildings as possible, namely, at a distance of 2 m, and in Fig. 7.5b, the fences must run at a distance of 6 m from the river, the minimal width required for the patrol road. Once the number of barracks is substantially increased, the minimal solution, if the specified number of observation posts is used, cannot be found by simple inspection. All that we know a priori is that the abscissas of the fences minimizing the area of the camp are of the form

$$(7.34) \qquad a_i \pm 5, \qquad i = 1, 2, \ldots, k.$$

Here a_i, b_i are the abscissa and the ordinate of the center of building i (which are known). The number 5 appears in (7.34) because the fencing cannot pass through barrack i, which has a half-width of 3 m and must have 2 m space left alongside it. All the possible values of a_i+5 are represented by $\delta_1, \delta_2, \delta_3, \ldots, \delta_8$ in Fig. 7.5b. Let x_i be the ordinate of the horizontal fence between the abscissas δ_i and δ_{i+1}. For example, in Fig. 7.5c we have

$$(7.35) \qquad x_0 = 0,\ x_1 = x_2 = 20\,\mathrm{m}, \qquad x_3 = x_4 = x_5 = x_6 = x_7 = 38\,\mathrm{m}, \\ x_8 = 0,$$

where x_8 is, by convention, equal to 0 since the fence ends at δ_8, and $x_0 = 0$ since it begins at δ_1; they are introduced to make the problem more general.

It must be understood that the a_i, $i = 1, 2, \ldots, k$, are data whereas the x_i terms are nonnegative variables that are also greater than or equal to 6 in order to allow for the patrol road (see Fig. 7.5b). Our aim is to minimize the area of the camp, which is expressed as

$$(0) \quad [\mathrm{MIN}]\, z = (\delta_2-\delta_1)\, x_1 + (\delta_3-\delta_2) x_2 + (\delta_4-\delta_3)\, x_3 \\ + (\delta_5-\delta_4)\, x_4 + (\delta_6-\delta_5)\, x_5 + (\delta_7-\delta_6)\, x_6 \\ + (\delta_8-\delta_7)\, x_7,$$

$$(7.36) \qquad \begin{aligned} &(1) \quad x_1 \geqslant b_1+5 = 20, \quad \text{that is, } b_1 = 15 \text{ and } x_1 \geqslant 20, \\ &(2) \quad x_2 \geqslant 6, \\ &(3) \quad x_3 \geqslant b_2+5 = 38, \quad \text{that is, } b_2 = 33 \text{ and } x_3 \geqslant 38, \\ &(4) \quad x_4 \geqslant 6, \\ &(5) \quad x_5 \geqslant \max(b_3+5,\, b_4+5) = 38, \quad \text{that is, } x_5 \geqslant \max(38{,}26) \\ &\qquad\qquad = 38, \\ &(6) \quad x_6 \geqslant 6, \\ &(7) \quad x_7 \geqslant b_5+5 = 32, \quad \text{that is, } x_7 \geqslant 32, \\ &(8) \quad x_8 = 0. \end{aligned}$$

Constraint (1) of (7.36) implies that the ordinate of the fencing between δ_1 and δ_2 must run at least 5 m from the centerline of barrack 1 (half-width 3 m). We now have to introduce constraints to indicate that the stock of observation posts is limited. Let us note that if $x_i = x_{i+1}$ there is a pair of observation posts in the portion of fence parallel to Ob of the abscissa δ_i. We introduce integer variables $w_i = 0$ or 1, $i = 1, 2, \ldots, 8$, with $w_i = 0$ if there is no vertical length of fence of abscissa δ_i and $w_i = 1$ if there is such a length. We then add the following constraints:

$$(7.37) \qquad |x_i - x_{i-1}| \leqslant 38 w_i, \qquad i = 1, 2, \ldots, 8.$$

Indeed, in considering constraint (5) of (7.36) while trying to minimize the area of the camp we still find $x_i \leqslant 38$. Also, if $x_i = x_i - 1$, we have $w_i = 1$ in accordance with (7.37), corresponding to a fence along abscissa δ_i. We shall now transform inequalities (7.37) into two inequalities without using absolute values, giving,

$$(7.38) \qquad \begin{array}{lll} (1) & x_i - x_{i-1} \leqslant 38 w_i, & i = 1, 2, \ldots, 8, \\ (2) & x_i - x_{i-1} \geqslant -38 w_i, & i = 1, 2, \ldots. 8. \end{array}$$

The reader can easily verify that (7.38) implies (7.37) and reciprocally. If we have a fence with abscissa δ_i, namely, $w_i = 1$, this will require two observation posts. Hence we have

$$(7.39) \qquad w_1 + w_2 + w_3 + w_4 + w_5 + w_6 + w_7 + w_8 \leqslant r,$$

with $2r$ as the number of available observation posts. The problem of minimizing the camp area compatible with this number is therefore equivalent to minimizing the economic function (0) of program (7.36). In addition, the variables of this program are submitted to constraints (7.38) and (7.39) with $w_i = 0$ or 1. The problem is, in fact, one of programming in mixed numbers with continuous variables x_i and integer variables w_i that can be solved either by Benders's method (see Section 21) or by the *branch and bound* procedure outlined in Section 6.

5. Production Scheduling with Machine Tools

In operations research the words *problems of planning* include a wide range of problems from the evaluation of total time by the PERT method[1] to the most diverse cases of assignment and allocation, as well as those involving the order of precedence for factory-produced goods. We shall now show that this latter type of problem can be reduced to linear programs with mixed

[1] See Volume 2, page 11, and also for more details, see A. Kaufmann and G. Desbazeille, "The Critical Path Method," Gordon & Breach, New York, 1969, page 23.

variables. It will be illustrated by a very simple case, the production of two articles on three machines, but the method is still valid if there are *m* products and *n* machines.

The problem to be considered is the scheduling of production in a workshop and, simple as it is, the reader should find generalizing it an easy matter. Let us therefore take two products, a tube *A* and a sleeve *B*, and three machines, a lathe *T*, a draw plate *F*, and a milling machine *M*. The tube has to be processed in the order $T \rightarrow F$, and *M* is not used in its production; the sleeve is subject to operations in the order $F \rightarrow T \rightarrow M$. Figure 7.6 gives the production time for both products on each of the machines involved, a machine being capable only of processing one article at the same time.

The graph in Fig. 7.7 shows the possible succession of operations, ignoring the condition that each machine can only process one article at a time. From point *a*, representing the start of operations, two arcs are drawn, one for *A* and another for *B*, enabling us to define a single starting point. The arc from *b* to *d* shows the work carried out by *A* on *T*, the arc from *d* to *g* that performed by *A* on *F*, and similarly for *B* on *T*, *F*, and *M*. The production times are shown on all the arcs.

	T	F	M
A	10	2	×
B	2	2	3

FIG. 7.6.
Time of execution in minutes.

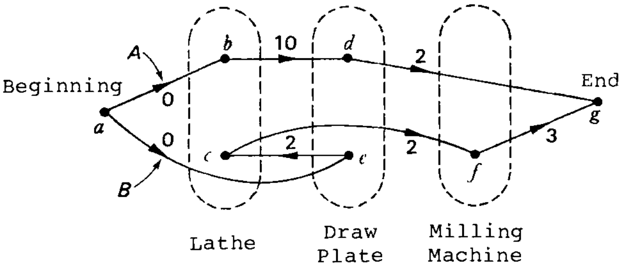


FIG. 7.7

Now let us consider how we can introduce the condition "each machine can only process one article at the same time," a type of condition that is termed *disjunctive*. To do so we use dotted arrows in the graph to indicate the production order for each article on each machine. Given that the milling machine *M* only processes product *B*, there is no alternative for *M*. The four alternatives of production are shown in Fig. 7.8; we have $\{(b, c), (d, e)\}$, $\{(c, b), (e, d)\}$, $\{(b, c),)e, d)\}$, $\{(c, b), (d, e)\}$. However the last case is impossible, since it would entail *A* and *B* being processed on the lathe at the same time. Alternatively, as can be observed from the diagram, product *A*, before it could pass through the lathe, would have to wait until *B* had passed through it, while product *B* on the draw plate machine would have to wait until *A* had been processed, in which case we should have a graph forming a circuit.

If we take the total time as the criterion for the best solution, case 1 gives 19m, case 2 gives 16m and case 3, which is the best, gives 15m.

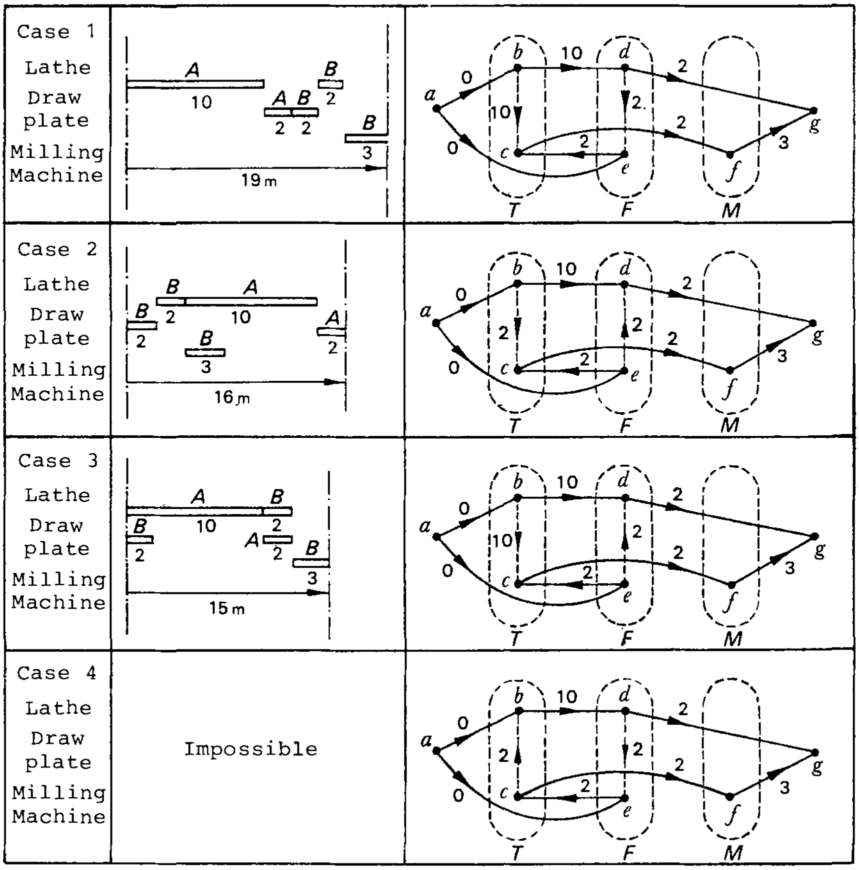


FIG. 7.8

For such problems of arrangement other criteria may be chosen, such as the minimum dead time (idleness of the machines—this obviously being calculated for the same predetermined time interval) or the minimum delay in the delivery date for the product.

We shall now present in analytical form the problem defined by the graph of Fig. 7.7. In this graph the vertices a, b, c, d, e, f, and g represent the conclusions of the operations, and we shall take t_a, t_b, ..., t_g for the instants or dates (in minutes) when these operations end. We have $t_a = 0$ and t_g as the date for the conclusion of all the operations.

This problem of arrangement can now be expressed

$$(7.40)\qquad \begin{array}{ll} (0) & [\mathrm{MIN}]\, F = t_g \\ (1) & t_g - t_f \geqslant 3, \\ (2) & t_g - t_d \geqslant 2, \\ (3) & t_g - t_c \geqslant 2, \\ (4) & t_d - t_b \geqslant 10, \\ (5) & t_c - t_e \geqslant 2, \\ (6) & t_b,\, t_c,\, t_d,\, t_e,\, t_f,\, t_g \geqslant 0, \end{array}$$

that is to say, in the form of a linear program if we do not introduce the disjunctive constraints implying that a machine can only process one article at the same time.

Let us now introduce the disjunctive constraints.

(7.41) For the lathe: $t_c - t_b \geqslant 10$ or $t_b - t_c \geqslant 2$ or exclusive.

(7.42) For the draw plate: $t_c - t_d \geqslant 2$ or $t_d - t_e \geqslant 2$ or exclusive.

The set of relations (7.40)–(7.42) no longer constitutes a linear program on account of the *or exclusive* in the disjunctive constraints.

We shall now reduce these constraints to another form in which the whole program will become one with mixed integer values.

Let us consider the graph of Fig. 7.7 together with the disjunctive constraints, and state

(7.43) $y_1 = 1$, if we impose an arc (b, c) of value 10, that is to say, a linear constraint $t_c - t_b \geqslant 10$;

(7.44) $y_1 = 0$, if we impose an arc (c, b) of value 2, that is to say, a linear constraint $t_b - t_c \geqslant 2$;

(7.45) $y_2 = 1$, if we impose an arc (d, e) of value 2, that is to say, a linear constraint $t_e - t_d \geqslant 2$;

(7.46) $y_2 = 0$, if we impose an arc (e, d) of value 2, that is to say, a linear constraint $t_d - t_e \geqslant 2$.

Let us introduce a constant M large enough to be always a priori greater than t_g (for instance, $M = 1000$).

Now let us replace (7.41) by

$$t_c - t_b \geqslant 10 + M(y_1 - 1), \tag{7.47}$$

$$t_b - t_c \geqslant 2 - My_1 \tag{7.48}$$

and let us verify that we indeed obtain (7.41) if we take $y_1 = 0$ or 1. In the former case we have

$$t_c - t_b \geqslant 10 - M \quad \text{(still verified)}, \tag{7.49}$$

$$t_b - t_c \geqslant 2 \quad \text{(only the right-hand constraint in (7.41) remains)}. \tag{7.50}$$

If we take $y_1 = 1$, it follows that

$$t_c - t_b \geqslant 10 \quad \text{(this is the left-hand constraint in (7.41))}. \tag{7.51}$$

$$t_b - t_c \geqslant 2 - M \quad \text{(still verified)}. \tag{7.52}$$

Hence (7.47) and (7.48) replace (7.41).

Similarly (7.42) will be replaced by

$$t_e - t_d \geqslant 2 + M(y_2 - 1), \tag{7.53}$$

$$t_d - t_e \geqslant 2 - My_2 . \tag{7.54}$$

Finally program (7.40)–(7.42) will be replaced by

$$(7.55) \qquad \begin{array}{rl}
(0) & [\text{MIN}]\, F = t_g \,, \\
(1) & t_g - t_f \geqslant 3 , \\
(2) & t_g - t_d \geqslant 2 , \\
(3) & t_f - t_c \geqslant 2 , \\
(4) & t_d - t_b \geqslant 10 , \\
(5) & t_c - t_e \geqslant 2 , \\
(6) & t_c - t_b - M(y_1 - 1) \geqslant 10 , \\
(7) & t_b - t_c + My_1 \geqslant 2 , \\
(8) & t_e - t_d - M(y_2 - 1) \geqslant 2 , \\
(9) & t_d - t_e + My_2 \geqslant 2 , \\
(10) & t_b,\, t_c,\, t_d,\, t_e,\, t_f,\, t_g \in \mathbf{R}^+ , \\
(11) & y_1,\, y_2 \in \{0, 1\} ,
\end{array}$$

that is, a program with mixed variables, the t terms being continuous and the y terms bivalent.

This method of treating such a type of scheduling problem can be generalized.[1] Of course the number of supplementary constraints and of bivalent variables may increase somewhat quickly, but this is to be expected in combinatorial problems of this kind.

It should be noted that there are other methods of solving such problems. In particular, where they are restricted to two machines and n products or m machines and two products, there is a neat analytical procedure introduced as long ago as 1954 by S. M. Johnson,[2] and in certain conditions this is also valid if $m = 3$ and n is unlimited, and vice versa. As we have shown, the method explained above can be used whatever the values of m and n, but a computer of large dimensions is needed as soon as m and n undergo a substantial increase.

Section 8. **Practical Cases**

1. Observations Concerning Case Studies

From considerations of space we shall not give the full details of the three practical cases for which we propose to explain the models, and we have restricted ourselves to what is needed for the optimal calculations on a computer. Each of these concrete problems is, in fact, a case study and deserves a much longer presentation, but the references given should enable the more curious reader to examine such studies in greater depth. While some nonspecialist readers may consider them somewhat esoteric, it must be remembered that the most difficult tasks for an engineer are those practical problems that it is very often impossible to treat fully by analytical methods on account of their combinatorial complexity. It becomes necessary for the engineer to separate them into a set of subproblems that are smaller and more clearly defined and therefore easier to solve. It is by such means that the problem of the assignment of air crews has been divided into two problems, one of generating rotations and the other of choosing the optimal rotation. The second case is somewhat technical, but we have tried to render it comprehensible for the reader who has already been introduced to a printed circuit card for a computer.[3] We have divided the problem of the conception of these cards into a series of simpler subproblems that can be solved on a computer.

It may be observed that operations research uses methods that can be applied to the problems of every science and technique, whether these prob-

[1] See the article by J. F. Raymond, An Algorithm for the Exact Solution of the Machine Scheduling Problem, IBM New York Scientific Center, Rep. 320-2925, Jan. 1968.

[2] See [K53] and also A. Kaufmann and R. Faure, "Introduction to Operations Research," Academic Press, New York, 1968, pages 264–274.

[3] The majority of our readers have certainly visited a computer center together with the maintenance workshop where dismantled circuit cards can often be seen.

lems are of a normative character (that is to say, must result in choices), or of a nonnormative one (that is to say, must provide solutions). Problems with integer or mixed values are to be encountered wherever a solution has to be found with denumerable means and discrete characteristics.

2. The Problem of Crew Assignment for a Commercial Airline[1]

The reader of Volume 1 has already been introduced to this problem in a very simplified form (page 64) where it was used to illustrate possible methods for solving problems of assignment. We now return to this problem in a form more closely allied to the complex realities of industry and economics.

The restriction of the problem in Volume 1 to two cities meant that it bore little relation to the complex networks of the major airlines. Indeed, such companies make use of the latest computers and of teams using the most advanced operations research techniques.

Before explaining all the practical and concrete details for this type of problem, we propose to give an intermediate problem between the elementary case in Volume 1 and the extremely diversified and highly combinatorial cases that are encountered in practice.

Let us consider an airline with a transportation network comprising seven cities: A, B, C, D, E, F, and G. Its air fleet includes two types of plane, a middle-distance type and a long-distance one, and we will suppose that the aircrews have specialized training for one of the two types (see Fig. 8.1). In our example we shall concentrate on the middle-distance flights between cities A, B, C, D, and F.

We shall use the term *connection* or *junction* for a flight from city X to city Y on a given day with specified times of departure and arrival (example: flight X to Y leaves at 8:30, arriving 9:45). A connection may be formed from several elementary connections with intermediate stages, and will then be termed *composite*. A *segment* will designate the simplest connection taken into consideration when planning flights. A *rotation* is a circuit that leaves one city and returns to it after traversing several segments. The point of departure and of return of a rotation is called the *base*. It is the place at which the crew making the rotation on the same plane or on a different plane of the same type is understood to be stationed.

After leaving the base assigned to it, an aircrew carries out a rotation that fulfills the connections printed in the company's time table. It sometimes happens, for example, in order to complete a rotation or to take on passengers at a further and later point, that a crew may traverse a section without anyone but themselves on board.

[1] The present study is indebted on various fundamental points to the works and thesis of M. Hervé Thiriez [K24]. We warmly appreciate M. Thiriez's permission to make use of it.

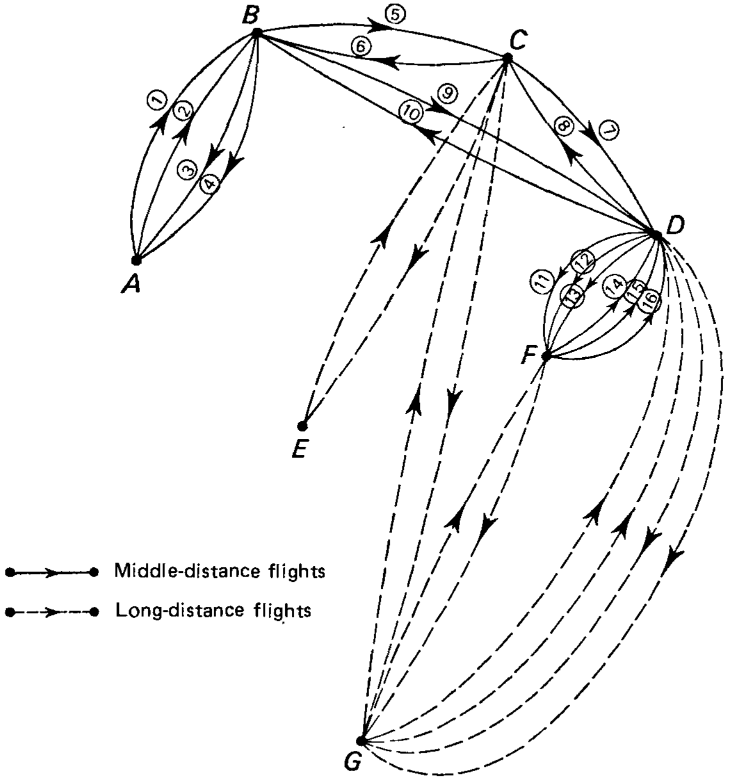


FIG. 8.1

The work of an aircrew can be divided into two parts: in flight and on the ground, before and after flight. The period during which the crew is flying without rest will be called the *period of effective activity*. By the *period of operational activity* we mean the period of effective activity with the addition of the *briefing* time lasting about an hour preceding take off, together with the period of some 15 minutes for *debriefing* after landing.[1]

Let us now return to the small-scale problem illustrating these explanations and let us suppose that the time table for the middle-distance flights shown on the network of Fig. 8.1 is given in Fig. 8.2. The graph of Fig. 8.1 is, in accordance with the terminology given in Volume 2, page 247, a three-mapped graph, since there are never more than three arcs connecting any two vertices (there are three arcs from *D* toward *F* and three from *F* toward *D*). Each arc constitutes a segment, and here all the segments are connections. For each arc or segment the departure and arrival times are given in Fig. 8.2 with a base of 5 minutes.

[1] The rules for operational activity may vary according to the airline.

Number of flight	Departure point	Time of departure	Arrival point	Time of arrival	Duration of flight
①	*A*	08.00	*B*	09.00	01.00
②	*A*	15.35	*B*	16.40	01.05
③	*B*	10.20	*A*	11.15	00.55
④	*B*	19.30	*A*	20.25	00.55
⑤	*B*	10.30	*C*	11.55	01.25
⑥	*C*	12.25	*B*	13.55	01.30
⑦	*C*	13.30	*D*	15.15	01.45
⑧	*D*	20.00	*C*	21.45	01.45
⑨	*B*	10.15	*D*	12.05	01.50
⑩	*D*	13.50	*B*	15.50	02.00
⑪	*D*	07.10	*F*	08.10	01.00
⑫	*D*	12.55	*F*	13.55	01.00
⑬	*D*	21.20	*F*	22.20	01.00
⑭	*F*	09.00	*D*	09.55	00.55
⑮	*F*	16.00	*D*	17.00	01.00
⑯	*F*	20.35	*D*	21.35	01.00

FIG. 8.2

The first problem for the planners is to determine all the possible circuits or rotations in the p-mapped graph that forms the network. Numerous methods exist for their enumeration or denumeration, of which we shall summarize one, termed *latin multiplication*,[1] which was explained in Volume 2, page 271. To carry out the enumeration in this case we shall, while taking into account the fact that 8.1 is a three-mapped and not a one-mapped graph, enumerate the circuits as if it were a one-mapped graph in which all the arcs between two vertices are merged. It is by this method that all the rotations that do not include more than four segments or arcs have been enumerated, the restriction to four arising from professional and not mathematical reasons. The rotations containing only two segments have been set aside and the 44 rotations with three or four segments have been retained. In Fig. 8.4 we show the segments belonging to each connection, and in this way we obtain a Boolean matrix that enables us to discover the rotations that permit us to carry out the segments of the time table. However, each rotation has a cost that may include a large number of factors: salaries, displacement expenses, hotel bills, compensation for absence, and so forth. In this way we obtain an

[1] In practice other methods are often used for such problems because of a large number of professional or technical constraints that make it possible to reduce the enumeration. The airline companies have, in fact, perfected special procedures for enumerating rotations.

(a)

	A	B	C	D	F
A		AB			
B	BA		BC	BD	
C		CB		CD	
D		DB	DC		DF
F				FD	

○

	A	B	C	D	F
		B			
	A		C	D	
		B		D	
		B	C		F
				D	

=

	A	B	C	D	F
	ABA		ABC	ABD	
		BAB BCB BDB	BDC	BCD	BDF
	CBA	CDB	CBC CDC	CBD	CDF
	DBA	DCB	DBC	DBD DCD DFD	
		FDB	FDC		FDF

(b)

	A	B	C	D	F
A	ABA		ABC	ABD	
B		BAB BCB BDB	BDC	BCD	BDF
C	CBA	CDB	CBC CDC	CBD	CDF
D	DBA	DCB	DBC	DBD DCD DFD	
F		FDB	FDC		FDF

○

	A	B	C	D	F
		B			
	A		C	D	
		B		D	
		B	C		F
				D	

=

	A	B	C	D	F
		ABAB ABCB ABDB	ABDC	ABCD	ABDF
	BABA BCBA BDBA	BDCB BCDB	BABC BCBC BDBC BCDC	BABD BCBD BDBD BDCD BDFD	BCDF
	CDBA	CBAB CBCB CDCB CBDB	CDBC CBDC	CDBD CBCD CDCD CDFD	CBDF
	DCBA	DBAB DBCB DBDB DCDB DFDB	DCBC DBDC DCDC DFDC	DCBD DBCD	DBDF DCDF DFDF
	FDBA	FDCB	FDBC	FDBD FDCD FDFD	

FIG. 8.3. Enumeration of the rotations without constraints. For simplification the set of the segments has been reduced; for instance, between *A* and *B* where there should be two, only one is shown.

(a) Rotations with two segments: (*ABA*), (*BAB*), (*BCB*), (*BDB*), (*CBC*), (*CDC*), (*DBD*), (*DCD*), (*DFD*), (*FDF*), that is, ten rotations.

(b) Rotations with three segments: (*BDCB*), (*BCDB*), (*CDBC*), (*CBDC*), (*DCBD*), (*DBCD*), namely, six rotations.

(c)

	A	B	C	D	F
A	ABAB ABCB ABDB	ABDC	ABCD	ABDF	
B	BABA BCBA BDBA	BDCB BCDB	BABC BCBC BDBC BCDC	BABD BCBD BDBD BDCD BDFD	BCDF
C	CDBA	CBAB CBCB CDCB CBDB	CDBC CBDC	CDBD CBCD CDCD CDFD	CBDF
D	DCBA	DBAB DBCB DBDB DCDB DFDB	DCBC DBDC DCDC DFDC	DCBD DBCD	DBDF DCDF DFDF
F	FDBA	FDCB	FDBC	FDBD FDCD FDFD	

○

A	B	C	D	F
	B			
A		C	D	
	B		D	
	B	C		F
			D	

=

	A	B	C	D	F
A	ABABA ABCBA ABDBA				
B		BABAB BCBAB BDBAB BABCB BCBCB BDBCB BCDCB BABDB BCBDB BDBDB BDCDB BDFDB			
C			CBABC CBCBC CDCBC CBDBC CDBDC CBCDC CDCDC CDFDC		
D				DBABD DBCBD DBDBD DCDBD DFDBD DCBCD DBDCD DCDCD DFDCD DBDFD DCDFD DFDFD	
F					FDBDF FDCBF FDFDF

FIG. 8.3 (*continued*)

(c) Rotations with four segments: (*ABABA*), (*ABCBA*), (*ABDBA*), (*BABAB*), (*BCBAB*), (*BDBAB*), (*BABCB*), (*BCBCB*), (*BDBCB*), (*BCDCB*), (*BABDB*), (*BCBDB*), (*BDBDB*), (*BDCDB*), (*BDFDB*), (*CBABC*), (*CBCBC*) (*CDCBC*), (*CBDBC*), (*CDBDC*), (*CBCDC*), (*CDCDC*), (*CDFDC*), (*DBABD*), (*DBCBC*), (*DBDBD*), (*DCDBD*), (*DFDBD*), (*DCBCD*), (*DBDCD*), (*DCDCD*), (*DFDCD*), (*DBDFD*), (*DCDFD*), (*DFDFD*), (*FDBFD*), (*FDCBF*), (*FDFDF*), namely, 38 rotations.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
	BDCB	*BCDB*	*CDBC*	*CBDC*	*DCBD*	*DBCD*	*ABABA*	*ABCBA*	*ABDBA*	*BABAB*	*BCBAB*	*BDBAB*	*BABCB*	*BCBCB*	*BDBCB*	*BCDCB*	*BABDB*	*BCBDB*	*BDBDB*	*BDCDB*	*BDFDB*	*CBABC*	*CBCBC*	*CDCBC*	*CBDBC*	*CDBDC*	*CBCDC*	*CDCDC*	*CDFDC*	*DBABD*	*DBCBD*	*DBDBD*	*DCDBD*	*DFDBD*	*DCBCD*	*DBDCD*	*DCDCD*	*DFDCD*	*DBDFD*	*DCDFD*	*DFDFD*	*FDBDF*	*FDCBF*	*FDFDF*
①							1	1	1	1	1	1	1				1					1								1														
②							1			1																																		
③							1			1																																		
④							1	1	1	1	1	1	1				1					1								1														
⑤		1	1			1		1			1		1									1																						
⑥	1			1	1			1			1		1									1																						
⑦		1	1			1																																						
⑧	1			1	1																																							
⑨	1			1	1				1			1					1													1												1		
⑩		1	1			1			1			1					1													1												1		
⑪																																									1			
⑫																																												1
⑬																																									1	1		1
⑭																																									1	1		
⑮																																									1			1
⑯																																												1
	★	★	★	★	★	★	★	★	★	★	★	★	★				★					★								★											★	★		★

FIG. 8.4. *Note.* Certain rotations, such as (*ABABA*) for example, may borrow different segments; thus we can take (*AB*) = ① or (*AB*) = ②, and in like manner (*BA*) = ③ or (*BA*) = ④, which gives four possible rotations. To have given all the variations would have required a much larger figure exceeding the size of these pages. Hence we have eliminated those not compatible with professional rules, conditions of lodging, and other restrictions. Only those marked with a star have been retained.

economic function and a program with integer values:

$$
\begin{aligned}
&[\text{MIN}]\ z = c_1x_1 + c_2x_2 + \dots + c_nx_n, \\
&\qquad\qquad\qquad\qquad\qquad\qquad n = \text{number of rotations}, \\
&a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \geqslant 1, \\
&a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \geqslant 1, \\
&\dots\dots\dots\dots\dots\dots\dots\dots \\
&a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \geqslant 1, \\
&\qquad\qquad\qquad\qquad\qquad\qquad m = \text{number of segments}, \\
&x_j = 0 \text{ or } 1, \qquad j = 1, 2, \dots, n, \\
&c_j \in \mathbf{R}^+, \qquad j = 1, 2, \dots, n, \\
&a_{ij} = 0 \text{ or } 1, \qquad i = 1, 2, \dots, m, j = 1, 2, \dots, n.
\end{aligned}
\tag{8.1}
$$

It must be observed that the first and second members of the constraints must be connected by the sign $\geqslant$ and not by the sign $=$, since a crew may return to their base as passengers in another plane if the planning program requires this exceptional procedure.

Finding a solution to the system of constraints of (8.1) while ignoring the economic function can be reduced to the very well-known problem in the theory of graphs, that of finding the cover for a simple graph.[1] This is carried out by means of a transportation network associated with the simple graph, and an algorithm derived from that of Ford–Fulkerson is employed.

Thus a solution corresponding to Fig. 8.4 would be

rotation (*BDCB*) that ensures segments ⑥, ⑧, and ⑨,
rotation (*BCDB*) that ensures segments ⑤, ⑦, and ⑩,
rotation (*ABABA*) that ensures segments ①, ②, ③, and ④,
rotation (*DFDFD*) that ensures segments ⑪, ⑬, ⑭, and ⑮,
rotation (*FDFDF*) that ensures segments ⑫, ⑬, ⑮, and ⑯.

It will be observed that this solution duplicates flights 13 and 15, and rotation *FDFDF* will obviously be reduced to *FDF*, since it is uneconomic to carry a crew as passengers, although it may be necessary to do so with other solutions.

However, among all the covers some are better than others when compared with the criterion provided by the economic function of the costs. As a result we are faced with a problem based on a program with integer and even bivalent

[1] See, for example, [K18]. The concept of a minimal cover is derived from the search for a cover that contains a minimal number in the simple graph.

values for which various algorithms are available: implicit enumeration, branch and bound, groups of Gomory cuts, and so on. Before giving some fuller details on this subject, we shall examine all the practical operations that the planning department must undertake by mental calculation, by computer, and by more or less heuristic procedures to solve the very large combinatorial problems that occur in every important airline.

Generating the Segments

Commencing with the connections, next considering the stages, and eventually including the technical stages, the planner has to formulate the list of segments, some of them formed by simple and others by composite connections. In certain cases, as we have seen, a crew may be obliged to return to its base as passengers instead of in an active capacity, for instance, if it has accomplished a flight of 8–10 hours and has to wait several days to resume its duties in the opposite direction. We mention this to show that a considerable period of preparation is required to determine those segments that should be included in the planning.

Generating the Rotations

This problem is that of enumerating all the circuits of a p-mapped graph in which the arcs are the segments and the vertices represent the cities. There are a number of algorithms available for such an enumeration, and we have already explained, for example, how latin multiplication can be used. But in the usual network of cities served by a large airline several million possible rotations could be defined and it would be a tiresome, costly, and pointless task to enumerate all of them. To reduce the number of rotations to be enumerated and evaluated against a criterion of cost, we introduce a priori conditions to determine whether a rotation is to be considered. These conditions include very numerous and sometimes complicated constraints of a technical, trade union, or other kind, and enable us to discard in advance entire classes of rotations that do not satisfy them. In addition, the function of cost attached to each rotation permits us to discard other classes that seem to have little chance of forming part of an optimal solution, and by these means the number of rotations included in the program of optimization can be greatly reduced. To give an idea of the possible degree of reduction, an American company was able to reduce the number of rotations to be considered for optimization from 2.10^6 to 4.10^3. Certain airlines employ instruments to effect this reduction. In addition, the complexity of the cost functions to be introduced leads to the use of a program of evaluation starting with the complex elements included in the cost.

Let us first examine the form in which the cost is commonly structured, although different companies use varied methods to establish the cost, and certain bonuses and indemnities must be added to the fixed salaries. Here, for example, is a formula used by several American airlines:

$$(8.2)\qquad FC(R) = \left[\frac{TAFB(R)}{3.5} - FT(R)\right] \vee \left[\sum_{i=1}^{N(R)} P(i)\right] \vee 0$$

where

$$(8.3)\qquad P(i) = \left[\frac{DT(i)}{2} - FT(i)\right] \vee (240 - FT(i)) \vee 0$$

the symbol $\vee$ signifying *maximum of* and meaning that among the three terms of $FC(R)$ and $P(i)$ the largest term is taken.

The meaning of the other symbols is as follows:

$FT(R)$: total flight time of rotation R,
$TAFB(R)$: time away from base of rotation R,
$FC(R)$: flight time credit of rotation R,
$N(R)$: number of days of rotation R,
$FT(i)$: flight time on day i of rotation R,
$DT(i)$: duty time on day i of rotation R.

All these periods are expressed in minutes.

The following conditions are associated with formulas (8.2) and (8.3): a minimum daily credit of four hours flying that is at least half that of a period of activity and a total time credited that is at least 2/7 of the periods of absence from base.

It goes without saying that this formulation may have to be revised if a new contract between the unions and the management is negotiated.

For many airlines the cost of a rotation is the sum of the costs fixed by contract to which must be added hotel and transportation expenses if one or more nights have to be spent away from base. Special regulations may also apply to certain countries in which stages occur, but we do not propose to enter into all such details.

With the aim of eliminating a large number of rotations that obviously have little chance of appearing in an optimal solution, we use the criteria of elimination for the matrices $[A]$ with elements a_{ij}, $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$ in (8.1) of which Fig. 8.4 gives an example. Let us examine some of these criteria.

1. If a rotation R_k is contained in a rotation R_l and if $c_k \geqslant c_l$, we eliminate R_k, which cannot belong to an optimal solution. (This reduction is only valid if we resolve the system $[A].[x] \geqslant [1]$.)

2. If there is a set of rotations $\{R_{k1}, R_{k2}, \ldots, R_{kr}\}$ such that, for a rotation R_l, we have

$$R_{k_1} \cup R_{k_2} \cup \ldots \cup R_{k_r} \subset R_l \quad \text{and} \quad c_l \leqslant \sum_{i=1}^{r} c_{k_i},$$

we can then eliminate the r columns $R_{k1}, R_{k2}, \ldots, R_{kr}$ with the same proviso as in criterion 1.

3. If the system of constraints

$$\underset{m\times n}{[A]}.\underset{n\times 1}{[x]} \geqslant \underset{m\times 1}{[1]}$$

defined by (8.1) can be reduced to

$$\underset{m\times n}{[A]}.\underset{n\times 1}{[x]} = \underset{m\times 1}{[1]} ,$$

the following rule for elimination can be used: if a line S_i of matrix $[A]$ is contained by its coefficients of 1 in a line S_k, then all the columns belonging to $S_k \cap \bar{S}_i$ may be suppressed. ($\bar{S}_i$ is the complementary line of S_i; if a_{ij} in S_i has a value of 1, then $\bar{a}_{ij}$ has a value of 0, and conversely).

4. In the case where $[A].[x] = [1]$, if there is a set of p lines S_{i1} and a line S_k such that $\{S_{i1}, S_{i2}, \ldots, S_{ip}\} \subset S_k$, then we can eliminate this set of lines as well as the columns for which S_k has a 1 and $\{S_{i1}, S_{i2}, \ldots, S_{hp}\}$ does not have one.

These four procedures do not affect the search for an optimum, since none of the rotations that are eliminated can belong to an optimal solution and the lines discarded by criteria 3 and 4 represent redundant constraints.

It may happen that the eliminations thus effected do not reduce the columns of matrix $[A]$ sufficiently for a computer to be used, and we are sometimes obliged to introduce heuristic rules of reduction that do not guarantee that an optimal solution has not been excluded. Such rules include the elimination of all rotations for which the period of activity contains less than four hours flight time or of those for which the ratio of flight time to time of activity is less than a given number, and they vary according to the airline concerned. Obviously we must also include the cost of the elimination procedures that may result in prohibitive computer costs not compensated for by the operative savings. As always, the evaluation of cost-efficiency must be considered in the use of the computer.

Supplementary Constraints

In theory the crews could be required to stay wherever it would prove least costly to the airline, but practical considerations usually prevent this. At each base B_α there are a number of available crews V_α, and the problem of assignment will be complicated in the following manner. To the m constraints for n variables of the type $[A].[x] \geqslant [1]$ or $[A].[x] = [1]$, we must add r new constraints called *constraints for the availability of the bases.*

$$\underset{r\times n}{[G]}.\underset{n\times 1}{[x]} \leqslant \underset{r\times 1}{[g]},$$

where

$$(8.4)\qquad [G] = \begin{bmatrix} G_{11} & G_{12} & \cdots & G_{1n} \\ G_{21} & G_{22} & \cdots & G_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ G_{r1} & G_{r2} & \cdots & G_{rn} \end{bmatrix},$$

$$(8.5)\qquad [g] = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_r \end{bmatrix},$$

where G_{ij}, $i = 1, 2, ..., r$; $j = 1, 2, ..., n$ represents the potential in hours of flight from base i to perform rotation j, while g_i, $i = 1, 2, ..., r$ represents the total potential available at base i in hours of flight. $G_{ij} \in N$ and $g_i \in N$ (the potential having been evaluated in minutes or, if necessary, in hours or any other basic time). Hence matrices (8.4) and (8.5) are no longer Boolean. Finally, the Boolean constraints on x_i, $i = 1, 2, ..., n$ must not be forgotten.

It is obviously possible that no solution will be found with the crews remaining at their bases and that they will have to be transferred with consequent supplementary expense, another special case to be considered in this problem.

Planning Period

As we have stated, planning for the crews is formulated for intervals of a week or a month or even for longer periods, the times for formation, recycling, and rest being included, in addition to cases of replacement and so forth. We must consider as distinct rotations those that contain the same number of segments but that take place on different dates. The potential availability in flight hours also intervenes in this larger and more concrete formulation. If the combinatorial problem is extended, for instance, over a period of a month, it must take account of the date on which a segment will be traversed. As a result the potential in flight hours available at a base can be used on certain days and not on others, depending on the solutions selected.

Daily Constraints

In the concrete problem it is necessary to add other constraints, one for each base and for each day for the period of rotations being planned. Thus with r bases and 30 days there will be 30 × r more constraints that will take the form

$$(8.6)\qquad [H]_{s\times n} \cdot [x]_{n\times 1} \leqslant [h]_{s\times 1}.$$

where $H_{ij} = 1$, $i = 1, 2, \ldots, s$; $j = 1, 2, \ldots, n$, if on a specified day, a specified base can provide an aircrew for rotation j, and $H_{ij} = 0$ in the contrary case.

At this stage of the construction of a model for the assignment of crews we proceed from (8.1) to the following program:

$$
\begin{gathered}
[\mathrm{MIN}]\ z = c_1x_1+c_2x_2+\ldots+c_nx_n \\
m\begin{cases}
a_{11}x_1+a_{12}x_2+\ldots+a_{1n}x_n \geqslant 1, \\
a_{21}x_1+a_{22}x_2+\ldots+a_{2n}x_n \geqslant 1, \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
a_{m1}x_1+a_{m2}x_2+\ldots+a_{mn}x_n \geqslant 1,
\end{cases} \\
r\begin{cases}
G_{11}x_1+G_{12}x_2+\ldots+G_{1n}x_n \leqslant g_1, \\
G_{21}x_1+G_{22}x_2+\ldots+G_{1n}x_n \leqslant g_2, \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
G_{r1}x_1+G_{r2}x_2+\ldots+G_{rn}x_n \leqslant g_r,
\end{cases} \\
s\begin{cases}
H_{11}x_1+H_{12}x_2+\ldots+H_{1n}x_n \leqslant h_1, \\
H_{21}x_1+H_{22}x_2+\ldots+H_{2n}x_n \leqslant h_2, \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
H_{s1}x_1+H_{s2}x_2+\ldots+H_{sn}x_n \leqslant h_s.
\end{cases}
\end{gathered}
\tag{8.7}
$$

$$
\begin{array}{lll}
x_j = 0 \text{ or } 1, \quad j = 1, 2, \ldots, n & & \\
a_{ij} \in \{0, 1\}, & i = 1, 2, \ldots, m, & j = 1, 2, \ldots, n, \\
G_{ij} \in \mathbf{N}, & i = 1, 2, \ldots, r, & j = 1, 2, \ldots, n, \\
H_{ij} \in \{0, 1\}, & i = 1, 2, \ldots, s, & j = 1, 2, \ldots, n, \\
g_i \in \mathbf{N}, & i = 1, 2, \ldots, r, & \\
h_i \in \mathbf{N}, & i = 1, 2, \ldots, r. &
\end{array}
$$

The indices i of a_{ij}, G_{ij}, H_{ij} do not apply to the same concepts. Hence the above program in bivalent variables includes $m+r+s$ constraints for n variables.

Availability of Crews at Their Bases

Obviously aricrews are not always available since account has to be taken of their rest (activity days and rest days), vacations, recycling periods, and training times with new planes. Aircrews have contracts with their companies that guarantee their conditions of work, salaries, and security. While considering the required number of rotations from each base, it is necessary to ensure that these are compatible with the regulations in the contracts. This is another combinatorial problem distinct from the one that we have explained, though attempts have been made to treat them at the same time. The search for the solution to the operations rotations can be performed mentally or by means of appropriate softwares that are generally constituted by heuristic programs. It has also been suggested that integer programs might be optimized, but from practical experience the cost of treating these programs on a computer is considered exorbitant in relation to the value obtained from their optimization. Besides, these programs are multicriterion and optimization does not always make sense under these considerations.

In addition, the problem of crew availability is complicated by other factors. Members of a crew may become ill, and accidents or incidents of all kinds may occur, so that there is scarcely a day on which the potential is assured. In order, therefore, to ensure the orderly continuation of flights, it becomes necessary to form a reserve of air crews. Finally, airlines are organizing an increasing number of charter flights for which the reserve aircrews are often used. The above are other factors that must be introduced into the model.

As we shall see and, as we may have already suspected, the general model for this problem is far from a simple one. Nearly all the large airlines now plan the rotation of crews with the help of models produced by operations research and of computers, the latter being among the most powerful available in view of the size of the programs and the fact that these are in integer values.

Finally, the number and nature of the crews vary from one cyclical rotation to another, since place and circumstances impose many kinds of variation. There are different regulations for the pilots, the navigators, the flight engineers, and the stewards and stewardesses. Superimposed on the main problem are other problems to be solved by common sense, intuition, and experience. We are concerned with a question of organization in which all the capacities of people and machines have to be integrated in order that the vast system can operate and be adapted and readapted according to the requirements of technological progress and the constant evolution of air transportation.

Let us now consider some aspects of the methods of calculating programs with integer values. We shall differentiate between two types of model.

1. Models of a theoretical type without the supplementary and daily constraints:

$$(8.8) \qquad [\text{MIN}]z = [c]_{1\times n} \cdot [x]_{n\times 1};$$

$$[A]_{m\times n}\cdot[x]_{n\times 1} \geqslant [1]_{m\times 1};$$
$$[c]\in \mathbf{R}^+,\quad a_{ij}=0 \text{ or } 1,\qquad i=1,2,\ldots,m;\ j=1,2,\ldots,m;$$
$$x_j=0 \text{ or } 1,\qquad j=1,2,\ldots,n.$$

2. Practical models that are noticeably more complicated, with the supplementary and daily constraints:

$$\begin{aligned}
&[\mathrm{MIN}]z = [c]_{1\times n}\cdot[x]_{n\times 1};\\
&[A]_{m\times n}\cdot[x]_{n\times 1} \geqslant [1]_{m\times 1};\\
&[G]_{r\times n}\cdot[x]_{n\times 1} \leqslant [g]_{r\times 1};\\
&[H]_{s\times n}\cdot[x]_{n\times 1} \leqslant [h]_{s\times 1};\\
&[c]\in \mathbf{R}^+,\\
&a_{ij}\in\{0,1\}, \qquad i=1,2,\ldots,m;\quad j=1,2,\ldots,n;\\
&G_{ij}\in\mathbf{N}, \qquad i=1,2,\ldots,r;\quad j=1,2,\ldots,n;\\
&H_{ij}\in\{0,1\}, \qquad i=1,2,\ldots,s;\quad j=1,2,\ldots,n;\\
&g_i\in\mathbf{N}, \qquad i=1,2,\ldots,r;\\
&h_i\in\mathbf{N}, \qquad i=1,2,\ldots,s.
\end{aligned}\tag{8.9}$$

The solution of the economic program of (8.8), with which we are less concerned, can be found by methods that are well-established in the theory of graphs:

a. The method termed *covering a simple graph.*

b. The knapsack method, that is, a procedure of Boolean optimization suitable for bivalent programs of this type.

c. Optimization of the flood in a network.

d. Branch and bound method.

e. Heuristic methods that do not always produce an optimal solution.

Space is lacking for the detailed explanation of these methods, but we shall refer the reader to a number of basic works. As an introduction, the general study of the assignment of aircrews, including a comparative appraisal of the results achieved by the different methods, is given in detail in [K70]. The method known as *seeking the optimal cover for a simple graph* can be studied in [K23], [K78], and [K18]; the knapsack method and its variations, as well as the branch and bound method, are given in [K23], while some references to heuristic procedures are made in [K70].

The airplane planners will obviously be more interested in the practical models in which the constraints are appreciably more complicated but which more closely resemble concrete cases. Nevertheless, the simplified models such as (8.1) are important from a methodological standpoint. These are problems in which all the elements of the matrix of constraints are equal to 0 or 1 with an increased percentage of zeros. It has been observed experimentally that Gomory's method (Section 19) rapidly produces a result in this case. This is because the absolute value of any determinant taken from the matrix of constraints of (8.7) is a small one, and the reader may refer to the thesis of H. Thiriez for a fuller explanation of this property. The acquired experience of the last ten years shows, however, that the results obtained from the use of Gomory's method are completely satisfactory only for problems of this type.

Like Gomory's method, that of Lemke–Spielberg, explained in Section 4, yields good results for the problem of (8.1) but is distinctly less successful for a more general problem of this type. Various other methods of enumeration such as those of Balas [K26] or Geoffrion [K35] could be used, but all reveal the above tendency.

Gomory's method of asymptotic programming is given in Section 20; it consists of finding an initial solution by not restricting the variables to integer values and by replacing the constraints $x_j = 0$ or $1, j = 1, 2, \dots, n$ by

$$(8.10)^1 \qquad 0 \leqslant x_j \leqslant 1, \qquad j = 1, 2, \dots, n.$$

In this way we obtain a solution that does not necessarily possess integer values. Starting from the simplex table representing this solution, the method of asymptomatic programming produces a point with integer (but not necessarily nonnegative) components. If they are nonnegative this is the optimal solution, otherwise it is not a solution, since (8.10) is not verified.

Thiriez's procedure [K71] known as *the method of groups of cuts* produces excellent results for problems of this type. It consists of employing an asymptomatic algorithm in two stages: finding the solution of a linear program and then using the result to obtain an optimal solution with integer values. We show the periods of calculation on a large computer of the third generation in Fig. 8.5.

As this table shows, problems of considerable dimensions can be solved very quickly. Trubin's algorithm given in Section 23 is suitable for solving problems such as (8.1) but has not been systematically tested. If the necessary theoretical developments are carried out it could still further reduce the time needed for a solution.

[1] Equation numbers (8.11) and (8.12) omitted in the French edition.

Source of the problem	Dimensions	Type	Time of linear programming	Time after the first stage	Total time
American Airlines	104 × 132	(8,7)	15.6 s	9.6 s	25.2 s
American Airlines	104 × 236	(8,1)	31.8 s	22.8 s	54.6 s
Air France	67 × 536	(8,1)	28.2 s	40.8 s	69 s

FIG. 8.5

3. The Problem of Resistance Chips Placement[1]

This problem, encountered by every manufacturer of computers and printed circuits, will show how far the techniques of operations research have penetrated the domain of the engineer. They are currently employed to optimize various combinatorial problems that occur in the most diverse operations, and it should not be a matter for surprise that a number of algorithms used in CAD[2] closely resemble some with which the readers of the earlier volumes of this work will have become familiar. Thus in the conception of electronic circuits, it is necessary to calculate the length of the longest path between two components of the same circuit. If this length exceeds a figure based on the technology employed, the circuit will not function correctly. In this case we are concerned with the conception of printed circuits using very rapid technology with integrated circuits. The latter are linked by connections traced on printed circuits, and to illustrate the problem an example of an integrated circuit is shown in Fig. 8.6. In this circuit there are 81 possible positions in which chips could be sited, but as a rule they are not all used. These positions are identified by a letter and a number, and to simplify the diagram only ten chips are shown on the plate of the printed circuit. From each chip in our example, eight pins appear, these being the feet that are used as connections, but this number could be much larger where the chips are more complex circuits.

[1] A chip is an integrated circuit in the form of a small flat case of parallelepiped shape from which the connecting wires emerge.

[2] Computer Aided Design or Design Automation (DA). This procedure can be used equally to plan printed circuits or motorways or to calculate the shape of metal girders.

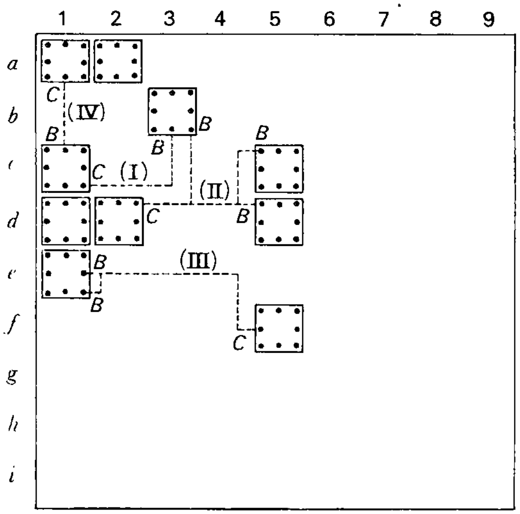


FIG. 8.6

We shall apply the term *network* to a set of pins and the printed wires that link them together electrically.

In Fig. 8.6 only four networks marked (I)–(IV) have been shown, and in general, for electrical reasons, the networks are restricted to some dozen pins. A network requires a pin that represents a receiver of transistors, the other pins in the network being either passive elements (resistances, capacities) or bases for transistors.[1] We have indicated by the letters *C* and *B* the receivers and the bases of the four networks. When the receiver in a network changes polarity, it transmits this change to the other pins of the network by means of the printed connections. In the conception of very rapid circuits, it is important to have short connections,[2] otherwise the speed with which the impulses are propagated would restrict the speed of the computer. In addition, to ensure their correct functioning, these circuits must be *harmonized* by placing in each network a resistance that avoids the impulse reflections that occur in these very rapid circuits. In effect, if receiver *C* of network (II) switches, base *B* of chip *c*5 switches after the time required to propagate this impulse. If this impulse is not absorbed by adding a resistance at the end of the network (the base furthest from the receiver), a part of it is reflected and could again cause the bases of the network to switch. In Fig. 8.7 we have shown a plate on which three special chips comprising resistances only have been added; heavy lines

[1] Specialists and radio "hams" will understand this terminology, but it is not essential for what follows.

[2] In a nanosecond (10^{-9} sec), light theoretically travels 30 cm. In printed circuits, in fact, the electric impulses travel more slowly (only 20 cm each nanosecond).

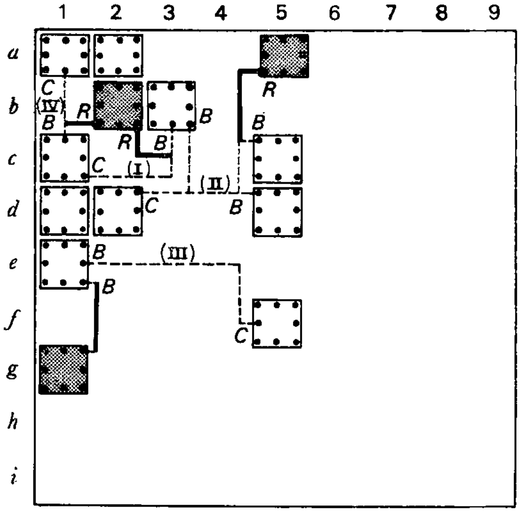


FIG. 8.7

indicate the connections needed to harmonize the four networks with these special chips that are indicated by shading in their temporary arbitrary location. With the chips in position we attempt to connect the end bases of the network to the nearest resistance. But each network must have one with a different harmony without which they would be electrically unified, and the connections must not cross. As may be suspected, this problem includes numerous parameters and an overall solution of the placement of all the ordinary and resistance chips, together with the selection of a resistance for each network and, finally, the tracing of the printed circuit is outside the scope of even the largest and most advanced computer, so that the problem is often divided into four parts:

a. Placement of the ordinary chips

We know the terminals that must be linked electrically and we attempt to find a placement that will avoid those of the same network being too far apart. To effect this, the criterion of the total minimal length of the networks is often chosen. The length of the link between two pins, ultimately to be carried out by a printed circuit containing zigzags, is estimated by the shortest distance separating them, and the length of a network is estimated by the $p(p-1)/2$ distances between its pins.

b. Ordering of the networks

This is the determining of the p straight line connections of a network with p terminals. In the placement stage we should consider that the network included $p(p-1)/2$ connections, which is greater (as soon as $p > 1$) than the minimal number p required to link p points without a loop. In this phase we also seek the base furthest from each receiver in the networks, distance being the sum of the straight line distances between terminals from the receiver to the last base.

c. *Placement of the resistance chips and selection of a resistance for each network*

d. *Tracing the printed circuit*

By now we have determined the placement of all the cases. For some ten years, computer manufacturers have possessed satisfactory heuristic methods for tracing the printed circuits. The exact solution is a problem of programming in integers of immense dimensions, since a variable 0 or 1 corresponds to every possible linkage on the plate.

Here we shall only consider the placement of the resistance chips. For this the reader may consult [K75] and [K76], and for tracing printed circuits [K57]. In Fig. 8.8 we have only traced the final bases of the networks and a certain number of placements available for locating resistance chips. The available placements (those not occupied by ordinary chips) are unshaded. To convey the complexity of the problem we have imagined that there are more final bases than the four shown in Figs. 8.6 and 8.7. In practice there may be up to several thousand final bases, and the solution cannot then be obtained by inspection as in our present example.

In Fig. 8.8 there 28 final bases of transistors represented by heavy dots. Since the resistance chips are somewhat costly components, the aim is to use as few of them as possible. As there are only eight resistances for each chip in our example, we shall need four chips ($4 \times 8 \geqslant 28$) that must occupy the available sites. These four chips are shown in Fig. 8.8, three of them ($b2$, $a5$, $g1$) having already appeared in Fig. 8.7.

Usually the correct placement of these chips can be summarized as follows: locating them in the available placements to effect a resistance in each final base so that the wires do not cross and the total of the straight line distances connecting them is as short as possible. Presented in this form the problem is *multicriterion*, that is to say, it includes two aims that are not necessarily compatible: to minimize the number of straight line connections that cross, so that their final realization by printed circuit will be easier,[1] and to minimize the total length. In Fig. 8.8 we have shown 20 straight line connections that do not cross.

We can, however, simplify this multicriterion problem by using an old theorem of Monge[2] who showed by purely geometric proof that, in the

[1] We may also treat the graph formed by straight line connections as being plane. In that case the criterion of noncrossing becomes a constraint. See the concept of a plane graph in Volume 2.

[2] G. Monge, "Deblais et Remblais," Mémoire de l'Académie des Sciences, 1787.

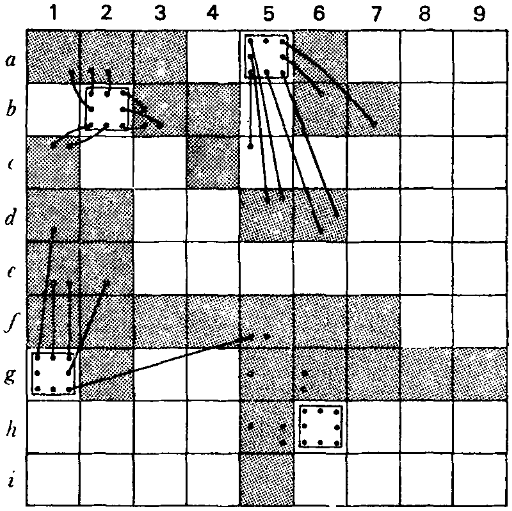


FIG. 8.8

Euclidian plane, for a figure such as Fig. 8.9, we always have

$$a+b \leqslant c+d. \tag{8.13}$$

In other words, by minimizing the total length, we shall a fortiori eliminate any crossing.

Hence we are faced with finding the placement of four chips with eight resistances each, while minimizing the total length of the connections (measured "as the crow flies") between the final bases and the resistances. Let us observe that, if the placement of the resistance chips is known, the problem becomes a generalized problem of assignment[1] that can be solved by means of one of the algorithms given in the first two volumes of this work. Our aim is to assign one and only one resistance to each final base in such a way as to minimize the total length of the connections. The cost of assigning a resistance to a final base is the distance between them.

However this problem can reach considerable proportions, since there may be 2000 or more final bases of networks for a very small-scale technology. It is not then possible to employ the Hungarian method in which the matrix of costs would contain at least 2000 × 2000 elements that would exceed the

[1] In Volume 1, page 68, and Volume 2, page 265, the problem of assignment was discussed, together with the Hungarian method for solving it. Here we are concerned with an extension of the problem, in which a person can perform r tasks. In the present problem the people are the resistance chips and the tasks are the final bases, so that we are confronted by a transportation problem. This type of problem is discussed in Volume 1, page 51.

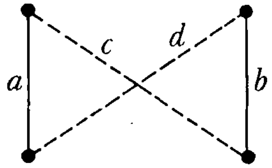


FIG. 8.9

memory range of the very largest computer. In these cases a method adapted to such large-scale problems is used; this will be explained, as well as some results from its use on a computer. Let us first examine the principle of this method.

Since the chips are very small there is little difference in the distance between a final base and a resistance and that between the base and the center of the chip containing the resistance. We consider the problem of assigning a chip to each final base. The same resistance chip could be assigned to eight final bases. In this manner the problem becomes one of assignment with a smaller matrix, namely, 2000 × 2000/8, which could be retained in the memory bank of a large computer, and the value of this optimal solution is not far removed from that of the initial problem (see Figs. 8.10a and b, where the total length of the connections is little different).

Next, we successively solve the assignment problems of reduced dimensions (Fig. 8.10 shows the solution of two such problems). One after the other, each resistance chip and the final bases to which it is assigned (eight at the most) are considered. We solve the problem: to assign each resistance of a chip and one only to each final base. At most the matrix for this problem contains 8 × 8 elements. In fact, a resistance chip can be assigned to less than eight resistances, as shown in the case of the two chips in Fig. 8.10.

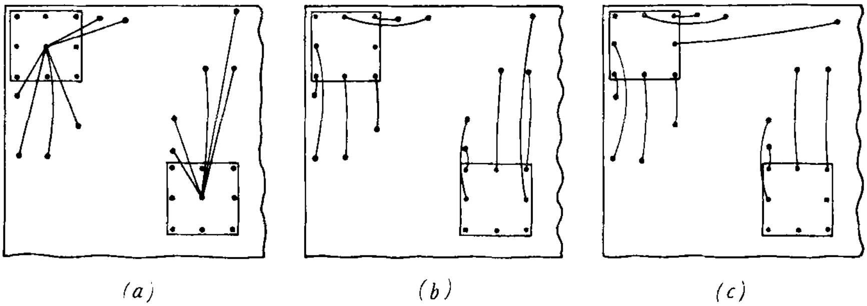


FIG. 8.10

The solution obtained after solving all these small-scale problems closely resembles the one that would have been found if there had been a computer large enough to avoid having to reduce the problem. It is nevertheless possible (see [K78]) to verify whether the solution is optimal and, if not, to obtain this from the approximate solution. We show in Fig. 8.10 the three stages of the algorithm that presupposes the previous placement of the resistance chips.

Once these have been located we are faced with a sequence of assignment problems. Some times required to reach an optimal solution such as that shown in Fig. 8.10c on a powerful third-generation computer are shown in Fig. 8.11.

Number of bases	Number of chips with 8 resistances	Time in seconds
45	10	3.33
88	15	4.97
350	46	41.05
1 000	130	110.5
6 000	190	450

FIG. 8.11. *Note.* Values on the bottom line correspond to chips with 32 resistances.

We shall now see how we can solve the preceding problem of locating the resistance chips so as to minimize the total length of the connections, given that the length of each connection is the straight line distance between a base and the center of a chip. Let us suppose there are m available placements for the chips and n final bases in the network.

Let there be m integer variables with values of 0 or 1:

$$(8.14) \qquad y_i = 0 \text{ or } 1, \qquad i = 1, 2, \ldots, m,$$

with the value $y_i = 1$ we site a chip at placement i and for $y_i = 0$ we do not site one there. Let us now suppose

$$(8.15) \qquad x_{ij} = 0 \text{ or } 1, \qquad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n,$$

with $x_{ij} = 1$, base j is connected to a chip in placement i; for $x_{ij} = 0$ it is the contrary case. Also let

$$(8.16) \qquad c_{ij}, \qquad i = 1, 2, \ldots, n; \quad j = 1, 2, \ldots, m,$$

be the distance between the final base j and the available placement i. We define a positive value M that greatly exceeds the longest possible length of all the connections. Mathematically, the problem of placing the chips is expressed as

$$
(8.17)\qquad
\begin{array}{lll}
(1) & [\mathrm{MIN}]\, z = \sum_{i=1}^{m} \left[M y_i + \sum_{j=1}^{n} c_{ij} x_{ij} \right], & \\
(2) & \sum_{i=1}^{m} x_{ij} = 1\,, & j = 1, 2, \ldots, n\,, \\
(3) & \sum_{j=1}^{n} x_{ij} \leqslant 8 y_i\,, & i = 1, 2, \ldots, m\,, \\
(4) & x_{ij} = 0 \text{ or } 1\,, & i = 1, 2, \ldots, m\,; \quad j = 1, 2, \ldots, n\,, \\
(5) & y_i \;= 0 \text{ or } 1\,, & i = 1, 2, \ldots, m\,.
\end{array}
$$

Constraints (2) of (8.17) show that one resistance and one only must be assigned to each final base of the network. Constraints (3) mean that no base can be connected to a chip in placement i if there is no chip there ($y_i = 0$). They also indicate that if a chip is in this placement ($y_i = 1$), not more than eight final bases can be linked with it. Problem (8.17) is a special example of programming in mixed numbers identical to the problem of locating factories given on page 91. But here we should have a unit cost of construction M that would be the same for all the factories. It is a problem that can be solved by *Benders's method*, described in Section 20.

When the chips have been placed in position we assign each of the final bases to one and only one resistance by means of the special method already described and illustrated in Fig. 8.10. In this we first explained the problem of assigning the final bases to a resistance, and then that of placing the chips containing these resistances. These procedures could have been reversed, but the order was disregarded for instructional reasons.

Some readers may have concluded that, even in problems of substantial dimensions, a good placement could have been obtained by a rapid visual examination. But it should be noted that the procedures for placing the chips and assigning the bases can be integrated in a CAD system; hence, it would be uneconomic to interrupt the entire process of conception of a printed circuit card by carrying out even such a simple operation as the above.

The type of problem described is of concern to all manufacturers of computer components; hence they have studied a great variety of mathematical models, including in them various constraints suitable for the particular technology employed.

4. The Synthesis of Telecommunication Networks

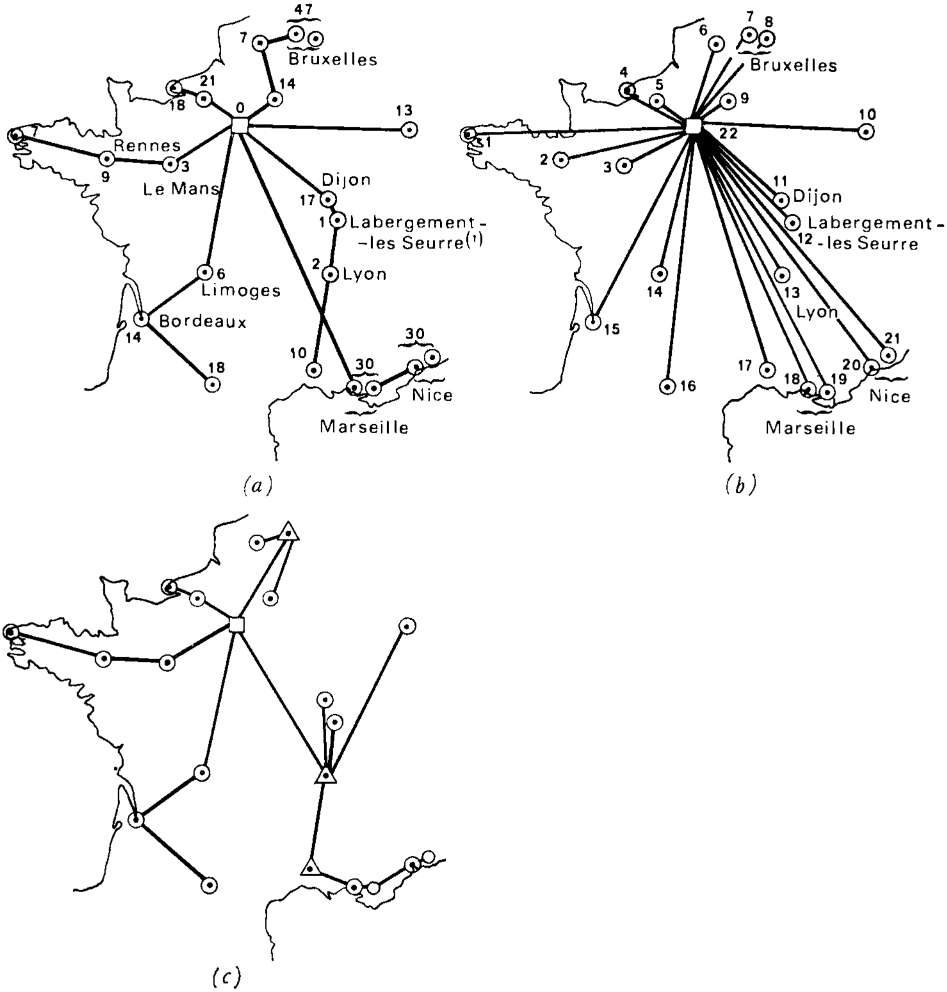
Operations researchers should be called in at the first development stage of a project in order to estimate its cost, to find the best solution, and to consider how the variations of the different parameters will affect this optimal cost. Among the most difficult problems with which they have to contend we may cite the synthesis of telecommunication networks. Given the amount of information to be transmitted between different pairs of points, the requirement is to construct a network of minimal cost while taking the following considerations into account: (1) the cost of the telephone lines used for transmitting information, (2) the cost of the concentrators, (3) the fact that the Post Office (especially in England) offers graded tariffs when lines are rented for 6, 12, or 24 hours.

Another type of problem, involving the analysis of circuits, is where a network is already established and the maximal volume of information that it can transmit has to be calculated, or where it has to be calculated whether this network is capable of satisfying the communication requirements for several pairs of points. This latter problem may be treated as one of linear programming of very large proportions that theoretically can be solved by the algorithm of the simplex or by the algorithms of flows if there are not more than three points between which communications are to be established.[1] The very high costs for privative networks (more than half the operating costs) of certain long distance services make it very important to find the optimal solution.

Some of the different types of network encountered in practice are shown in Fig. 8.12. We have confined ourselves to transmission networks connecting terminals to a central computer in a question-and-answer system, the number of terminals installed in each city being given in the figure. The conception of the network implies the choice of equipment and of the outline and length of the lines, so as to minimize the costs of loan charges (in the case of purchase) or of renting the lines and equipment (modems, concentrators). In the present concrete case only two types of concentrators have been considered; computerized satellites that can be programmed, costing 150,000 F and capable of handling the traffic from 180 terminals, and small cabled concentrators costing 40,000 F, to which 24 terminals can be connected. We presuppose that not more than four concentrators can be arranged in multipoint, that is to say, that they can be attached to the same telephone line.

In our example the problem has been simplified since, in practice, the cost of the given transmission equipment depends on the exact number of terminals, each one having an adapter. To further simplify the example we will suppose that the rental for a good quality telephone line is proportionate to its length and costs 14 F per kilometer. In Fig. 8.12a we show a multipoint network

[1] For this problem of the analysis of networks the reader may consult the article by B. Rothschild and A. Whinston, On Two-Commodity Network Flows, *Journal de l'ORSA*, **14** (3), 377–388, May 1966.



(a)

(b)

(c)

FIG. 8.12. (a) Number of terminals in each of the 20 cities. (Labergement-les-Seurre: Côte-d'Or with 800 inhabitants. In a decentralized framework, this country retreat of a well-known author might be connected to a telecommunication network. We have anticipated this and have included it in the model.)

(b) Enumeration of the cabled concentrators from 1 to 22. There are no concentrators in Paris and number 22 is used to number it.

(c) Key: □ Central computer; △ computerized satellite used as concentrator; ⊙ cabled concentrator.

using cabled concentrators, a solution that is clearly more profitable than the point-to-point network illustrated in Fig. 8.12b. In practice, the fact that the software administration of concentrators in multipoint is more costly than that of concentrators with point-to-point connections should be taken into account. In Fig. 8.12c we show a network using computerized satellites with less cost in lines than that shown in Fig. 8.12a, but with more for the equipment. We have shown only a fraction of the combinations available in practice, and the combinatorial element in this problem is among the greatest. A universal model taking account of every possibility would be altogether too complicated.

The solution of minimal cost is then evaluated for each type of network in Fig. 8.12 or for other types. After examining other factors that cannot be included in the model, such as the cost of the software and the progressiveness of geographical extensions, it is possible to make a choice. The evaluation of the cost of network (b) is obviously easy: enough cabled concentrators are installed in each city to make it possible to connect each terminal to the city (20 cities in Fig. 8.12a and 21 transmitters in Fig. 8.12b), and a line is rented to connect each of these concentrators with Paris. Finding the minimal cost of the multipoint network is considerably more complicated (see Fig. 8.12a).

t_i, $i = 1, 2, \ldots, 21$, the number of terminals of concentrator i. This number t_i is determined by inspection: the terminals are distributed to the cities where there are several concentrators until the maximal capacity of each concentrator is attained.

d_{ij}, $i, j = 1, 2, \ldots, 22$, the cost of a line between concentrators i and j and the central system. If i and j are in the same city, d_{ij} is then very small.

x_{ij}, $i, j = 1, 2, \ldots, 22$, integer values of 0 and 1. If $x_{ij} = 1$, we use the connection between concentrators i and j in the optimal network where concentrator j is the furthest away or *upstream* from the central system in Paris. In the solution corresponding to the optimal network, we still have $x_{ij} \neq x_{j_i}$. For example, in Fig. 8.12a we have $x_{11,12} = 1$, with the numeration given in Fig. 8.12b.

y_i, $i = 1, 2, \ldots, 22$, the number of segments of circuits upstream from concentrator i and connected to it. For example, in Fig. 8.12a, y_{14} (Limoges) $= 2$ and y_{22} (Paris) $= 21$.

The problem of the synthesis with minimal cost of a multipoint network using only cabled concentrators is

$$(8.18) \qquad [\mathrm{MIN}]\ z = \sum_{i=1}^{22} \sum_{j=1}^{22} d_{ij} x_{ij},$$

$$(8.19) \qquad \sum_{j=1}^{21} x_{ij} \leqslant 1, \qquad i = 1, 2, \ldots, 21,$$

$$(8.20) \qquad \sum_{j=1}^{22} x_{ji} = 1, \qquad i = 1, 2, \ldots, 21,$$

Constraints (8.19) imply that exactly 21 segments of circuits are used in the network. Constraints (8.20) imply that there cannot be any Y-shaped junction of the lines in a city that has a cabled concentrator i.

We have also

$$(8.21) \qquad y_i = \sum_{j=1}^{21} x_{ij}(y_j+1), \qquad i = 1, 2, \ldots, 22.$$

For example, in Fig. 8.12a, $y_{18} = 0$ (Toulouse), $x_{14,18} = 1$, and $y_{14} = x_{14,18}(y_{18}+1) = 1$ (Bordeaux); we also have $x_{6,14} = 1$ and $y_6 = x_{6,14} \cdot (y_{14}+1) = 2$ (Limoges). We add the further constraint

$$(8.22) \qquad y_{22} = 21 \quad \text{(Paris)}.$$

In effect (8.19) implies that there cannot be a Y-shaped connection, but there could still be loops in the network. Constraint (8.22) implies that the 21 concentrators are connected to Paris.

We now have

$$(8.23) \qquad y_i \leqslant 3, \qquad i = 1, 2, \ldots, 21.$$

This indicates that we cannot, for technical reasons (loss of power), have more than four concentrators on the same line. We have $y = 3$ (Dijon, see Fig. 8.12a) which means that when Dijon is connected to Paris there are four concentrators in the line, the maximum permitted. Lastly, let us recall that

$$(8.24) \qquad x_{ij} = 0 \text{ or } 1, \qquad i, j = 1, 2, \ldots, 22,$$

$$(8.25) \qquad y_i \in \mathbf{N}, \qquad i = 1, 2, \ldots, 22.$$

The problem of minimizing (8.18) subject to (8.19)–(8.25) is a nonlinear program with integer values. Indeed, constraint (8.21) uses the nonlinear terms $x_{ij} y_j$. There are 484 bivalent variables x_{ij} and 22 integer variables y_i.

This somewhat large-scale program can, however, be solved on the most powerful computers by a variation of the branch and bound method in a time the cost of which will be amply repaid by the savings effected over a period of several years. The reader will realize from this example that the synthesis of circuits provides operations research with some of the best cost–efficiency results. The special network (see Fig. 8.12b) satisfies the constraints of the preceding program with integer values. Nevertheless the network is not one of minimal cost.

We shall now present a mathematical model using an integer program that will enable us to find the type of network shown in Fig. 8.12c; such a network will permit the use of computerized satellites if they reduce the total cost of the network.

For this model we shall define some supplementary variables. Let

$$w_i, \qquad i = 1, 2, \ldots, 21, \qquad i \neq 8, 19, 21,$$

be bivalent variables. If $w_i = 1$ we put a computerized satellite in i; if $w_i = 0$ we install sufficient cabled concentrators at i to serve the terminals. Numbers 18, 19, 21 correspond to the locations of a second cabled concentrator situated either at Brussels, Marseilles, or Nice where there are more than 24 terminals. At Nice, either a computerized satellite ($w_{20} = 0$) or two cabled concentrators can be placed. We shall never install several satellites at Nice or elsewhere and we shall not consider w_{21}. The total cost of the network to be minimized is

$$(8.26) \qquad [\text{MIN}]\ z = \sum_{i=1}^{22} \sum_{j=1}^{22} d_{ij} x_{ij}$$

$$+ 150{,}000 \sum_{\substack{i=1 \\ i \neq 8, 19, 21}}^{20} w_i + 40{,}000 \sum_{\substack{i=1 \\ i \neq 8, 19, 21}}^{20} (1 - w_i)$$

$$+ 40{,}000\,((1 - w_7) + (1 - w_{18}) + (1 - w_{20})).$$

The first term represents the total cost of the lines for the 5-year period of redemption, the stock being presumed to have been purchased. The second term is the purchase price of the computerized satellites, and it should be noted that for cities such as Limoges a single cabled concentrator is sufficient and that there are no satellites. In contrast, if there is no satellite ($1 - w_7 = 1$) for Brussels ($i = 7$), two cabled concentrators are needed, and during the calculations the term 40,000 ($1 - w_7$, the cost of a cabled concentrator) therefore appears twice.

$$(8.27) \qquad \sum_{j=1}^{22} x_{ji} = 1, \qquad i = 1, 2, \ldots, 21,$$

showing that there is only one line leaving each concentrator or satellite.

On the other hand, (8.19) is transformed, since several segments of circuit may be attached to a satellite. In Fig. 8.12c, 3 lines are connected to the satellite at Lyons. We have

$$(8.28) \qquad \sum_{j=1}^{21} x_{ij} \leqslant 1+20w_i, \qquad i = 1, 2, \ldots, 21, \quad i \neq 8, 19, 21,$$

$$(8.29) \qquad \sum_{j=1}^{21} x_{8j} \leqslant 1+20w_7,$$

$$(8.30) \qquad \sum_{j=1}^{21} x_{19j} \leqslant 1+20w_{18},$$

$$(8.31) \qquad \sum_{j=1}^{21} x_{21j} \leqslant 1+20w_{20},$$

In fact, if there is no computerized satellite at i, constraint (8.28) repeats (8.19) and, if there is one, it is without effect. The number 21 has been introduced since there will not be more than 21 nonnull x_{ij} in the optimum. Constraints (8.29)–(8.31) take account of the cities where two cabled concentrators might be needed. We also have the same equation as in (8.21), but (8.23) is transformed into

$$(8.32) \qquad y_i \leqslant 3+21w_i, \qquad i = 1, 2, \ldots, 21; \quad i \neq 8, 19, 21.$$

Indeed, if there is no computerized satellite, we return to (8.23), otherwise this constraint is inoperative (still verified). There are also the special cases:

$$(8.33) \qquad y_8 \leqslant 3+21w_7,$$

$$(8.34) \qquad y_{18} \leqslant 3+21w_{18},$$

$$(8.35) \qquad y_{21} \leqslant 3+21w_{20}.$$

Constraint (8.22) applies as before to Fig. 8.12c; we have $y_{18} = 3$ which satisfies $y_{18} \leqslant 3+21.0$, with $w_{18} = 0$. We also have $y_{13} = 8$ (Lyons) that satisfies (8.32), since $8 \leqslant 3+21$.

The problem of designing a network of minimal cost using satellites and/or cabled concentrators is a program with integer values in which the constraints, with the exception of (8.21), are linear and, as in the preceding example, its solution is obtained by the branch and bound procedure. Indeed, we might only solve the last problem that could have as its solution (if the cost of the computerized satellites is comparatively high) the network shown in Fig. 8.12a.

Part 2. MATHEMATICAL THEORY

Chapter II. ALGORITHMS AND HEURISTICS FOR INTEGER OR MIXED PROGRAMS

Section 9. Introduction

This chapter will obviously prove more difficult to follow, since the principle of operations researchers is to differentiate the motivating from the activating parts. However, much care has been taken over the method of presentation, and the most difficult parts of all have been included in a supplement.

Those engaged in operations research range from the practical industrial user to the advanced university research worker. The concern of the former is to discover whether (and, if so, how) a problem can be programmed; the interest of the latter lies in improving the science and in extending objective knowledge. More and more it becomes of importance that these two classes should communicate with each other, which is what, in a modest degree, we are aiming to achieve.

We shall review the algorithms and heuristics that have been accepted over the last fifteen years as best fitted to solve that special and very important class of combinatorial problems comprising integer or mixed programs. Before doing so, however, it will be useful to consider the respective meanings given to the terms *algorithm* and *heuristic*. The former is a set of rules that enables us by a sequential and rigid method to calculate a solution proposed in advance (to find, for example, the optimal solution or solutions of a program). A heuristic is a set of rules sometimes introduced solely by intuition that enables us to obtain what we consider a priori an acceptable result (otherwise why use the rules?): to find, for example, a "good" solution or solutions to a program.

These two procedures are supplementary rather than contradictory. Numerous algorithms were heuristics before being improved until the rules became a rigid procedure. Equally, some algorithms have led to heuristics when the field of conception and of operations became too large for the original rules to retain their validity.

The reader should also note that this second part of the work contains an important recapitulation of basic Boolean properties that are so often needed. This was not given in the earlier parts devoted to the theory of operations research and will complete the mathematical equipment of some of our readers.

And, of course, as always in these theoretical parts, simple but suggestive examples of properties will precede or follow the methods that are suggested and the models that are constructed.

Section 10. **Mathematical Properties of Boole's Binary Algebra**

1. Introductory Remarks

The theory of sets, in its most elementary form, will certainly be familiar to readers of the second volume of this work, but we propose to recall certain of its properties as an introduction to Boole's binary algebra. Even if these concepts are well-known to every university or high school student studying one of those sciences in which mathematics play a major role, even if they are well-known to every young engineer, perhaps those of our readers who are not equally young will be glad to find a résumé of the more common properties of sets. It is, indeed, a principle of this series to provide recapitulations for the benefit of those who did not in their time have the opportunity of acquiring knowledge that has since become commonplace. This is, however, a very condensed résumé, and we refer our readers to works dealing more fully with the subject such as [K11], [K12], and [K14].

2. Characteristic Function of a Subset

Let us consider a subset $\mathbf{A}$ forming part of a referential $\mathbf{E}$[1] (Fig. 10.1). With each subset such that $\mathbf{A} \subset \mathbf{E}$ we associate a *characteristic function* of the form[2]

$$f_a(x) = F(\mathbf{A}; x), \tag{10.1}$$

[1] For example, the set of all human beings is a referential, of which all those of male sex form a subset.

[2] We ought strictly to use a notation that recalls the defined set, for example, $f_{\mathbf{A}}(x) = F(\mathbf{A}; x)$. To simplify the printing we shall henceforward use the notation $f_a(x)$, for which the theoretical justification will be given later on.

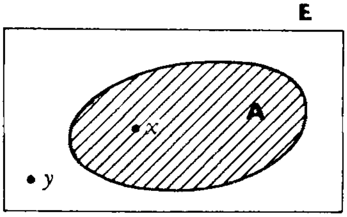


FIG. 10.1

such that

$$\text{(10.2)}\qquad \begin{array}{ll} \text{if } x \in \mathbf{A}: & f_a(x) = 1\,, \\ \text{if } x \notin \mathbf{A}, \text{ that is } x \in \overline{\mathbf{A}}: & f_a(x) = 0\,. \end{array}$$

Hence, such a characteristic function can only assume the values 0 or 1.

We shall now show that, to each operation of complementation ($^-$), union ($\cup$), and intersection ($\cap$) of the theory of sets that constitute a Boolean type algebra, there corresponds, respectively, an operation of complementation or negation ($^-$), Boolean sum ($\dot{+}$), and Boolean product ($\odot$) in Boole's binary algebra.

The Negative Case

It is sufficient to revert to the definition of the characteristic function to discover the relation between $F(\overline{\mathbf{A}}; x)$ and $F(\mathbf{A}; x)$.

If we assume

$$\text{(10.3)}\qquad F(\mathbf{A}; x) = f_a(x)\,,$$

either $f_a(x) = 1$ and x does not belong to the negative complementary of **A**, then:

$$\text{(10.4)}\qquad F(\overline{\mathbf{A}}; x) = 0$$

or $f_a(x) = 0$ and x belongs to **A**, then

$$\text{(10.5)}\qquad F(\mathbf{A}; x) = 1\,.$$

Figure 10.2 confirms that

$$\text{(10.6)}\qquad F(\overline{\mathbf{A}}; x) = 1 - F(\mathbf{A}; x)\,.$$

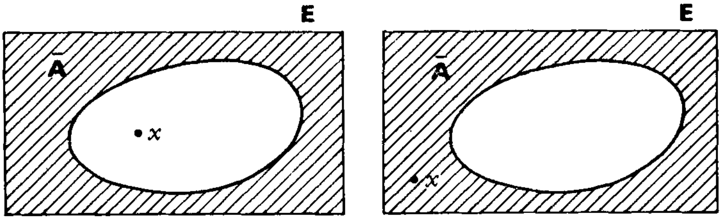
or

$$f_{\bar{a}}(x) = 1 - f_a(x)\,.$$

We can equally well write $\overline{f_a(x)}$ instead of $f_{\bar{a}}(x)$.

The Case of Intersection

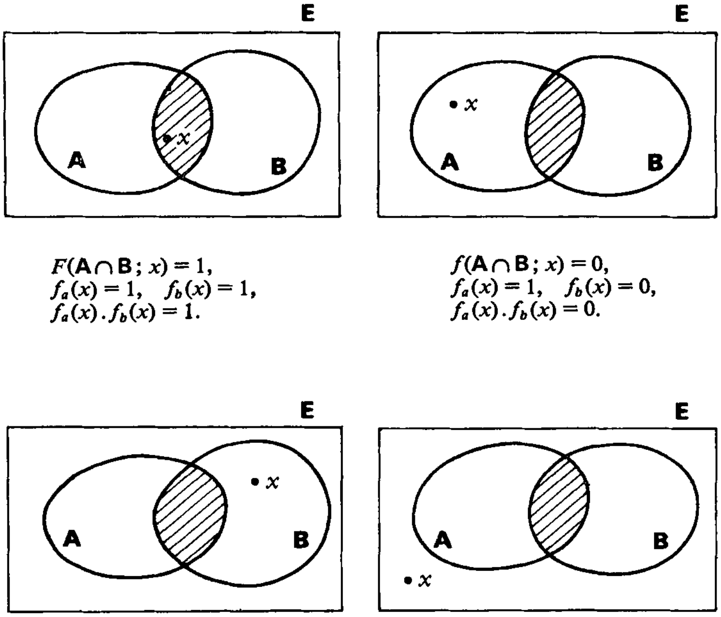
Let us consider two subsets $\mathbf{A} \subset \mathbf{E}$ and $\mathbf{B} \subset \mathbf{E}$ as well as the two character-



$$F(\bar{\mathbf{A}};x)=0,\quad f_a(x)=1,\quad F(\bar{\mathbf{A}};x)=1-f_a(x)=1-F(\mathbf{A};x).$$

$$F(\bar{\mathbf{A}};x)=1,\quad f_a(x)=0,\quad F(\bar{\mathbf{A}};x)=1-f_a(x)=1-F(\mathbf{A};x).$$

FIG. 10.2



$$F(\mathbf{A}\cap\mathbf{B};x)=1,\quad f_a(x)=1,\quad f_b(x)=1,\quad f_a(x).f_b(x)=1.$$

$$f(\mathbf{A}\cap\mathbf{B};x)=0,\quad f_a(x)=1,\quad f_b(x)=0,\quad f_a(x).f_b(x)=0.$$

$$F(\mathbf{A}\cap\mathbf{B};x)=0,\quad f_a(x)=0,\quad f_b(x)=1,\quad f_a(x).f_b(x)=0.$$

$$F(\mathbf{A}\cap\mathbf{B};x)=0,\quad f_a(x)=0,\quad f_b(x)=0,\quad f_a(x).f_b(x)=0.$$

FIG. 10.3

istic functions associated with them,

$$f_a(x) = F(\mathbf{A};x) \tag{10.7}$$

and

(10.8) $$f_b(x) = F(\mathbf{B}; x).$$

If $x \in \mathbf{A} \cap \mathbf{B}$ then we can state

(10.9) $$F(\mathbf{A} \cap \mathbf{B}; x) = 1.$$

If $x \notin \mathbf{A} \cap \mathbf{B}$ then we can state

(10.10) $$F(\mathbf{A} \cap \mathbf{B}; x) = 0.$$

From an examination of Fig. 10.3 we can easily see that, with the values of $f_a(x)$ and $f_b(x)$ known, we can determine the value of $F(\mathbf{A} \cap \mathbf{B}; x)$ by finding their product. As a result we can state

(10.11) $$F(\mathbf{A} \cap \mathbf{B}; x) = F(\mathbf{A}; x) . F(\mathbf{B}; x),$$

or again

(10.12) $$f_{a.b}(x) = f_a(x) . f_b(x).$$

The Case of Union

Having assumed

(10.13) $$f_a(x) = F(\mathbf{A}; x)$$

and

(10.14) $$f_b(x) = F(\mathbf{B}; x)$$

in the same manner as above, let us try to express $F(\mathbf{A} \cap \mathbf{B}; x)$ with the aid of Fig. 10.4.

We establish that the arithmetical sign + cannot be used to calculate the values of $F(\mathbf{A} \cup \mathbf{B}; x)$ commencing with $f_a(x)$ and $f_b(x)$. On the other hand we can state

(10.15) $$F(\mathbf{A} \cup \mathbf{B}) = f_a(x) \dotplus f_b(x)$$

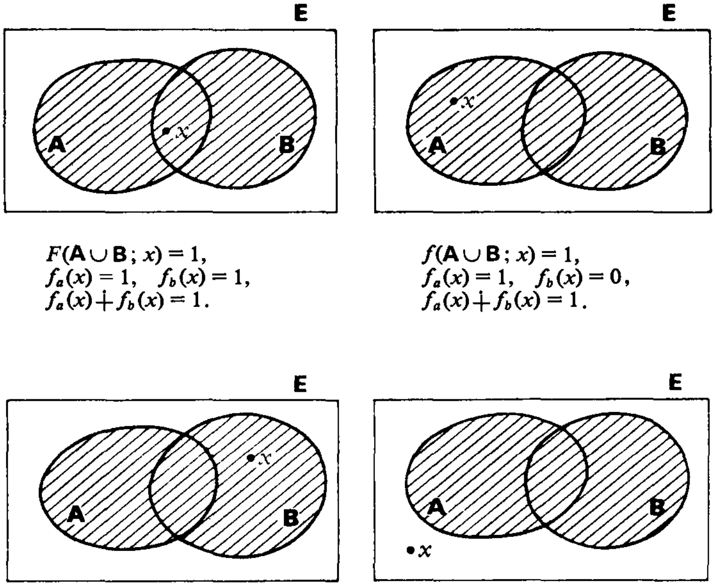
on the condition it is agreed that

$$1 \dotplus 1 = 1, \quad 1 \dotplus 0 = 1, \quad 0 \dotplus 1 = 1, \text{ and } \quad 0 \dotplus 0 = 0.$$

3. Canonical Forms

Let us consider the subsets $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n \subset \mathbf{E}$ and their characteristic functions that we express as x_i, 1, 2, ..., n; that is to say,

(10.16) $$f_{a_1}(x) = x_1, \quad f_{a_2}(x) = x_2, \ldots, \quad f_{a_n}(x) = x_n.$$



$F(\mathbf{A} \cup \mathbf{B}; x) = 1,$
$f_a(x) = 1, \quad f_b(x) = 1,$
$f_a(x) \dotplus f_b(x) = 1.$

$f(\mathbf{A} \cup \mathbf{B}; x) = 1,$
$f_a(x) = 1, \quad f_b(x) = 0,$
$f_a(x) \dotplus f_b(x) = 1.$

$F(\mathbf{A} \cup \mathbf{B}; x) = 1,$
$f_a(x) = 0, \quad f_b(x) = 1,$
$f_a(x) \dotplus f_b(x) = 1.$

$F(\mathbf{A} \cup \mathbf{B}; x) = 0,$
$f_a(x) = 0, \quad f_b(x) = 0,$
$f_a(x) \dotplus f_b(x) = 0.$

FIG. 10.4

Thus

$$\begin{aligned} x_i &= 1 \quad \text{if} \quad x \in \mathbf{A}_i \\ &= 0 \quad \text{if} \quad x \notin \mathbf{A}_i, \qquad i = 1, 2, \ldots, n. \end{aligned}$$

We can equally consider a Boolean function of the n subsets,

$$\text{(10.17)} \qquad \mathbf{\Phi}(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n),$$

and its characteristic function,

$$\text{(10.18)} \qquad \begin{aligned} f_{\Phi}(x) &= 1 \quad \text{if} \quad x \in \mathbf{\Phi}(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n) \\ &= 0 \quad \text{if} \quad x \notin \mathbf{\Phi}(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n). \end{aligned}$$

We shall associate with the function $\mathbf{\Phi}(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n)$ a function with binary values $\varphi(x_1, x_2, \ldots, x_n)$ depending on the binary variables $x_1, x_2, \ldots, x_n$. To effect this we need only replace the subsets $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n$ in $\mathbf{\Phi}$ by the variables $x_1, x_2, \ldots, x_n$, which will enter into φ and the operations $(\cup)$, $(\cap)$, $(^-)$ by the operations $(.)$, $(\dotplus)$, $(^-)$.

For instance, to

(10.19) $$\mathbf{\Phi}(\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3) = (\mathbf{A}_1 \cup \mathbf{A}_2) \cup \bar{\mathbf{A}}_3)$$

will correspond

$$\varphi(x_1, x_2, x_3) = x_1 . x_2 \dotplus \bar{x}_3 .$$

We can easily see that

(10.20) $$\varphi(x_1, x_2, \ldots, x_n) = f_{\Phi\,(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n)}(x) ,$$

since, by the inherent structure of $\varphi(x_1, x_2, \ldots, x_n)$, we shall have

(10.21) $$\varphi(x_1, x_2, \ldots, x_n) = 1 \quad \text{if} \quad x \in \mathbf{\Phi}(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_n) .$$

Let us now take the function

(10.22) $$y = \varphi(x_1, x_2, \ldots, x_n) ;$$

and let us assume

(10.23) $$y = x_1 . r \dotplus \bar{x}_1 . s ,$$

where r and s are the Boolean functions to be determined.

If

(10.24) $$x_1 = 1, \quad \bar{x}_1 = 0, \quad \text{then} \quad y = \varphi(1, x_2, \ldots, x_n) = r ,$$

and if

(10.25) $$x_1 = 0, \quad \bar{x}_1 = 1, \quad \text{then} \quad y = \varphi(0, x_2, \ldots, x_n) = s.$$

Hence we can state

(10.26) $$y = x_1 . \varphi(1, x_2, \ldots, x_n) \dotplus \bar{x}_1 . \varphi(0, x_2, \ldots, x_n) .$$

Proceeding in the same manner for the other variables, $x_2, \ldots, x_n$, we see that

(10.27)

$$\varphi(1, x_2, x_3, \ldots, x_n) = x_2 . \varphi(1, 1, x_3, \ldots, x_n) \dotplus \bar{x}_2 . \varphi(1, 0, x_3, \ldots, x_n) .$$

and

(10.28)

$$\varphi(0, x_2, x_3, \ldots, x_n) = x_2 . \varphi(0, 1, x_3, \ldots, x_n) \dotplus \bar{x}_2 . \varphi(0, 0, x_3, \ldots, x_n) .$$

Whence

(10.29) $$y = x_1 . x_2 . \varphi(1, 1, x_3, \ldots, x_n) \dotplus x_1 . \bar{x}_2 . \varphi(1, 0, x_3, \ldots, x_n) \\ \dotplus \bar{x}_1 . x_2 . \varphi(0, 1, x_3, \ldots, x_n) \dotplus \bar{x}_1 . \bar{x}_2 . \varphi(0, 0, x_3, \ldots, x_n) .$$

If we continue the same procedure for $x_2, \dots, x_n$, we finally obtain

$$\begin{aligned}
y &= \varphi(x_1, x_2, x_3, \dots, x_{n-1}, x_n) \\
&= x_1.x_2.x_3.\dots.x_{n-1}.x_n.\varphi(1, 1, 1, \dots, 1, 1) \\
&\quad + x_1.x_2.x_3.\dots.x_{n-1}.\bar{x}_n.\varphi(1, 1, 1, \dots, 1, 0) \\
&\quad + x_1.x_2.x_3.\dots.\bar{x}_{n-1}.x_n.\varphi(1, 1, 1, \dots, 0, 1) \\
&\quad + x_1.x_2.x_3.\dots.\bar{x}_{n-1}.\bar{x}_n.\varphi(1, 1, 1, \dots, 0, 0) \\
&\quad + \dots \\
&\quad + \bar{x}_1.\bar{x}_2.\bar{x}_3.\dots.x_{n-1}.x_n.\varphi(0, 0, 0, \dots, 1, 1) \\
&\quad + \bar{x}_1.\bar{x}_2.\bar{x}_3.\dots.x_{n-1}.\bar{x}_n.\varphi(0, 0, 0, \dots, 1, 0) \\
&\quad + \bar{x}_1.\bar{x}_2.\bar{x}_3.\dots.\bar{x}_{n-1}.x_n.\varphi(0, 0, 0, \dots, 0, 1) \\
&\quad + \bar{x}_1.\bar{x}_2.\bar{x}_3.\dots.\bar{x}_{n-1}.\bar{x}_n.\varphi(0, 0, 0, \dots, 0, 0).
\end{aligned} \tag{10.30}$$

Hence, assuming[1]

$$\begin{aligned}
\varphi_0 &= \varphi(0, 0, 0, \dots, 0, 0), \\
\varphi_1 &= \varphi(0, 0, 0, \dots, 0, 1), \\
&\dots\dots\dots\dots\dots\dots \\
\varphi_{2^n-2} &= \varphi(1, 1, 1, \dots, 1, 0), \\
\varphi_{2^n-1} &= \varphi(1, 1, 1, \dots, 1, 1).
\end{aligned} \tag{10.31}$$

and by naming as *minterms* the products represented in the following manner:

$$\begin{aligned}
m_0 &= \bar{x}_1.\bar{x}_2.\bar{x}_3.\dots.\bar{x}_{n-1}.\bar{x}_n, \\
m_1 &= \bar{x}_1.\bar{x}_2.\bar{x}_3.\dots.\bar{x}_{n-1}.x_n, \\
m_2 &= \bar{x}_1.\bar{x}_2.\bar{x}_3.\dots.x_{n-1}.\bar{x}_n, \\
&\dots\dots\dots\dots\dots\dots \\
m_{2^n-2} &= x_1.x_2.x_3.\dots.x_{n-1}.\bar{x}_n, \\
m_{2^n-1} &= x_1.x_2.x_3.\dots.x_{n-1}.x_n;
\end{aligned} \tag{10.32}$$

[1] It will be observed that for index we have used the decimal system of notation for the number in the binary system corresponding to the succession of binary figures within the parentheses.

we can express (10.30) as follows:

$$\begin{aligned} y &= \varphi(x_1, x_2, \ldots, x_n) \\ &= m_0 . \varphi_0 + m_1 . \varphi_1 + \ldots + m_{2^n-2} . \varphi_{2^n-2} + m_{2^n-1} . \varphi_{2^n-1} . \end{aligned} \tag{10.33}$$

Form (10.33) is termed the *canonical disjunctive form* or *first canonical form*. Let us now assume

$$\begin{aligned} y &= \mu(x_1, x_2, \ldots, x_n) \\ &= (x_1 + r') . (\bar{x}_1 + s'), \end{aligned} \tag{10.34}$$

where r' and s' are the Boolean functions to be determined.

Using the same procedure as before and taking $x_1 = 1$ and $x_1 = 0$, then $x_2 = 1$ and $x_2 = 0$, and so on, we now obtain

$$\begin{aligned} y &= \mu(x_1, x_2, x_3, \ldots, x_{n-1}, x_n) \\ &= [x_1 + x_2 + x_3 + \ldots + x_{n-1} + x_n + \mu(0, 0, 0, \ldots, 0, 0)] \\ &\quad . [x_1 + x_2 + x_3 + \ldots + x_{n-1} + \bar{x}_n + \mu(0, 0, 0, \ldots, 0, 1)] \\ &\quad . [x_1 + x_2 + x_3 + \ldots + \bar{x}_{n-1} + x_n + \mu(0, 0, 0, \ldots, 1, 0)] \\ &\quad . [x_1 + x_2 + x_3 + \ldots + \bar{x}_{n-1} + \bar{x}_n + \mu(0, 0, 0, ... 1, 1)] \\ &\quad . \ldots \\ &\quad . [\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \ldots + x_{n-1} + x_n + \mu(1, 1, 1, \ldots, 0, 0)] \\ &\quad . [\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \ldots + x_{n-1} + \bar{x}_n + \mu(1, 1, 1, \ldots, 0, 1)] \\ &\quad . [\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \ldots + \bar{x}_{n-1} + x_n + \mu(1, 1, 1, \ldots, 1, 0)] \\ &\quad . [\bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \ldots + \bar{x}_{n-1} + \bar{x}_n + \mu(1, 1, 1, \ldots, 1, 1)] . \end{aligned} \tag{10.35}$$

The following Boolean sums are called *maxterms*:

$$\begin{aligned} M_0 &= \bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \ldots + \bar{x}_{n-1} + \bar{x}_n, \\ M_1 &= \bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \ldots + \bar{x}_{n-1} + x_n, \\ M_2 &= \bar{x}_1 + \bar{x}_2 + \bar{x}_3 + \ldots + x_{n-1} + \bar{x}_n, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ M_{2^n-2} &= x_1 + x_2 + x_3 + \ldots + x_{n-1} + \bar{x}_n, \\ M_{2^n-1} &= x_1 + x_2 + x_3 + \ldots + x_{n-1} + x_n . \end{aligned} \tag{10.36}$$

We can then express (10.35) as

$$(10.37)\qquad \begin{aligned} y &= \mu(x_1, x_2, \cdots x_n) \\ &= (M_{2^n-1}+\mu_0).(M_{2^n-2}+\mu_1)\dots(M_1+\mu_{2^n-2}).(M_0+\mu_{2^n-1}). \end{aligned}$$

This structure is called the *canonical conjunctive form* or *second canonical form.*

It can be shown that expansions (10.30) and (10.35) are unique (see, for example, [K12]).

Example

The detailed calculation is left to the reader.

Let

$$(10.38)\qquad y = a.(\overline{\bar{b}.\bar{d}+b.c})+\bar{a}.\bar{b}.(c+d).$$

We find successively that

(10.39)

$$\varphi_0=0,\ \varphi_1=1,\ \varphi_2=1,\ \varphi_3=1,\ \varphi_4=0,\ \varphi_5=0,\ \varphi_6=0,\ \varphi_7=0,$$
$$\varphi_8=0,\ \varphi_9=1,\ \varphi_{10}=0,\ \varphi_{11}=1,\ \varphi_{12}=1,\ \varphi_{13}=1,\ \varphi_{14}=0,\ \varphi_{15}=0.$$

Let us merely give the minterms with nonnull coefficients

$$(10.40)\qquad \begin{aligned} &m_1 = \bar{a}.\bar{b}.\bar{c}.d, && m_2 = \bar{a}.\bar{b}.c.\bar{d}, && m_3 = \bar{a}.\bar{b}.c.d, \\ &m_9 = a.\bar{b}.\bar{c}.d, && m_{11} = a.\bar{b}.c.d, && m_{12} = a.b.\bar{c}.\bar{d}, \\ &m_{13} = a.b.\bar{c}.d, \end{aligned}$$

and

$$(10.41)\qquad y = m_1+m_2+m_3+m_9+m_{11}+m_{12}+m_{13}.$$

Let us now consider the expansion in maxterms (the calculation of which is left to the reader), giving only the maxterms preceded by a null term:

(10.42)

$$\begin{aligned} &M_0 = \bar{a}+\bar{b}+\bar{c}+\bar{d}, && M_1 = \bar{a}+\bar{b}+\bar{c}+d, && M_5 = \bar{a}+b+\bar{c}+d, \\ &M_7 = \bar{a}+b+c+d, && M_8 = a+\bar{b}+\bar{c}+\bar{d}, && M_9 = a+\bar{b}+\bar{c}+d, \\ &M_{10} = a+\bar{b}+c+\bar{d}, && M_{11} = a+\bar{b}+c+d, && M_{15} = a+b+c+d. \end{aligned}$$

We pass easily from a decomposition in minterms to one in maxterms: (a) write down the junction of all the minterms not included in y; (b) replace this junction by the intersection of the maxterms corresponding to the minterms in such a way that to the index i of a minterm there corresponds the index 2^n-1-i of the maxterm obtained.

The passage from a decomposition in maxterms to one in minterms proceeds in the same manner, but the union is exchanged for the intersection and the term minterm for that of maxterm.

Identity of Two Boolean Functions

An easy method of finding whether two Boolean functions are identical is to draw up the table of values of each and to compare them. But we can also employ the canonical forms, since two Boolean functions are identical if they possess the same cononical form (first or second form).

Section 11. **Lattice Theory**

1. Observations

In this work we are concerned only with finished lattices, that is to say, those possessing a finite number of elements or half-lattices that are denumerable, although the principal properties can be extended to nondenumerable lattices.

The structure of the lattice that can also, for those that are denumerable, be shown in the form of a graph is one of the most important in the whole field of modern mathematics, whether pure or applied,[1] and we shall examine it in considerable detail.

2. A Reminder of the Concept of an Ordered Set

In Volume 2 we dealt very briefly with the concept of an ordered set, since it was only occasionally needed for our explanations. A fuller treatment of this concept is now required for the study of lattices. Before providing this, however, it is advisable to recall the nature of the properties of reflexivity, transitivity, symmetry, and asymmetry.

Reflexivity

A binary relation defined by a graph $\mathbf{G} \subset \mathbf{E} \times \mathbf{E}$ is *reflexive* if all the pairs (x, x) belong to the graph.[2] In the terminology of modern mathematics this is expressed

$$(11.1) \qquad \forall x \in \mathbf{E}: \qquad (x, x) \in \mathbf{G} \subset \mathbf{E} \times \mathbf{E}.$$

In Figs. 11.1 and 11.2 we have shown an identical relation. Figure 11.1, in

[1] To demonstrate our explanations we shall employ a set with five elements, but these explanations apply equally to any finite set (or even to an infinite set, though we are not generally concerned with such here).

[2] In Volume 2 a graph was defined by the pair $(\mathbf{E}, \Gamma)$ or the pair $(\mathbf{E}, \mathbf{U})$ (see pp. 229–230). A graph can also be defined by set $\mathbf{U}$ or $\mathbf{G}$ of the arcs, with the reservation of referring to the referential $\mathbf{E}$, as $\mathbf{G} \subset \mathbf{E} \times \mathbf{E}$.

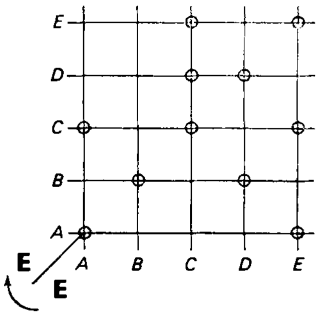


FIG. 11.1

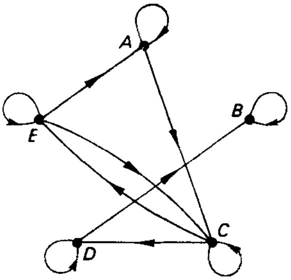


FIG. 11.2

the form of a grid, represents the binary relation, while Fig. 11.2 gives an arrowed representation of it, the two diagrams being shown together for instructional purposes. It is easy to verify that this binary relation is reflexive.

Transitivity

A binary relation defined by a graph $\mathbf{G} \subset \mathbf{E} \times \mathbf{E}$ is *transitive* if, with two pairs (x, y) and (y, z) belonging to the graph, (x, z) also belongs to it.

$$(11.2) \qquad (x, y) \in \mathbf{G} \text{ and } (y, z) \in \mathbf{G} \Rightarrow (x, z) \in \mathbf{G}.$$

The binary relation shown in Figs. 11.3 and 11.4 is transitive; if we verify a few pairs the reader can check the remainder.

$$(11.3) \qquad \begin{array}{lll} (E, A) \in \mathbf{G},\ (A, B) \in \mathbf{G}\,; & \text{we find} & (E, B) \in \mathbf{G}. \\ (A, B) \in \mathbf{G},\ (B, A) \in \mathbf{G}\,; & \text{we find} & (A, A) \in \mathbf{G}. \\ (C, B) \in \mathbf{G},\ (B, A) \in \mathbf{G}\,; & \text{we find} & (C, A) \in \mathbf{G}. \end{array}$$

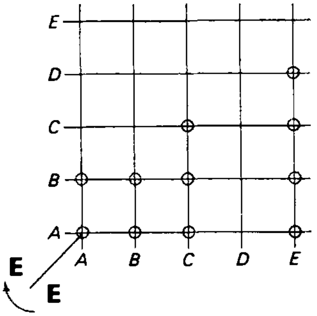


FIG. 11.3

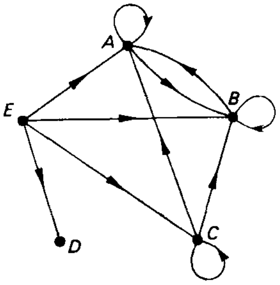


FIG. 11.4

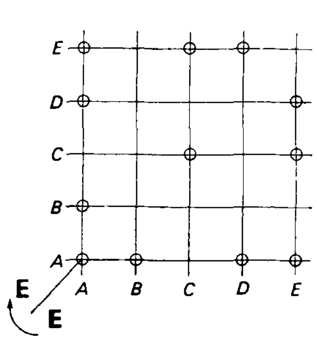


FIG. 11.5

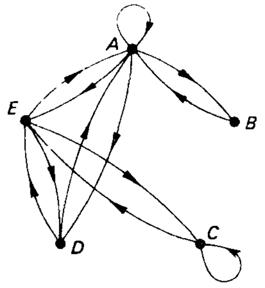


FIG. 11.6

Symmetry

A binary relation[1] is symmetrical if, with a pair (x, y) belonging to the graph, the pair (y, x) also belong to it. This is expressed as

$$(11.4) \qquad (x, y) \in \mathbf{G} \Rightarrow (y, x) \in \mathbf{G}.$$

Or, better still as

$$(11.5) \qquad (x, y) \in \mathbf{G} \Leftrightarrow (y, x) \in \mathbf{G}$$

since the implication applies as much to (y, x) as to (x, y).

The binary relation shown in Figs. 11.5 and 11.6 is symmetrical.

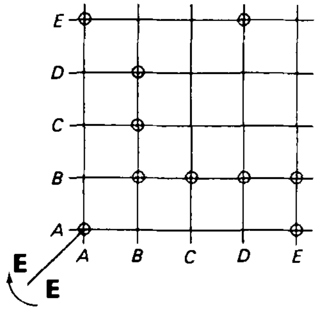


FIG. 11.7

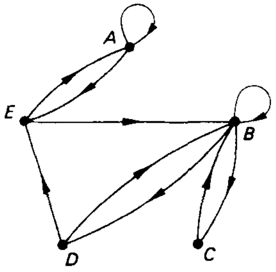


FIG. 11.8

Nonsymmetry

For at least one pair this property exists if there is no symmetry and is

[1] We shall no longer repeat "defined by a graph $\mathbf{G} \subset \mathbf{E} \times \mathbf{E}$."

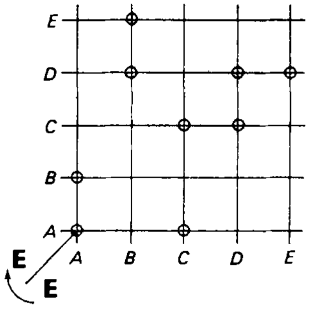


FIG. 11.9

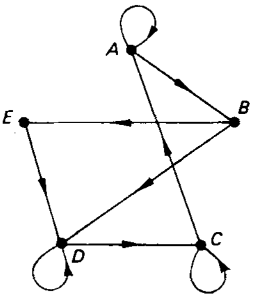


FIG. 11.10

expressed as

(11.6) $\exists((x, y) \in \mathbf{G} \text{ and } (y, x) \notin \mathbf{G})$.

The binary relation shown in Figs. 11.7 and 11.8 is asymmetrical.

Asymmetry in the Broad Sense

If a pair (x, y) where $y \neq x$ belong to the graph, then (y, x) do not belong to it. This does not apply to pairs in which $y = x$. This property is expressed as

(11.7) $(x, y) \in \mathbf{G} \text{ and } y \neq x \Rightarrow (y, x) \notin \mathbf{G}$.

Or, better still as

(11.8) $((x, y) \in \mathbf{G} \text{ and } (y, x) \in \mathbf{G}) \Rightarrow (x = y)$.

The binary relation shown in Figs. 11.9 and 11.10 is asymmetrical in the broad sense.

Asymmetry in the Strict Sense

If a pair (x, y) belong to the graph, then (y, x) do not belong to it; this implies that a pair (x, x) also do not belong. This is written as

(11.9) $(x, y) \in \mathbf{G} \Rightarrow (y, x) \notin \mathbf{G}$.

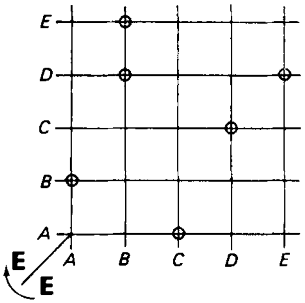


FIG. 11.11

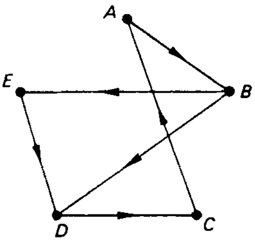


FIG. 11.12

The binary relation shown in Figs. 11.11 and 11.12 is asymmetrical in the strict sense. This example corresponds to that of Figs. 11.9 and 11.10 in which the pairs (A, A), (C, C), and (D, D) have been suppressed.

Relation of Preorder or Weak Order

A binary relation that is transitive and reflexive is a *relation of preorder* or *of weak order.*

The binary relation shown in Figs. 11.13 and 11.14 represents one of preorder.

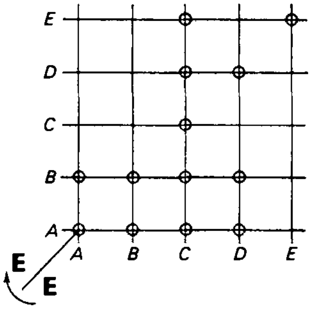


FIG. 11.13

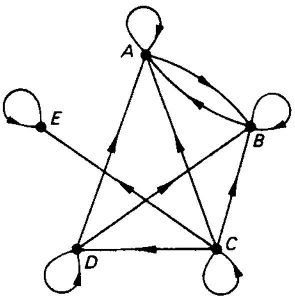


FIG. 11.14

Relation of Equivalence

A relation of preorder that is symmetrical is called a *relation of equivalence* and an example of this is shown in Figs. 11.15 and 11.16.

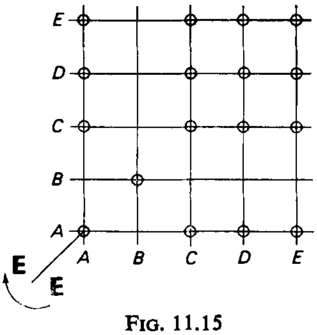


FIG. 11.15

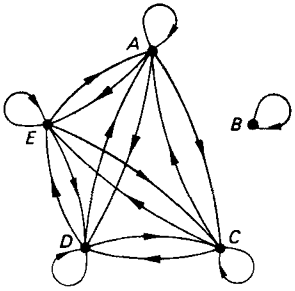


FIG. 11.16

Relation of Nonstrict Order

A relation of preorder that is asymmetric in the broad sense is called a *relation of nonstrict order*, and an example of this is given in Figs. 11.17 and 11.18.

Relation of Strict Order

This implies a transitive and asymmetrical relation in the strict sense. To every relation of a strict order there is one and only one corresponding relation of a nonstrict order to which the property of reflexivity has been added. The

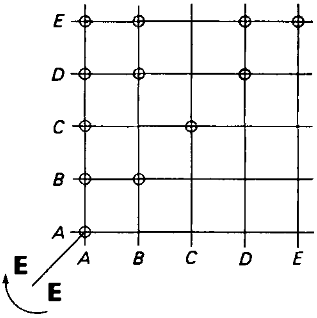


FIG. 11.17

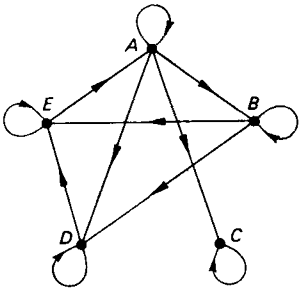


FIG. 11.18

example given in Figs. 11.19 and 11.20 should be compared with that of Figs. 11.17 and 11.18.

Unless specified differently we shall always consider relations of a nonstrict order in what follows.

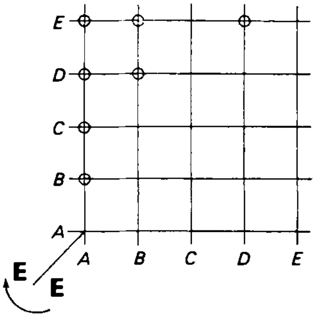


FIG. 11.19

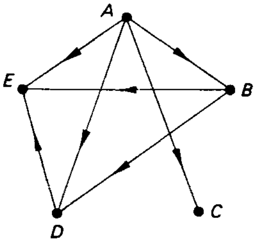


FIG. 11.20

Relation of Total Order

When, for each pair $(x, y) \in \mathbf{E} \times \mathbf{E}$ we have $(x, y) \in \mathbf{G}$ and|or $(y, x) \in \mathbf{G}$ and the binary relation is one of order, we say that the order is total.

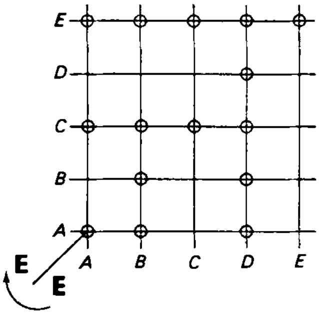


FIG. 11.21

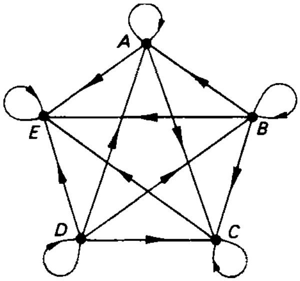


FIG. 11.22

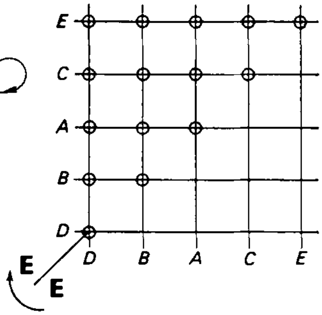


FIG. 11.23

This is the case in the example shown in Figs. 11.21 and 11.22; in Fig. 11.23 we have permutated the elements of **E** in Fig. 11.21 in order to demonstrate an important and fundamental property.

If the pair $(x, y) \in \mathbf{G}$ and if we are concerned with a relation of order, we can write

$$(11.10) \qquad x \preccurlyeq y .$$

We then say that the elements x and y are comparable. In a relation of total order all the elements are comparable (see Fig. 11.23). This is the important property referred to above.

Relation of Partial Order

An ordered set that is not totally ordered is said to be *partially ordered.* Stated differently, this is a set in which all the elements are not comparable.

Figures 11.24–11.26 on the one hand and Figs. 11.27–11.29 on the other hand provide two examples of partial order.

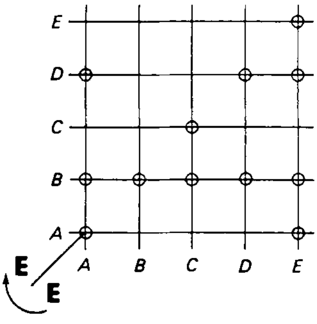


FIG. 11.24

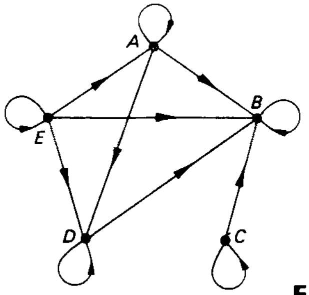


FIG. 11.25

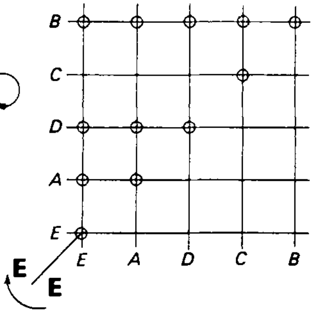


FIG. 11.26

Let us now consider the three figures, 11.22, 11.25, and 11.28.

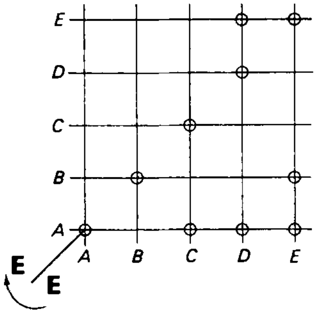


FIG. 11.27

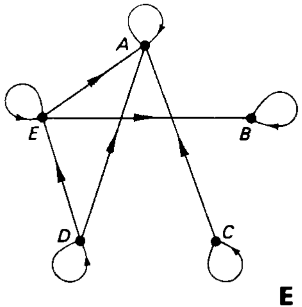


FIG. 11.28

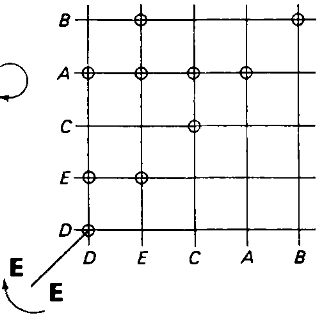


FIG. 11.29

For Fig. 11.22 we can state,

$$D \preccurlyeq B \preccurlyeq A \preccurlyeq C \preccurlyeq E. \tag{11.11}$$

Since all the elements are comparable with each other we can place them in a unique sequence that characterizes the total order.

For Fig. 11.25 we can say,

$$E \preccurlyeq A \preccurlyeq D \preccurlyeq B, \qquad C \preccurlyeq B. \tag{11.12}$$

The order is partial since we cannot obtain a unique sequence. The same applies to Fig. 11.28:

$$D \preccurlyeq E \preccurlyeq A, \qquad E \preccurlyeq B, \qquad C \preccurlyeq A. \tag{11.13}$$

Minimal and Maximal Element

An element a of an ordered set $\mathbf{E}$ is called the *minimal element* if no element x other than a is such that $x \preccurlyeq a$. In other words,

$$(x \preccurlyeq a) \Rightarrow (x = a). \tag{11.14}$$

In a similar manner, an element b of an ordered set $\mathbf{E}$ is called the *maximal element* if no element other than b is such that $x \preccurlyeq b$. In other words,

$$(x \succcurlyeq b) \Rightarrow (x = b). \tag{11.15}$$

In Fig. 11.30 the reader can verify whether there are four maximal and two minimal elements.

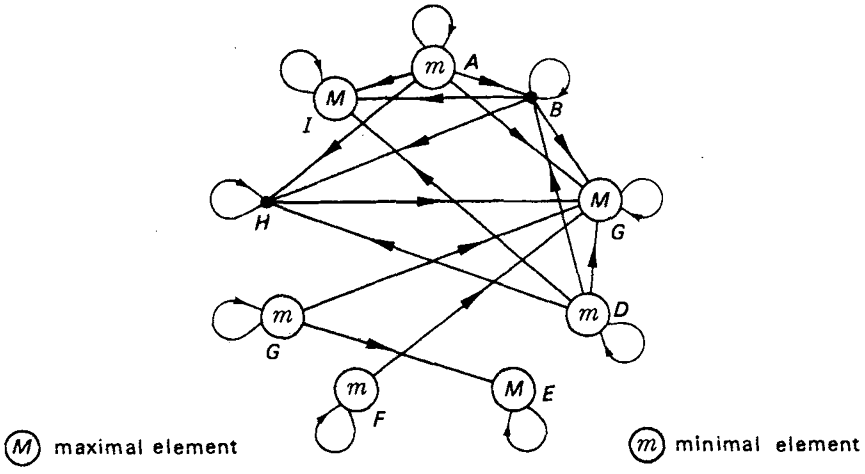


FIG. 11.30

We have

$$\begin{aligned} &A \preccurlyeq B \preccurlyeq H \preccurlyeq C, \qquad A \preccurlyeq B \preccurlyeq I, \\ &D \preccurlyeq B \preccurlyeq H \preccurlyeq C, \qquad D \preccurlyeq B \preccurlyeq I, \\ &F \preccurlyeq C, \qquad G \preccurlyeq C, \qquad G \preccurlyeq E. \end{aligned} \tag{11.16}$$

Smallest and Largest Element

Given a set $\mathbf{E}$ ordered by a relation of order, we say that an element $a \in \mathbf{E}$ is the *smallest element* of $\mathbf{E}$ if, for every $x \in \mathbf{E}$, we have

$$a \preccurlyeq x. \tag{11.17}$$

We say that $b \in \mathbf{E}$ is the *largest element* of $\mathbf{E}$ if, for every $b \in \mathbf{E}$, we have

$$b \succcurlyeq x. \tag{11.18}$$

Let us consider the examples shown in Figs. 11.31 to 11.33.

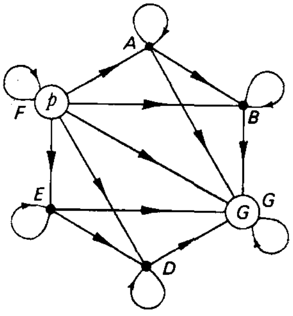


FIG. 11.31

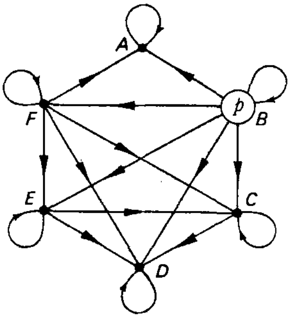


FIG. 11.32

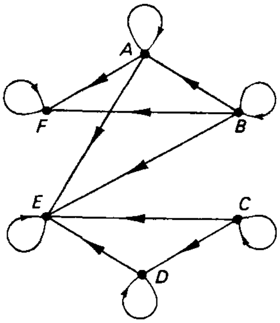


FIG. 11.33

In Fig. 11.31 we see that

$$\begin{aligned} &F \preccurlyeq A \preccurlyeq B \preccurlyeq C, \\ &F \preccurlyeq E \preccurlyeq D \preccurlyeq C. \end{aligned} \tag{11.19}$$

Hence this set contains a smallest element F and a largest element C.

In Fig. 11.32 we see that

$$\begin{aligned} &B \preccurlyeq F \preccurlyeq A, \\ &B \preccurlyeq F \preccurlyeq E \preccurlyeq C \preccurlyeq D. \end{aligned} \tag{11.20}$$

Hence this set contains a smallest element B but does not include a largest element.

In Fig. 11.33 we see that

$$(11.21)\qquad \begin{aligned} &B \preccurlyeq A \preccurlyeq F, \\ &B \preccurlyeq A \preccurlyeq E, \\ &C \preccurlyeq D \preccurlyeq E. \end{aligned}$$

This set possesses neither a smallest nor a largest element. The same applies to the example in Fig. 11.30.

Observation

It is unnecessary to prove that when a smallest element (and also a largest element) exists it is unique.

Minorant and Majorant

If $\mathbf{E}$ is an ordered set with $\mathbf{E}' \subset \mathbf{E}$, we apply the term *minorant* of $\mathbf{E}'$ to every element $a \in \mathbf{E}$ such that, for every element $b \in \mathbf{E}'$, we have $a \preccurlyeq b$. We also say that a minors $\mathbf{E}'$. In a similar manner we use the term *majorant* of $\mathbf{E}'$ for every element $a \in \mathbf{E}$ such that, for every element $b \in \mathbf{E}'$, we have $a \succcurlyeq b$.

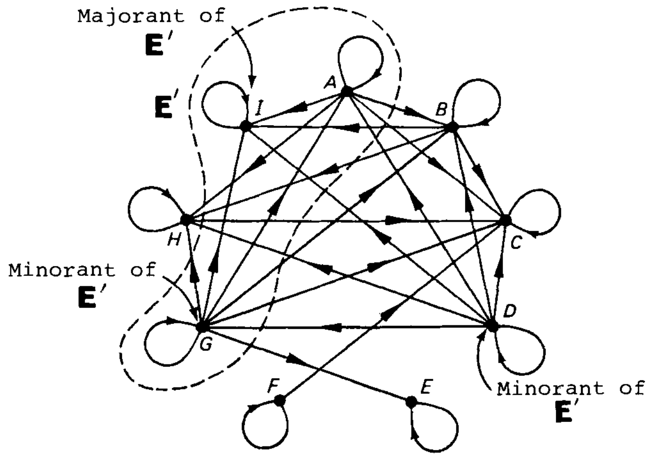


FIG. 11.34

Let us take as an example the graph of Fig. 11.34, and let

$$(11.22)\qquad \mathbf{E} = \{A, B, C, D, E, F, G, H, I\}$$

and

$$(11.23)\qquad \mathbf{E}' = \{A, G, I\}.$$

It can be seen that D is a minorant of $\mathbf{E}'$ for $D \preccurlyeq A$, $D \preccurlyeq I$, $D \preccurlyeq G$. G is a minorant of $\mathbf{E}'$ since $G \preccurlyeq G$, $G \preccurlyeq I$, $G \preccurlyeq A$. I is a majorant of $\mathbf{E}'$ since $I \succcurlyeq I$, $I \succcurlyeq G$, $I \succcurlyeq A$. The reader is left to discover other possible majorants and minorants.

Lower Bound and Upper Bound

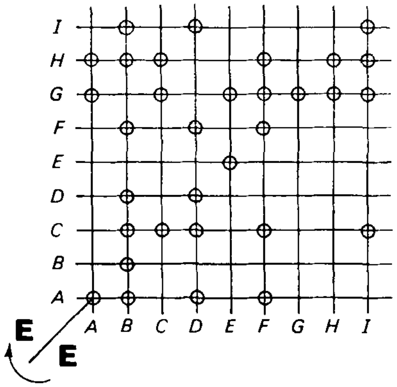
Let $\mathbf{F} \subset \mathbf{E}$ where E is an ordered set. Let us suppose that $\mathbf{F}$ is majored by certain elements of $\mathbf{E}$ and let $\mathbf{M}$ be the set of these majorants. If $\mathbf{M}$ includes a smallest element m, then m is called the *upper bound* of $\mathbf{F}$.

Similarly, let us suppose $\mathbf{F}$ is minored by certain elements of $\mathbf{E}$ and let $\mathbf{N}$ be the set of these minorants. If $\mathbf{N}$ includes a largest element n, then n is called the *lower bound* of $\mathbf{F}$.

The upper bound will be referred to as

$$\sup \mathbf{F} \quad \text{or} \quad \sup_{\mathbf{E}} \mathbf{F}$$

if there is no chance of confusion.



	A	B	C	D	E	F	G	H	I
I		⊕		⊕					⊕
H	⊕	⊕	⊕			⊕		⊕	⊕
G	⊕		⊕		⊕	⊕	⊕	⊕	⊕
F		⊕		⊕		⊕			
E					⊕				
D		⊕		⊕					
C		⊕	⊕	⊕		⊕			⊕
B		⊕							
A	⊕	⊕		⊕		⊕			

FIG. 11.35

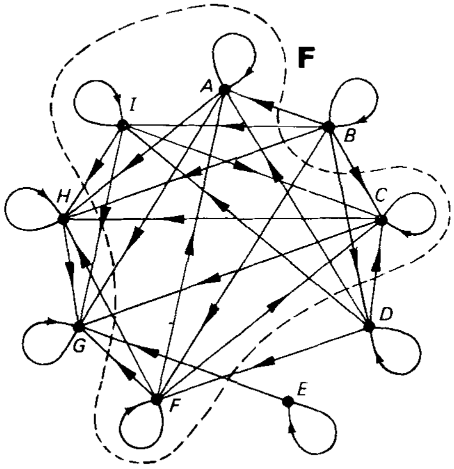
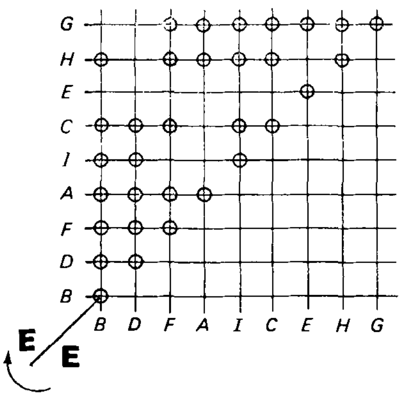


FIG. 11.36



	B	D	F	A	I	C	E	H	G
G			⊕	⊕	⊕	⊕	⊕	⊕	⊕
H	⊕		⊕	⊕	⊕	⊕		⊕	
E							⊕		
C	⊕	⊕	⊕		⊕	⊕			
I	⊕	⊕			⊕				
A	⊕	⊕	⊕	⊕					
F	⊕	⊕	⊕						
D	⊕	⊕							
B	⊕								

FIG. 11.37

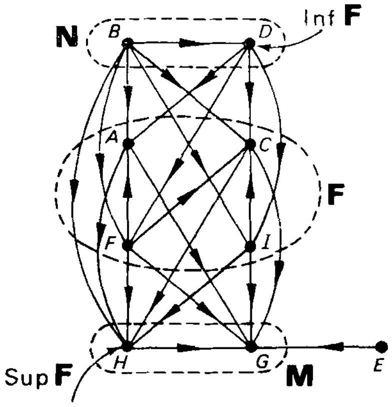


FIG. 11.38.
Note. Pairs (x, x) have not been shown.

Similarly the lower bound will be referred to as

$$\inf_E \mathsf{F}$$

or

$$\inf \mathsf{F}.$$

It is clear from the definition that when a lower (or upper) bound exists it must be unique.

Our first example is shown in Figs. 11.35 and 11.36. To make sets M and N more easily distinguishable these figures have been redrawn and modified (Figs. 11.37 and 11.38). We see that

(11.24) $\mathsf{M} = \{G, H\}$ and $\mathsf{N} = \{B, D\}$.

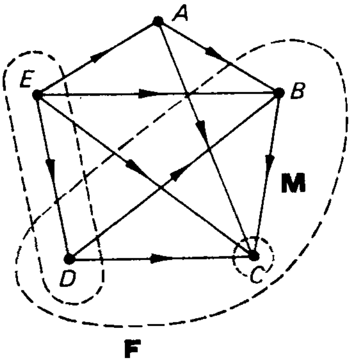


FIG. 11.39

Since D is the largest element of N it is the lower bound of F, and since H is the smallest element of M it is the upper bound of F.

In this example $\mathsf{F} \cap \mathsf{M} = \varnothing$ and $\mathsf{F} \cap \mathsf{N} = \varnothing$, but such is not the case in the following example.

Returning to Fig. 11.39, we have

(11.25) $\mathsf{M} = \{C\}$ and $\mathsf{N} = \{D, E\}$.

We see that

(11.26) $\sup \mathsf{F} = C \in \mathsf{F}$ and $\inf \mathsf{F} = D \in \mathsf{F}$.

Chain. Maximal Chain

Every totally ordered subset of an ordered set is called a *chain.*[1]

Let us take as an example the ordered set shown in Fig. 11.40.

[1] This concept has no resemblance to that of the same name that we defined in the theory of graphs (sequence of links), Volume 2, pp. 9 and 242. It is unfortunate that the same term should have been chosen for such different concepts, but so many names are required in mathematics.

$\{D, C, F, G\}$ is a chain since $D \preccurlyeq C \preccurlyeq G \preccurlyeq F$. $\{E, F\}$ is a chain since $E \preccurlyeq F$. $\{A, B\}$ is a chain since $A \preccurlyeq B$.

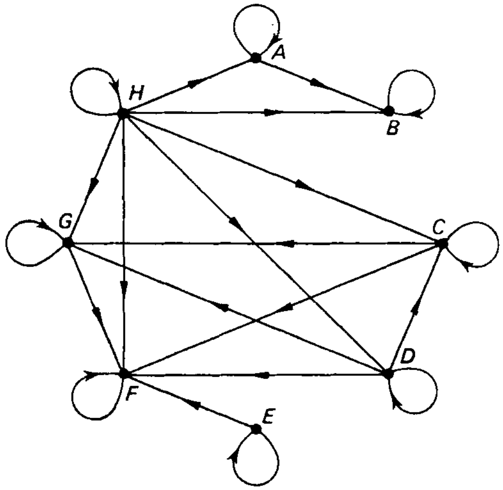


FIG. 11.40

A chain is said to be *maximal* if it is not a subset of a chain, apart from itself. Hence, still considering Fig. 11.40, $\{C, D, F, G, H\}$, $\{E, F\}$, $\{A, B, H\}$ are maximal chains. We have

$$H \preccurlyeq D \preccurlyeq C \preccurlyeq G \preccurlyeq F, \qquad E \preccurlyeq F \quad \text{and} \quad H \preccurlyeq A \preccurlyeq B.$$

Sup Half-Lattice

Let us consider an ordered set. If each pair of elements in this set posses an upper bound, we say that this ordered set is a *sup half-lattice.*

This is true of the ordered set shown in Fig. 11.41. We can verify that each pair of elements has an upper bound: $\sup\{A, B\} = F$, $\sup\{A, C\} = A$, $\sup\{A, D\} = F$, and so forth.

The maximal chains of this ordered set are $\{A, C, F\}$ and $\{B, D, E, F\}$, their upper bound being F.

We can verify that Fig. 11.42 does not represent a sup *half-lattice* but that Fig. 11.43 does represent one.

Inf Half-Lattice

If every pair of an ordered set possesses a lower bound, we say that this ordered set is an *inf half-lattice.*

Figure 11.41 does not represent such a half-lattice, but Figs. 11.42 and 11.43 do so.

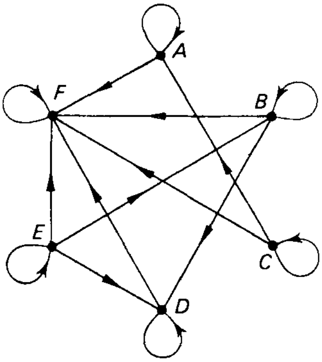


FIG. 11.41

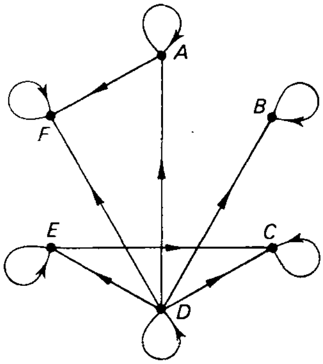


FIG. 11.42

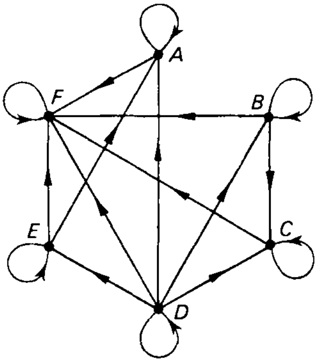


FIG. 11.43

3. Lattice[1]

An ordered set that is both a sup half-lattice and an inf half-lattice is a lattice.

Stated differently, we say that a lattice is an ordered set in which every pair of elements possesses an upper and a lower bound. A lattice may also be called a *trellis*, a *reticulated set* or even an *ordered network*.

Figures 11.41 and 11.42 do not show lattices, but Fig. 11.43 shows one.

(11.27)

$$\sup\{A,B\} = F, \quad \sup\{A,C\} = F, \quad \sup\{A,D\} = A, \ldots, \quad \sup\{E,F\} = F;$$

$$\inf\{A,B\} = D, \quad \inf\{A,C\} = D, \quad \inf\{A,D\} = D, \ldots, \quad \inf\{E,F\} = E.$$

It is convenient and also very rewarding in the theory of mathematics to employ an operative symbol to represent both the upper and lower bounds.

If X_k is the upper bound of $\{X_i, X_j\}$, we shall express this as

$$X_i \nabla X_j = X_k. \qquad (11.28)$$

If X_l is the lower bound of $\{X_i, X_j\}$, we shall write

$$X_i \triangle X_j = X_l. \qquad (11.29)$$

In a lattice the following properties can be verified: given any three elements A, B, and C belonging to the lattice, we always find

$$\left.\begin{aligned} A \nabla B &= B \nabla A, && (11.30) \\ A \triangle B &= B \triangle A, && (11.31) \end{aligned}\right\} \text{ commutativity,}$$

[1] Let us remember that here we are only considering finite-ordered sets. The lattice is a configuration that can be concerned with both finite and infinite sets.

$$\begin{array}{ll}(11.32) & A\,\nabla(B\,\nabla\,C) = (A\,\nabla\,B)\,\nabla\,C, \\ (11.33) & A\,\triangle(B\,\triangle\,C) = (A\,\triangle\,B)\,\triangle\,C,\end{array}\Bigg\}\quad\text{associativity,}$$

$$\begin{array}{ll}(11.34) & A\,\nabla\,A = A, \\ (11.35) & A\,\triangle\,A = A,\end{array}\Bigg\}\quad\text{idempotence,}$$

$$\begin{array}{ll}(11.36) & A\,\nabla(A\,\nabla\,B) = A, \\ (11.37) & A\,\triangle(A\,\nabla\,B) = A,\end{array}\Bigg\}\quad\text{absorption.}$$

We shall find that with every formula that contains the symbols ∇ and $\triangle$ we can associate another formula in which these symbols are interchanged. This property is often called *duality.*[1]

Diagram of Maximal Chains or "Hasse's Diagram"

For convenience in representing a finite lattice and, in geneıal, a denumerable ordered set, we draw only the maximal chains by means of a nonoriented line between the successive elements forming the total order of this maximal chain.

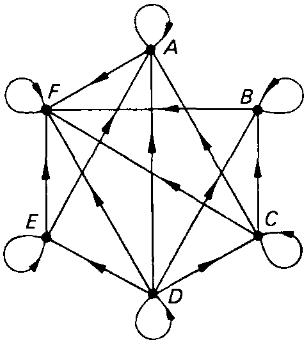


FIG. 11.44

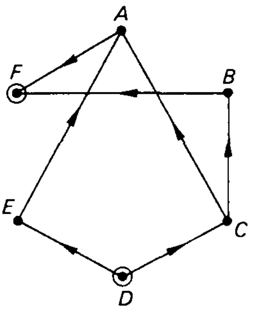


FIG. 11.45

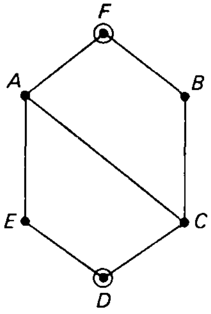


FIG. 11.46

Let us consider as an example the lattice of Fig. 11.44. The maximal chains are

$$D \preccurlyeq E \preccurlyeq A \preccurlyeq F, \qquad D \preccurlyeq C \preccurlyeq A \preccurlyeq F, \qquad D \preccurlyeq C \preccurlyeq B \preccurlyeq F.$$

We now pass from Fig. 11.44 to Fig. 11.45 where only the arrowed lines representing the chains are shown. Finally we move on to Fig. 11.46 in which the lower bound of the lattice is placed at the bottom and the upper bound at the top.

[1] More details about these properties will be found in [K18] and [K20].

Sublattice

A *sublattice* **T**′ of a lattice **T** is a subset of **T** such that if $\triangle$ and ∇ are the symbols for the inferior and superior bounds in **T**, then for all x and y in **T**′: $x \triangle y \in \mathbf{T}'$ and $x \nabla y \in \mathbf{T}'$.

Let us consider the example shown in Fig. 11.47 by means of Hasse's diagram. In Fig. 11.48 a sublattice of the lattice being considered is shown in unbroken lines.

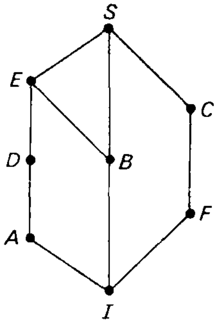


FIG. 11.47

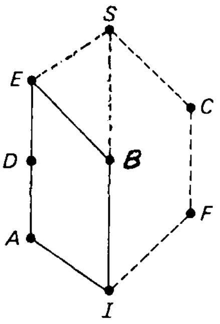


FIG. 11.48

It should be noted that every maximal chain of a lattice is a sublattice of that lattice.

It is also to be noted that, in accordance with this definition, every subset of a lattice **T** that would form a lattice is not always a sublattice of **T**.

4. Distributive Lattice

A lattice is said to be distributive if it conforms to the following conditions that are dual in relation to each other (one is obtained from the other by interchanging ∇ and $\triangle$).

If X_i, X_j, X_k are elements of the lattice,

$$X_i \nabla (X_j \triangle X_k) = (X_i \nabla X_j) \triangle (X_i \nabla X_k), \tag{11.38}$$

$$X_i \triangle (X_j \nabla X_k) = (X_i \triangle X_j) \nabla (X_i \triangle X_k). \tag{11.39}$$

Figure 11.49 shows an example of a distributive lattice. Starting with the tables of relations for ∇ and $\triangle$ given for this lattice in Figs. 11.50 and 11.51 we can easily verify that the relations (11.38 and 11.39) are true for any group of three elements, for instance,

$$B \triangle (D \nabla E) = B \triangle F = B,$$
$$(B \triangle D) \nabla (B \triangle E) = B \nabla B = B.$$

It can easily be verified that every half-lattice of a distributive lattice is itself distributive. Hence every maximal chain of a distributive lattice is a distributive lattice.

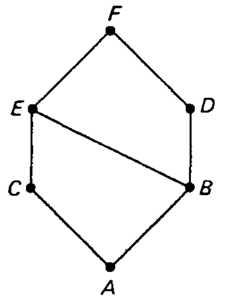


FIG. 11.49

Δ	A	B	C	D	E	F
A	A	A	A	A	A	A
B	A	B	A	B	B	B
C	A	A	C	A	C	C
D	A	B	A	D	A	D
E	A	B	C	A	E	E
F	A	B	C	D	E	F

FIG. 11.50

∇	A	B	C	D	E	F
A	A	B	C	D	E	F
B	B	B	E	D	E	F
C	C	E	C	F	E	F
D	D	D	F	D	F	F
E	E	E	E	F	E	F
F	F	F	F	F	F	F

FIG. 11.51

Free Distributive Lattice with n Generators

Distributive lattices include a particularly important type generated by n sets that do not possess any intersection empty 2 to 2, 3 to 3, ..., n to n. They are known as *free distributive lattices with n generators*. Figures 11.52–11.54 illustrate such lattices with 1, 2, and 3 generators, respectively.

The number of elements in these lattices increases very rapidly in proportion to the number of generators.

n	*Number of elements*
1	1
2	4
3	18
4	166
5	7579
6	7828532

These lattices play an important role in the theory of the availability of systems.[1]

[1] See A. Kaufmann, R. Cruon, and D. Grouchko, "Modèles mathématiques pour l'étude de la fiabilité des systèmes," Masson, Paris 1975.

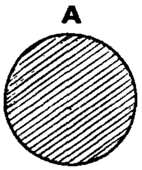


FIG. 11.52

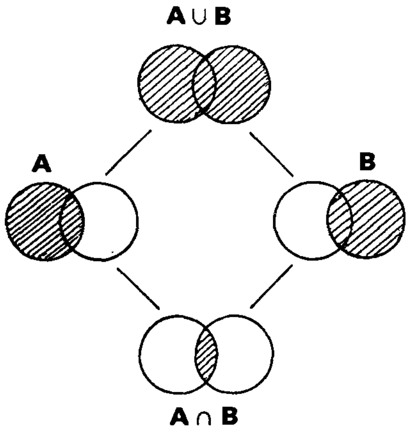


FIG. 11.53

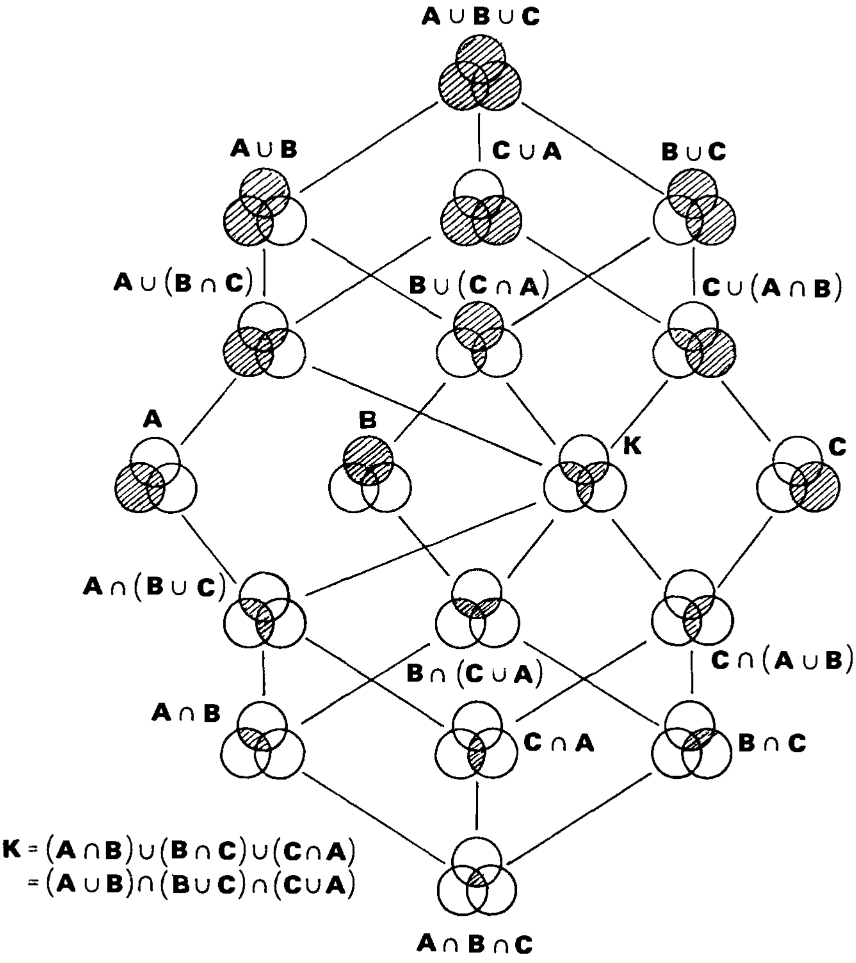


FIG. 11.54

5. Complemented Lattice

Let us take O for the lower bound of a finite lattice $\mathbf{T}$ and U for its upper bound. If for every $X_i \in \mathbf{T}$ there is a X_j such that

$$X_i \triangledown X_j = U \quad \text{and} \quad X_i \triangle X_j = O\,, \tag{11.40}$$

we say that the lattice is complemented. X_j is then called the *complement* or *complementary* of X_i and is written as $\overline{X}_i$.

Let us consider two examples, Figs. 11.55 and 11.56.

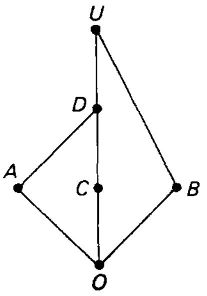


FIG. 11.55

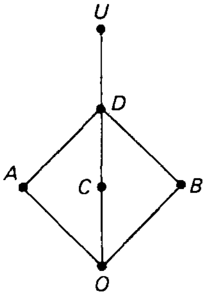


FIG. 11.56

The lattice of Fig. 11.55 is complemented. Indeed

$$\begin{aligned}
\bar{O} &= U \quad \text{since } O \triangledown U = U \text{ and } O \triangle U = O\,;\\
\bar{A} &= B \quad \text{since } A \triangledown B = U \text{ and } A \triangle B = O\,;\\
\bar{B} &= A \quad \text{or} \quad C \quad \text{or} \quad D\\
&\qquad \text{since } B \triangledown A = U \text{ and } B \triangle A = O\,;\\
&\qquad\qquad B \triangledown C = U \text{ and } B \triangle C = O\,;\\
&\qquad\qquad B \triangledown D = U \text{ and } B \triangle D = O\,;\\
\bar{C} &= B \quad \text{since } C \triangledown B = U \text{ and } C \triangle B = O\,;\\
\bar{D} &= B \quad \text{since } D \triangledown B = U \text{ and } D \triangle B = O\,;\\
\bar{U} &= O \quad \text{since } U \triangle O = U \text{ and } U \triangledown O = O\,.
\end{aligned} \tag{11.41}$$

As we see, the complement is not necessarily unique and we do not necessarily have $(\overline{X}_i) = X_i$.

Let us now consider Fig. 11.56.

$$\begin{gathered}
\bar{O} = U \text{ since } O \triangledown U = U \text{ and } O \triangle U = O, \quad \text{but}:\\
A \triangledown O = A, \quad A \triangledown B = D, \quad A \triangledown C = D, \quad A \triangledown D = D, \quad A \triangledown U = U,\\
A \triangle O = O, \quad A \triangle B = O, \quad A \triangle C = O, \quad A \triangle D = A, \quad A \triangle U = A.
\end{gathered} \tag{11.42}$$

No element X_j can be associated with A to satisfy (11.40).

It can be proved (see [K18]) that in a distributive lattice the complement, when it exists, is unique.

6. Boole's Lattice

A lattice that is both distributive and complemented is called a *Boole's lattice* or a *Boolean lattice.*

Figure 11.57 shows an example of such a lattice and the reader is left to determine whether relations (11.38)–(11.40) are verified.

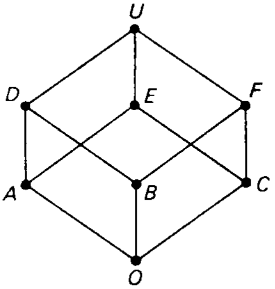


FIG. 11.57

The four main properties of a Boolean trellis are as follows:

1. Every element X_i possesses one and only one complement.
2. For every element X_i: $(\overline{X}_i) = X_i$.
3. Given two elements X_i and X_j,

(11.43) $$\overline{X_i \nabla X_j} = \overline{X}_i \triangle \overline{X}_j,$$

(11.44) $$\overline{X_i \triangle X_j} = \overline{X}_i \nabla \overline{X}_j.$$

4. Every finite Boolean lattice is isomorphic to the lattice constructed starting with the relation of inclusion of the parts of a finite set in that set. In other words, lattices $\mathcal{T}(\mathbf{E})$ of the parts of a set with n elements, ordered by inclusion, is a finite Boolean lattice, and conversely. Hence, for an ordered finite set with n elements, there is one and only one Boolean lattice.

This last property explains why Boole's algebra relating to the parts of a set has the same configuration as a Boolean lattice.

Lastly, the following operations correspond to each other:

Boolean lattice	Boolean algebra	Boole's binary algebra
∇	$\cup$	$(+)$
$\triangle$	$\cap$	$(\cdot)$

It should also be noted that the inverse of the correspondances is equally true. Finally, in the three concepts shown above, the notion of complementation may be considered to be the same.

Boolean Half-Lattice of a Boolean Lattice

Every half-lattice of a Boolean lattice **T** that itself forms a Boolean lattice is called a *Boolean half-lattice* of **T**.

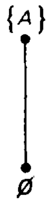


FIG. 11.58

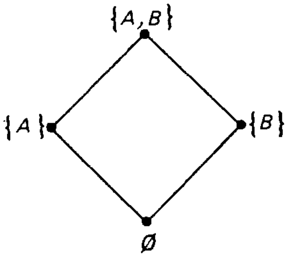


FIG. 11.59

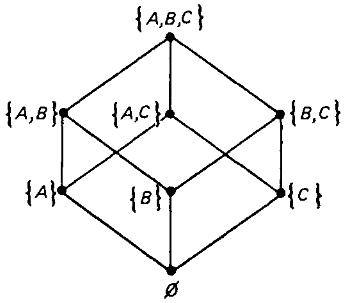


FIG. 11.60

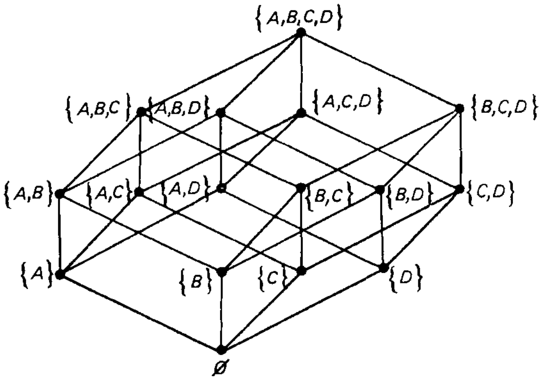


FIG. 11.61

In Figs. 11.62 and 11.63 we can observe an example. The subset

$$\Delta = \{\varnothing, \{B\}, \{C\}, \{A,D\}, \{B,C\}, \{A,B,D\}, \{A,B,C,D\}\} \tag{11.45}$$

forms a Boolean subset constructed by inclusion.

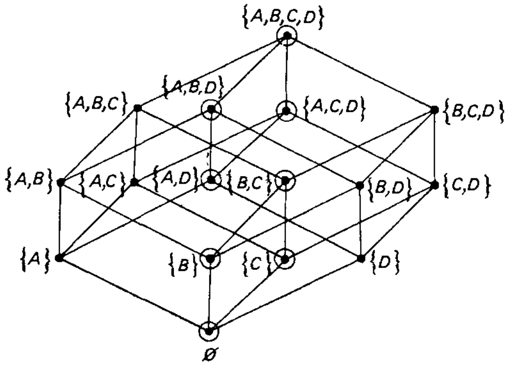


FIG. 11.62

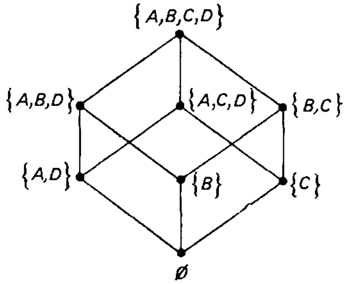


FIG. 11.63

7. Vectorial Lattice

Let us consider n finite sets,

$$\begin{aligned} \mathbf{A} &= \{A_1, A_2, \dots, A_\alpha\}, \\ \mathbf{B} &= \{B_1, B_2, \dots, B_\beta\}, \\ &\dots\dots\dots\dots\dots\dots \\ \mathbf{L} &= \{L_1, L_2, \dots, L_\lambda\}. \end{aligned} \tag{11.46}$$

Let us suppose these sets are totally and strictly ordered, that is to say,

$$\begin{aligned} &A_1 \prec A_2 \prec \dots \prec A_\alpha, \\ &B_1 \prec B_2 \prec \dots \prec B_\beta, \\ &\dots\dots\dots\dots\dots\dots \\ &L_1 \prec L_2 \prec \dots \prec L_\lambda. \end{aligned} \tag{11.47}$$

Let us define a relation of strict order that we will call the *relation of domination* for the elements:

$$[A_i, B_j, \dots, L_l] \in \mathbf{A} \times \mathbf{B} \times \dots \times \mathbf{L}, \tag{11.48}$$

while stating,

$$[A_i, B_j, \dots, L_l] \prec [A_{i'}, B_{j'}, \dots, L_{l'}]. \tag{11.49}$$

If the n-tuple to the left has all its elements less than[1] or equal to those of the n-tuple to the right, and at least one element that is smaller, then the product of the set $\mathbf{A} \times \mathbf{B} \times \dots \times \mathbf{L}$ forms a lattice for this relation of domination.

In Fig. 11.64 we have shown the vectorial lattice obtained from the following ordered sets:

$$\mathbf{A} = \{A_1, A_2\}, \quad \text{where} \quad A_1 \prec A_2; \tag{11.50}$$

$$\mathbf{B} = \{B_1, B_2, B_3\}, \quad \text{where} \quad B_1 \prec B_2 \prec B_3; \tag{11.51}$$

$$\mathbf{C} = \{C_1, C_2\}, \quad \text{where} \quad C_1 \prec C_2. \tag{11.52}$$

Hasse's diagram is shown in Fig. 11.64a while Fig. 11.64b shows a more formal representation corresponding to the Cartesian coordinates.

A Boolean lattice is a vectorial lattice, a property that is evident from Figs. 11.65–11.68.

Lexicagraphical Vectarial Lattice

This is a vectorial lattice that is reduced to a total order (for example, that of a dictionary, whence the name is derived). We consider the following relation of domination: an n-tuple $[A_i, B_j, \dots, L_l]$ will dominate an n-tuple

[1] A similar relation can also be defined by the words *greater than.*

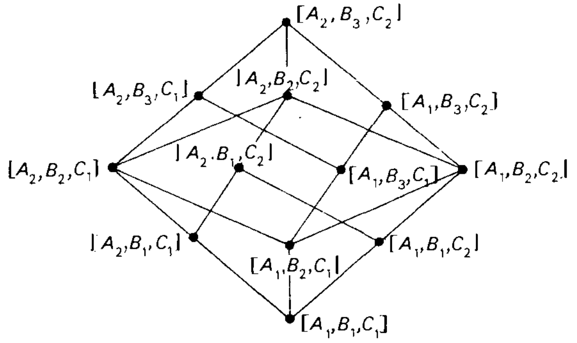


FIG. 11.64a

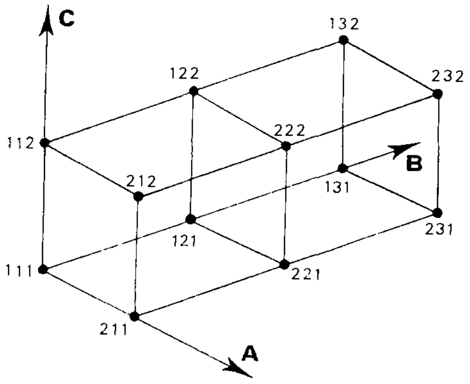


FIG. 11.64b

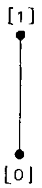


FIG. 11.65

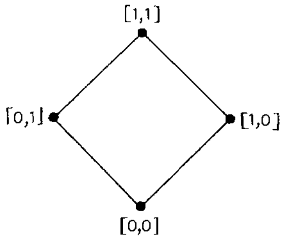


FIG. 11.66

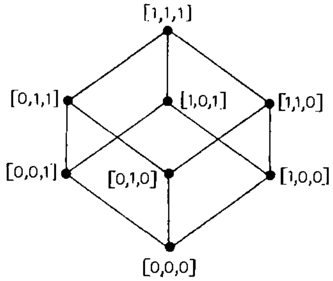


FIG. 11.67

$[A_{i'}, B_{j'}, \ldots, L_{l'}]$ if the first r elements (starting arbitrarily from the left) of the two n-tuples are equal but the $(r+1)$th of the first is superior (in the relation of order concerned) to the $(r+1)$th element of the second. In this way a total order is obtained; for example, [3, 5, 7, 1, 5] dominates [3, 5, 7, 2, 9], and

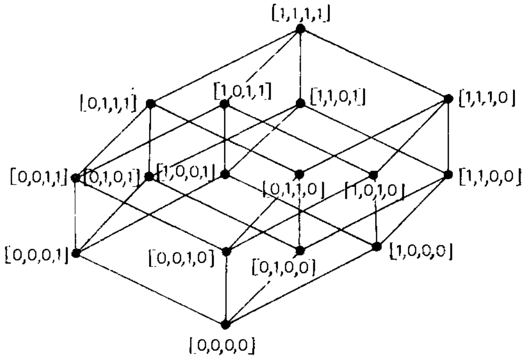


FIG. 11.68

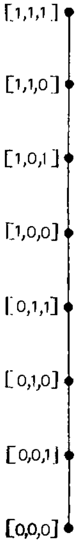


FIG. 11.69

[*R*, *M*, *N*] dominates [*R*, *S*, *B*] if the order selected places *A* before all the other letters, *B* before the remainder, and so forth.

Figure 11.69 provides an example; others have already been given in Figs. 4.5 and 4.6.

Section 12. Other Important Properties of Boole's Binary Algebra

1. Boolean Inequalities

By defining the inequalities between two binary variables *x* and *y* in Boole's binary algebra we can, as will be shown later, obtain Boolean equations and inequations. With this purpose we shall state various equivalences that the reader can easily verify by giving the two variables their possible values of 0 or 1.

We have

(12.1) $$(x \leqslant y) \Leftrightarrow (x \dotplus y = y) \Leftrightarrow (x . y = x).$$

To be certain that these three expressions are equivalent let us employ a *verification* or *true or false* table.

(12.2)

x	y	$x \leq y$	$x \dotplus y = y$	$x . y = x$
0	0	true	true	true
0	0	true	true	true
1	0	false	false	false
1	1	true	true	true

The following relations will be verified in the same manner:

$$(x \leqslant z \text{ and } y \leqslant z) \Leftrightarrow (x+y) \leqslant z, \tag{12.3}$$

$$(x \geqslant z \text{ and } y \geqslant z) \Leftrightarrow x.y \geqslant z, \tag{12.4}$$

$$(x \leqslant y) \Leftrightarrow \bar{x}+y = 1 \Leftrightarrow x.\bar{y} = 0, \tag{12.5}$$

$$(x = y) \Leftrightarrow x.\bar{y}+\bar{x}.y = 0 \Leftrightarrow (\bar{x}+y).(x+\bar{y}) = 1. \tag{12.6}$$

Let us now add the following much less important relations:

$$0 \leqslant x, \tag{12.7}$$

$$x \leqslant 1, \tag{12.8}$$

$$(x \leqslant y) \Rightarrow (x.z \leqslant y.z) \tag{12.9}$$

$$(x \leqslant y) \Rightarrow (x+z) \leqslant (y+z) \tag{12.10}$$

A general property of duality exists between all relations connected with the symbols $+$, ., and $\leqslant$ by replacing $+$ by $\cdot$, $\cdot$ by $+$, and $\leqslant$ by $\geqslant$, 0 by 1, and 1 by 0. This property of duality is one of the important characteristics of Boole's binary algebra that also appears by isomorphism in Boolean algebra, in certain lattices, and in the algebra of functional logic (see [K20]).

Thus, starting from (12.5),

$$(x \leqslant y) \Leftrightarrow \bar{x}+y = 1 \Leftrightarrow x-\bar{y} = 0, \tag{12.11}$$

we can say,

$$(x \geqslant y) \Leftrightarrow \bar{x}\cdot y = 0 \Leftrightarrow x+\bar{y} = 1. \tag{12.12}$$

That can be verified by a different method by replacing x by y and conversely.

Taking (12.9) as a further example, we obtain

$$(x \geqslant y) \Rightarrow (x+z) \geqslant (y+z), \tag{12.13}$$

and we can verify that the inverse implication is false.

2. Boolean Matrices

A Boolean matrix contains elements that cannot be equal except for 0 or 1. The following matrix is, for example, Boolean:

$$[A] = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}; \tag{12.14}$$

it is a 4×5 Boolean matrix, the 4 showing that it contains four lines and the 5 showing that it has five columns.

Boolean matrices have particularly important properties that must be examined.

Given two Boolean matrices $[A]_{m\times n}$ and $[B]_{m\times n}$ the respective elements of which are represented by a_{ij} and b_{ij}, we shall define the *Boolean sum* of the two matrices by a single matrix $[C]_{m\times n}$ such that its elements C_{ij} are

$$c_{ij} = a_{ij} \dotplus b_{ij}, \qquad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n. \tag{12.15}$$

Example

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \dotplus \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{12.16}$$

Let us now consider what is meant by the *Boolean product* of two Boolean matrices. Given two such matrices $[A]_{m\times r}$ and $[B]_{r\times n}$, we define their Boolean product by a matrix $[C]_{m\times n}$ the elements C_{ij} of which are such that

$$c_{ij} = a_{i1} . b_{1j} \dotplus a_{i2} . b_{2j} \dotplus \ldots \dotplus a_{ir} . b_{rj}, \qquad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n. \tag{12.17}$$

We shall indicate the Boolean product of two matrices by the symbol $\circ$.

Example

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \circ \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 1 \\ 0 & 0 \end{bmatrix}. \tag{12.18}$$

We also define the *complementation* or *negation* of a Boolean matrix $[A]_{m\times n}$ with elements a_{ij} by a matrix $[B]_{m\times n}$ such that

$$b_{ij} = \bar{a}_{ij}, \qquad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n. \tag{12.19}$$

Example

(12.20)

$$[A] = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \qquad [B] = [\bar{A}] = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

We can also define the *conjunction* of two Boolean matrices $[A]_{m\times n}$ and

$[B]_{m \times n}$ by a matrix $[C]_{m \times n}$ of which the elements c_{ij} are such that

$$c_{ij} = a_{ij} \, . \, b_{ij}, \qquad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n. \tag{12.21}$$

We shall indicate this operation by the symbol $\ddot{+}$.

Example

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \ddot{+} \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{12.22}$$

The *disjunctive sum* of two Boolean matrices $[A]_{m \times n}$ and $[B]_{m \times n}$ is given by a matrix $[C]_{m \times n}$ of which the elements c_{ij} are such that

$$c_{ij} = a_{ij} \oplus b_{ij} = a_{ij} . \bar{b}_{ij} + \bar{a}_{ij} . b_{ij}, \qquad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n. \tag{12.23}$$

Example

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \oplus \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{12.24}$$

The *disjunctive product* of two Boolean matrices $[A]_{m \times n}$ and $[B]_{m \times n}$ is given by a matrix $[C]_{m \times n}$ of which the elements c_{ij} are such that

$$c_{ij} = a_{i1} . b_{1j} \oplus a_{i2} . b_{2j} \oplus \ldots \oplus a_{ir} . b_{rj}, \qquad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n. \tag{12.25}$$

We shall indicate this operation by the sign $\odot$.

Example

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \odot \begin{bmatrix} 0 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix}. \tag{12.26}$$

Finally, we say that a matrix $[B]_{m \times n}$ *dominates* a matrix $[A]_{m \times n}$ if we have

$$b_{ij} \geqslant a_{ij}, \qquad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n. \tag{12.27}$$

We then state,

$$[B] \geqslant [A]. \tag{12.28}$$

If for every i and j we have

$$(12.29) \qquad b_{ij} > a_{ij}, \qquad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n,$$

we say that $[B]$ *strictly dominates* $[A]$.

Example

$$(12.30) \qquad \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \geqslant \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

In this example the symbol $>$ could also be used.

It must be observed that two Boolean matrices $m \times n$ are not necessarily comparable (one does not necessarily dominate the other). The reader should not find it difficult to discover a contrary example.

The relation of domination between Boolean matrices $m \times n$ possesses the following properties:

$$(12.31) \qquad [A] \leqslant [A],$$

$$(12.32) \qquad ([A] \leqslant [B] \text{ and } [B] \leqslant [A]) \Leftrightarrow ([A] = [B]),$$

$$(12.33) \qquad ([A] \leqslant [B] \text{ and } [B] \leqslant [C]) \Rightarrow ([A] \leqslant [C]),$$

$$(12.34) \qquad ([A] \leqslant [B]) \Rightarrow ([A] \circ [C] \leqslant [A] \circ [C]),$$

if these three matrices are square.

Of course, all the general properties of matrices are valid for Boolean matrices and we shall not recapitulate them here since it would involve a course in matrical calculation, but the reader is referred to works that have already been mentioned. In the usual accounts of matrices we consider the ordinary algebraic sum indicated by $+$ and the product shown as . or $\times$, whereas here we are concerned with other operations that are defined by (12.15), (12.17), (12.19), (12.21), and (12.25). It is possible to define many others of a greater or lesser value, but we shall concern ourselves mainly with the properties connected with the operations $\dotplus$ and $\circ$:

$$(12.35) \qquad ([A] \dotplus [B]) \dotplus [C] = [A] \dotplus ([B] \dotplus [C]) \qquad \text{associativity for } \dotplus;$$

$$(12.36) \qquad ([A] \circ [B]) \circ [C] = [A] \circ ([B] \circ [C]) \qquad \text{associativity for } \circ;$$

$$(12.37) \qquad [A] \circ ([B] \dotplus [C]) = [A] \circ [B] \dotplus [A] \circ [C] \qquad \text{distributivity to the left;}$$

$$(12.38) \qquad ([A] \dotplus [B]) \circ [C] = [A] \circ [C] \dotplus [B] \circ [C] \qquad \text{distributivity to the right.}$$

The reader should try to discover similar properties that are verified for suitable associations of the operations defined above.

Let us now consider other properties concerned with square Boolean matrices. In doing so let us employ the usual exponential symbol for the Boolean product of two Boolean matrices, and if other operations give cause for confusion we shall provide the necessary warning at the required time.

Hence, for a square Boolean matrix, we can state,

$$(12.39)\qquad [A]^r = \underbrace{[A]\circ[A]\circ\ldots\circ[A]}_{r\ \text{times}}$$

and it can easily be verified that we have

$$(12.40)\qquad [A]^r = [A]^{r-1}\circ[A] = [A]\circ[A]^{r-1},$$

$$(12.41)\qquad [A]^r\circ[A]^s = [A]^s\circ[A]^r = [A]^{r+s},$$

$$(12.42)\qquad ([A]^r)^s = ([A]^s)^r = [A]^{rs}.$$

An important special case concerns square Boolean matrices in which the principal diagonal is composed of 1, that is to say, such that $a_{ii} = 1$, $i = 1, 2, \ldots, n$, where n is the order of the square matrix. In this case we find the interesting property[1]

$$(12.43)\qquad [1] \leqslant [A] \leqslant [A]^2 \leqslant \ldots \leqslant [A]^{n-1} = [A]^n = [A]^{n+1} = \ldots .$$

Before giving the proof, let us take an example from our reference [K14].

$$[A] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}, \qquad [A]^2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix},$$

$$(12.44)\qquad [A]^3 = [A]^2\circ[A] = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix},$$

$$[A]^4 = [A]^3\circ[A] = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} \circ \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}.$$

[1] We shall represent as 1 every unit matrix, that is to say, a matrix such that

$$\begin{aligned} a_{ij} &= 0, \quad i \neq j, \\ &= 1, \quad i = j. \end{aligned}$$

It is known that such matrices play a unit role in the matrical product.

It can be seen that we have

$$(12.45)\qquad \underset{[1]}{\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}} < \underset{[A]}{\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 1\\ 1 & 0 & 1\end{bmatrix}} < \underset{[A]^2}{\begin{bmatrix}1 & 0 & 0\\ 1 & 1 & 1\\ 1 & 0 & 1\end{bmatrix}}$$

$$= \underset{[A]^3}{\begin{bmatrix}1 & 0 & 0\\ 1 & 1 & 1\\ 1 & 0 & 1\end{bmatrix}} = \underset{[A]^4}{\begin{bmatrix}1 & 0 & 0\\ 1 & 1 & 1\\ 1 & 0 & 1\end{bmatrix}} = \dots .$$

To prove property (12.43) let us first return to (12.17). By representing the elements of $[A]^r$ as $a_{ij}^{(r)}$ we obtain

$$(12.46)\qquad a_{ij}^{(2)} = \overset{\circ}{\sum_{\alpha}} a_{i\alpha} \cdot a_{\alpha j} \text{ ,}$$

where $\overset{\circ}{\sum}$ indicates that we perform the summation in accordance with formula (12.17). Likewise

$$(12.47)\qquad a_{ij}^{(3)} = \overset{\circ}{\sum_{\alpha_2}} \overset{\circ}{\sum_{\alpha_1}} a_{i\alpha_1} \cdot a_{\alpha_1 \alpha_2} \cdot a_{\alpha_2 j}$$

and, in a more general way,

$$(12.48)\qquad a_{ij}^{(r)} = \overset{\circ}{\sum_{\alpha_1, \alpha_2, \dots, \alpha_r}} a_{i\alpha_1} \cdot a_{\alpha_1 \alpha_2} \cdot \dots \cdot a_{\alpha_{r-1}, j} \cdot$$

By the definition of $[A]$ it is clear that we first have

$$(12.49)\qquad [1] \leqslant [A] \cdot$$

Because of (12.34) we can state,

$$(12.50)\qquad [A] \leqslant [A]^2 \leqslant [A]^3 \leqslant \dots \leqslant [A]^{r-1} \leqslant [A]^r \leqslant [A]^{r+1} \leqslant \dots$$

Let us now show that if $r = n$ where n is the order of the square matrix $[A]$, then

$$(12.51)\qquad [A]^{r-1} \geqslant [A]^r \geqslant [A]^{r+1} \geqslant \dots .$$

To do this let us consider one of the terms $a_{i\alpha_1} \cdot a_{\alpha_1 \alpha_2} \cdot \dots \cdot a_{\alpha_{r-1}, j}$ of (12.48) for $r = n$. Since there cannot be $n+1$ separate indices $i, \alpha_1, \alpha_2, \dots, \alpha_{n-1}, j$, there must be an $h < k$ such that $\alpha_h = \alpha_k$. Hence, we can write the right-hand expression of (12.48) thus:

$$(12.52)\qquad \begin{aligned} &a_{i\alpha_1} \cdot a_{\alpha_1\alpha_2} \cdot \dots \cdot a_{\alpha_{h-1}\alpha_h} \cdot a_{\alpha_h\alpha_{h+1}} \cdot \dots \cdot a_{\alpha_{k-1}\alpha_k} \cdot a_{\alpha_k\alpha_{k+1}} \cdot \dots \cdot a_{\alpha_{n-1},\,j} \\ &\quad\leqslant a_{i\alpha_1} \cdot a_{\alpha_1\alpha_2} \cdot \dots \cdot a_{\alpha_{h-1}\alpha_h} \cdot a_{\alpha_h\alpha_{h+1}} \cdot \dots \cdot a_{\alpha_{n-1},\,j}\,. \end{aligned}$$

In the second member of (12.52) there are at most $(n-1)$ factors. By supposing that there are less than $(n-1)$ we can complete the product with factors of the form $a_{\alpha_k\alpha_h} = a_{\alpha_k\alpha_k} = a_{\alpha_h\alpha_h} = 1$, in such a way that

$$(12.53)\qquad \begin{aligned} &a_{i\alpha_1} \cdot a_{\alpha_1\alpha_2} \cdot \dots \cdot a_{\alpha_{n-1},\,j} \\ &\quad\leqslant a_{i\alpha_1} \cdot a_{\alpha_1\alpha_2} \cdot \dots \cdot a_{\alpha_{h-1}\alpha_h} \cdot a_{\alpha_h\alpha_{h+1}} \cdot \dots \cdot a_{\alpha_{k-1}\alpha_k} \cdot a_{\alpha_k\alpha_{k+1}} \cdot \dots \cdot a_{\alpha_{n-1},\,j}\,, \end{aligned}$$

the right member of (12.53) having $(n-1)$ factors.

But this member is a term that belongs to the expansion, in accordance with (12.48), of $a_{ij}^{(n-1)}$. Since the relation (12.53) is true for all the possible values of the indices $\alpha_1, \alpha_2, \dots, \alpha_{n-1}$, we conclude $a_i^{(n)} \leqslant a_{ij}^{(n-1)}$. Since this is true for every i and j from 1 to n, we deduce that

$$(12.54)\qquad [A]^n \leqslant [A]^{n-1}.$$

From the consideration of (12.50) and (12.54) we certainly obtain (12.43).

The first exponent k for which $[A]^k = [A]^{k+1}$ is known as the *characteristic exponent* of $[A]$. Hence, in accordance with (12.43) we have $k \leqslant n-1$ and

$$(12.55)\qquad \begin{aligned} &[A]^r < [A]^k, &\quad r < k, \\ &[A]^r = [A]^k, &\quad r \geqslant k. \end{aligned}$$

3. Boolean Determinants

For a square Boolean matrix $[A]$ with elements a_{ij}, $i, j = 1, 2, \dots, n$, we describe as the *Boolean determinant* of $[A]$ the number

$$(12.56)\qquad \dot{\det}[A] = \dot{\sum_{\alpha_1, \alpha_2, \dots, \alpha_n}} a_{1\alpha_1} \cdot a_{2\alpha_2} \cdot \dots \cdot a_{n\alpha_n}$$

where the dot over *det* indicates that we are not concerned with the vulgar determinant, and the dot over the summation sign indicates that we are dealing with a Boolean summation in accordance with (12.15). Finally, the $\alpha_1, \alpha_2, \dots, \alpha_n$ show that the summation must be performed for all the permutations of the indices α_r, $r = 1, 2, \dots, n$. This is the determinant that is so well-known in classic matrical calculation, but here the operation $+$ is replaced by the operation $\dot{+}$.

Example

Given

$$(12.57)\qquad [A]=\begin{bmatrix}1 & 0 & 1\\ 0 & 1 & 1\\ 1 & 0 & 0\end{bmatrix}.$$

We have

$$(12.58)\qquad \overset{\bullet}{\det}\begin{bmatrix}1 & 0 & 1\\ 0 & 1 & 1\\ 1 & 0 & 0\end{bmatrix}=(1).(1.0\dotplus 0.1)\dotplus(0).(0.0\dotplus 0.1)\dotplus(1).(0.1\dotplus 1.1)=1.$$

If we employ M_{ij} for the minor of a_{ij}, defined as the Boolean determinant of matrix $[A]$ when deprived of its line i and its column j, we can then say,

$$(12.59)\qquad \overset{\bullet}{\det}[A]=a_{11}.M_{11}\dotplus a_{21}.M_{21}\dotplus\dots\dotplus a_{n1}.M_{n1},$$

an expansion that can be extended to any line or column, as occurs in Laplace's expansion in classic matricial calculation, although here the operation $\dotplus$ would be used.

Let us now consider some properties of Boolean determinants.

1. The value of a Boolean determinant is unchanged if we permutate the lines or the columns.

2. The value is unchanged if we permutate the lines with the columns.

3. If two columns (or two lines) are identical the determinant is not necessarily null, although in the case of a vulgar determinant we know that it is. This requires a simple counterexample.

$$(12.60)\qquad \overset{\bullet}{\det}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}=(1).(1)\dotplus(1).(1)=1\dotplus 1=1.$$

4. If $\alpha\in\{0,1\}$ and we multiply a line (or column) by α, then the Boolean determinant is also multiplied by α.

5. A Boolean determinant that possesses a line (or column) in which all the elements are 0 is equal to 0.

6. In a Boolean determinant if we multiply a line (or column) by $\alpha\in\{0,1\}$ and if we add ($\dotplus$) this line (or column) to another line (or column) we do not necessarily obtain the same determinant as occurs with vulgar determinants. This can be shown by a counterexample.

$$(12.61)\qquad \overset{\bullet}{\det}\begin{bmatrix}0 & 1\\ 0 & 1\end{bmatrix}=0,\qquad \overset{\bullet}{\det}\begin{bmatrix}0\dotplus 1 & 1\\ 0\dotplus 1 & 1\end{bmatrix}=\overset{\bullet}{\det}\begin{bmatrix}1 & 1\\ 1 & 1\end{bmatrix}=1.$$

We define the *conjugate Boolean matrix* $[A]^*$ of a Boolean matrix $[A]$ as follows, the elements a_{ij}^* of $[A]^*$ being the Boolean minors M_{ij} of $[A]$:

$$(12.62) \qquad [A] = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \qquad [A]^* = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}.$$

It can be proved (see [K14], p. 14) that if $[A]$ is such that $a_{ii} = 1$, $i = 1, 2, \ldots, n$. then

$$(12.63) \qquad [A]^* = [A]^k,$$

where k is the characteristic exponent.

4. Boolean Binary Functions

Given a function $f(x_1, x_2, \ldots, x_n)$ of n binary variables $x_i \in \{0, 1\}$, $i = 1, 2, \ldots, n$, such that the operations that take place between the variables can only be $(.)$, $(+)$, and $(^-)$ or can always be reduced to these operations, we say that these functions are *Boolean binary* or more simply, where no confusion is possible, *Boolean functions*.

Example

$$(12.64) \qquad f(x_1, x_2, x_3, x_4) = x_1.\bar{x}_2 + x_2.\bar{x}_3.x_4 + x_3$$

is a Boolean function.

It was shown in Section 10 that every function with binary values can be expressed in a unique form known as *canonical disjunctive* and in another unique form known as *canonical conjunctive* (see (10.33) and (10.37)).

Thus, let

$$(12.65) \qquad f(x_1, x_2, x_3) = x_1 . \bar{x}_2 + x_2 . x_3 .$$

This function expressed in its canonical disjunctive form is, in accordance with (10.30),

$$(12.66) \qquad f(x_1, x_2, x_3) = x_1.x_2.x_3 + x_1.\bar{x}_2.\bar{x}_3 + x_1.\bar{x}_2.x_3 + \bar{x}_1.x_2.x_3$$

and, in its canonical conjunctive form,

$$(12.67) \qquad f(x_1, x_2, x_3) = (x_1 + x_2 + x_3).(x_1 + x_2 + \bar{x}_3).(x_1 + \bar{x}_2 + x_3) \\ .(\bar{x}_1 + \bar{x}_2 + x_3).$$

We say that two Boolean binary functions are identical if they have the same canonical disjunctive form and/or the same canonical conjunctive form, that is to say, that they still possess the same minterms or the same maxterms.

5. Pseudo-Boolean Functions[1]

The term *pseudo-Boolean function* is given to a function of binary variables that takes its values from the set **Z** of the related integers. Here are two examples:

$$(12.68) \qquad f(x_1, x_2, x_3) = 3x_1\bar{x}_2 - 2x_3,$$

$$(12.69) \qquad f(x_1, x_2, x_3, x_4) = 8x_1^2 x_2 \bar{x}_3 + x_1 x_2 x_3 \bar{x}_4 + x_2 x_4$$

are pseudo-Boolean functions.

These functions possess an important characteristic property: they are always linear in relation to each of the variables that occur in them. By resuming the reasoning[2] established from (10.22) to (10.30), it can be seen that formula (10.30) remains valid whatever the nature of $\varphi(x_1, x_2, \ldots, x_n)$. Hence a pseudo-Boolean function can always be expressed in a canonical disjunctive form.

Let us take an example. Let

$$(12.70) \qquad f(x_1, x_2) = 5x_1\bar{x}_2 - 3x_2.$$

We have successively

(12.71)

$$f(0,0) = 0, \qquad f(0,1) = -3, \qquad f(1,0) = 5, \qquad f(1,1) = -3.$$

Then, for the canonical disjunctive form,

$$(12.72) \qquad f(x_1, x_2) = -3\bar{x}_1 x_2 + 5x_1\bar{x}_2 - 3x_1 x_2.$$

Or again by taking $\bar{x}_i = 1 - x_i$,

$$(12.73) \qquad f(x_1, x_2) = 5x_1 - 3x_2 - 5x_1 x_2.$$

We say that two pseudo-Boolean functions are identical if they have the same canonical disjunctive form, that is to say, they possess the same terms preceded by the same coefficients in this canonical form.

Owing to the very marked difference between the operations $(+)$ and $(\dotplus)$ we can only obtain a canonical conjunctive form for pseudo-Boolean functions at the expense of much greater complications.

[1] This term is due to P. Hammer [K14].

[2] We are concerned here with common and not Boolean addition, though the reasoning remains the same.

Section 13. Solutions for Boolean Equations and Inequations

1. Method of the Table of Binary Values. Solution by Complete Enumeration

Given the equation,

$$f(x_1, x_2, \dots, x_n) = g(x_1, x_2, \dots, x_n); \tag{13.1}$$

a method for solving it is to complete the table of values for f and g for all values of x_i and to compare the results. This method can easily be explained by means of an example. Let

$$x_1 . \bar{x}_2 \dotplus x_3 = \bar{x}_1 . x_2 . x_3 \dotplus \bar{x}_2 .$$

Let us draw up the following table:

(13.2)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
x_1	x_2	x_3	$\bar{x}_2$	$x_1\bar{x}_2$	$x_1\bar{x}_2 \dotplus x_3$	$\bar{x}_1$	x_2x_3	$\bar{x}_1x_2x_3$	$\bar{x}_1x_2x_3 \dotplus \bar{x}_2$
0	0	0	1	0	0	1	0	1	0
0	0	1	1	0	1	1	0	1	0
0	1	0	0	0	0	1	0	0	0
0	1	1	0	0	1	1	1	1	1
1	0	0	1	1	1	0	0	1	0
1	0	1	1	1	1	0	0	1	0
1	1	0	0	0	0	0	0	0	0
1	1	1	0	0	1	0	1	0	0

By comparing columns (6) and (10) we find that the solutions are given by the following values of (x_1, x_2, x_3):

$$(0,0,1),\quad (0,1,0),\quad (0,1,1),\quad (1,0,0),\quad (1,0,1),\quad (1,1,0). \tag{13.3}$$

Let us consider another example. Let

$$x_1 . \bar{x}_2 . x_3 \dotplus x_2 . x_4 \dotplus x_3 . \bar{x}_4 = 0 . \tag{13.4}$$

(13.5)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
x_1	x_2	x_3	x_4	$\bar{x}_2$	$x_1\bar{x}_2x_3$	x_2x_4	$\bar{x}_4$	$x_3\bar{x}_4$	$x_1\bar{x}_2x_3 \dotplus x_2x_4 \dotplus x_3\bar{x}_4$
0	0	0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0	0	0
0	0	1	0	1	0	0	1	1	1
0	0	1	1	1	0	0	0	0	0
0	1	0	0	0	0	0	1	0	0
0	1	0	1	0	0	1	0	0	1
0	1	1	0	0	0	0	1	1	1
0	1	1	1	0	0	1	0	0	1
1	0	0	0	1	0	0	1	0	0
1	0	0	1	1	0	0	0	0	0
1	0	1	0	1	1	0	1	1	1
1	0	1	1	1	1	0	0	0	1
1	1	0	0	0	0	0	1	0	0
1	1	0	1	0	0	1	0	0	1
1	1	1	0	0	0	0	1	1	1
1	1	1	1	0	0	1	0	0	1

The solutions are

(13.6) $(0,0,0,0)$, $(0,0,0,1)$, $(0,0,1,1)$, $(0,1,0,0,)$ $(1,0,0,0)$,
$(1,0,0,1)$, $(1,1,0,0)$.

The inequations are solved in the same manner. Let

(13.7) $$x_1x_2\bar{x}_3+\bar{x}_1x_2 \leqslant x_1\bar{x}_2x_3 .$$

(13.8)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
$\bar{x}_1$	x_2	x_3	$\bar{x}_3$	$x_1x_2\bar{x}_3$	$\bar{x}_1$	$\bar{x}_1x_2$	$x_1x_2\bar{x}_3 \dotplus \bar{x}_1x_2$	$\bar{x}_2$	$x_1\bar{x}_2x_3$
0	0	0	1	0	1	0	0	1	0
0	0	1	0	0	1	0	0	1	0
0	1	0	1	0	1	1	1	0	0
0	1	1	0	0	1	1	1	0	0
1	0	0	1	0	0	0	0	1	0
1	0	1	0	0	0	0	0	1	1
1	1	0	1	1	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0

By comparing columns (8) and (10) we find that the solutions are

$$(0,0,0),\quad (0,0,1),\quad (1,0,0),\quad (1,0,1),\quad (1,1,1). \tag{13.9}$$

2. Systems of Boolean Equations

Let us consider a system of Boolean equations of the form

$$f_i(x_1, x_2, \dots, x_n) = 0, \qquad i = 1, 2, \dots, m. \tag{13.10}$$

It is clear that this system of equations has exactly the same solutions as the equation

$$\overset{\bullet}{\sum_{i=1}^{m}} f_i(x_1, x_2, \dots, x_n) = 0. \tag{13.11}$$

For, if the Boolean sum is null all the elements of the sum must be null and conversely.

Let us now examine a system of Boolean equations of the form

$$g_i(x_1, x_2, \dots, x_n) = 1, \qquad i = 1, 2, \dots, m. \tag{13.12}$$

It is clear that this system has exactly the same solutions as the equation

$$\overset{\bullet}{\prod_{i=1}^{m}} g_i(x_1, x_2, \dots, x_n) = 1. \tag{13.13}$$

for, if the product is equal to 1, all the elements of the product must be equal to 1 and conversely.

If the system of equations contains equations of the form of (13.10) and (13.11), we reduce them to (13.12) or (13.13) by stating

$$f_r = 0 \Leftrightarrow \bar{f}_r = 1 \qquad \text{or} \qquad g_s = 1 \Leftrightarrow \bar{g}_s = 0. \tag{13.14}$$

Let us consider an example.

Let

$$x_1\bar{x}_2x_3 \dotplus \bar{x}_1 = 0, \tag{13.15}$$

$$x_1x_2 \dotplus \bar{x}_3 = 1, \tag{13.16}$$

$$x_1x_2\bar{x}_3 = 1. \tag{13.17}$$

Let us take the complementary of (13.16) and (13.17) so as to have null second members throughout. It follows that

$$\overline{x_1x_2 \dotplus \bar{x}_3} = \overline{x_1x_2}\,x_3 = (\bar{x}_1 \dotplus \bar{x}_2)\,x_3 = \bar{x}_1x_3 \dotplus \bar{x}_2x_3 = 0, \tag{13.18}$$

$$\overline{x_1x_2\bar{x}_3} = \overline{x_1x_2} \dotplus x_3 = \bar{x}_1 \dotplus \bar{x}_2 \dotplus x_3 = 0. \tag{13.19}$$

Whence the equivalent system,

(13.20) $$x_1\bar{x}_2x_3 \dotplus \bar{x}_1 = 0,$$

(13.21) $$\bar{x}_1x_3 \dotplus \bar{x}_2x_3 = 0,$$

(13.22) $$\bar{x}_1 \dotplus \bar{x}_2 \dotplus x_3 = 0.$$

Or again,

(13.23) $$\underset{(1)}{x_1\bar{x}_2x_3} \dotplus \underset{(2)}{\bar{x}_1} \dotplus \underset{(3)}{\bar{x}_1x_3} \dotplus \underset{(4)}{\bar{x}_2x_3} \dotplus \underset{(5)}{\bar{x}_1} \dotplus \underset{(6)}{\bar{x}_2} \dotplus \underset{(7)}{x_3} = 0,$$

that we shall simplify as follows:

(13.24) $$(2) \dotplus (3) \dotplus (5) : \qquad \bar{x}_1 \dotplus \bar{x}_1x_3 \dotplus \bar{x}_1 = \bar{x}_1,$$

(13.25) $$(1) \dotplus (4) \dotplus (6) : \qquad x_1\bar{x}_2x_3 \dotplus \bar{x}_2x_3 \dotplus \bar{x}_2 = \bar{x}_2;$$

(13.23) can then be expressed

(13.26) $$\bar{x}_1 \dotplus \bar{x}_2 \dotplus x_3 = 0.$$

In accordance with the following table:

(13.27)

x_1	x_2	x_3	$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_1 \dotplus \bar{x}_2 \dotplus x_3$
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	1	0	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	0	0	0
1	1	1	0	0	1

the only solution is

(13.28) (1, 1, 0).

3. Systems of Boolean Equations and Inequations

Relations (12.5) and (12.6) enable us to reduce any equation or inequation to a form $f = 0$ or $g = 1$.

If f and g are Boolean functions of the variables $x_1, x_2, \ldots, x_n$, we have

(13.29) $$(f \leqslant g) \Leftrightarrow \bar{f} \dotplus g = 1 \Leftrightarrow f.\bar{g} = 0,$$

(13.30) $$(f = g) \Leftrightarrow f.\bar{g} \dotplus \bar{f}.g = 0 \Leftrightarrow (\bar{f} \dotplus g).(f \dotplus \bar{g}) = 1.$$

Let us now see how a system of equations and inequations can be solved. Let

(13.31) $$\bar{x}_1 x_2 \dotplus \bar{x}_2 x_3 = 0,$$

(13.32) $$x_1 x_3 \leqslant \bar{x}_2,$$

(13.33) $$x_1 \bar{x}_2 \dotplus x_3 = x_2.$$

We have successively, by reduction to a form $F = 0$:

for (13.32):

(13.34) $$(x_1 x_3) . x_2 = 0, \quad \text{that is} \quad x_1 x_2 x_3 = 0;$$

for (13.33):

(13.35) $$(x_1 \bar{x}_2 \dotplus x_3)\, \bar{x}_2 \dotplus \overline{(x_1 \bar{x}_2 \dotplus x_3)}\, x_2 = 0,$$

that is

(13.36) $$x_1 \bar{x}_2 \dotplus \bar{x}_2 x_3 \dotplus x_2 \bar{x}_3 = 0.$$

From which finally

(13.37) $$\bar{x}_1 x_2 \dotplus \bar{x}_2 x_3 = 0,$$

(13.38) $$x_1 x_2 x_3 = 0,$$

(13.39) $$x_1 \bar{x}_2 \dotplus \bar{x}_2 x_3 \dotplus x_2 \bar{x}_3 = 0.$$

We leave the reader the task of discovering the solution (or solutions) if one exists.

Let us now consider the case of a strict inequation of the form

(13.40) $$f < g.$$

Let us say that

(13.41) $$f < g \Leftrightarrow f = 0 \text{ and } g = 1,$$

or again,

(13.42) $$f < g \Leftrightarrow \bar{f} = 1 \text{ and } \bar{g} = 0.$$

Hence we can now form two equations each of which is equivalent to (13.40), namely,

(13.43) $$(f < g) \Leftrightarrow \bar{f} . g = 1,$$

and

(13.44) $$(f < g) \Leftrightarrow f \dotplus \bar{g} = 0 .$$

Lastly, let us consider the case of inequations with the following forms:

(13.45) $$\begin{aligned} &f \leqslant 0 \Leftrightarrow f = 0 , \\ &f < 0 \quad \text{impossible}, \\ &f \geqslant 0 \quad \text{always true}, \\ &f > 0 \Leftrightarrow f = 1 . \end{aligned}$$

Hence all the equations or inequations, all the systems composed of equations and/or inequations can be reduced to an equation of the type of (13.11) and/or of (13.13).

4. Arborescent Method (Method of Branchings)

Once the number of variables exceeds four it becomes difficult and laborious to complete the enumeration table, so that it is desirable to avoid such enumeration. The following method is designed to reduce it, and will be explained by means of two examples. The first is of a purely instructional kind concerned with an equation with four unknowns, for which the method of complete enumeration might, in fact, prove easier.

First Example

Let us return to (13.4):

(13.46) $$x_1 \bar{x}_2 x_3 \dotplus x_2 x_4 \dotplus x_3 \bar{x}_4 = 0 .$$

Let us arbitrarily commence with the variable that occurs most often,[1] namely x_2 (we could equally have chosen x_4). Let us suppose

(13.47) $$x_2 = 1 .$$

Substituting (13.47) in (13.46), we obtain

(13.48) $$x_4 \dotplus x_3 \bar{x}_4 = 0 .$$

The consideration of this equation at once shows us that we must have $x_4 = 0$ and hence $x_3 = 0$. If we substitute these values and $x_2 = 1$ in (13.46) this equation is equally verified. Thus we can take $x_1 = 0$ or $x_1 = 1$ and have found two solutions,

(13.49) $$[0,1,0,0] \quad \text{and} \quad [1,1,0,0] .$$

Beginning with R in the arborescence of Fig. (13.1) let us mark the vertices

[1] There is no formal reason for beginning in this way, but we judge that it will provide a greater chance of progressing with increasingly simple formulas.

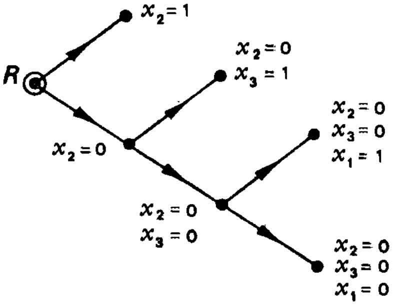


FIG. 13.1

corresponding to $x_1 = 1$ and $x_2 = 0$. Let us proceed by taking $x_2 = 0$ in (13.46) and let us then choose another variable x_3 that occurs most frequently (though this choice is not essential) and that we make equal to 1.

Substituting

(13.50) $$x_2 = 0, \quad x_3 = 1,$$

in (13.46); it follows

(13.51) $$x_1 \dotplus \bar{x}_4 = 0.$$

From which we conclude that $x_1 = 0$ and $x_4 = 1$. This gives a fresh solution,

(13.52) $$[0, 0, 1, 1].$$

Let us proceed by making $x_2 = 0$, $x_3 = 0$ in (13.46) and let us select another variable x_1 to which we shall give the value of 1. Let us now substitute

(13.53) $$x_2 = 0, \quad x_3 = 0, \quad x_1 = 1,$$

in (13.46) (this equation is verified whatever the value of x_4). We now have two further solutions,

(13.54) $$[1, 0, 0, 0] \quad \text{and} \quad [1, 0, 0, 1].$$

Lastly, let us make

(13.55) $$x_2 = 0, \quad x_3 = 0, \quad x_1 = 0, \quad x_4 = 1.$$

We can verify that this is a solution, namely,

(13.56) $$[0, 0\ 0, 1].$$

And, lastly, let us make

(13.57) $$x_2 = 0, \quad x_3 = 0, \quad x_1 = 0, \quad x_4 = 0.$$

This is also a solution,

(13.58) $$[0, 0, 0, 0].$$

It is not possible to find any more solutions since we have successively evaluated, without omission or repetition,

the subset of solutions for which $x_2 = 1$,
the subset of solutions for which $x_2 = 0,\ x_3 = 1$,
the subset of solutions for which $x_2 = 0,\ x_3 = 0,\ x_1 = 1$,
the subset of solutions for which $x_2 = 0,\ x_3 = 0,\ x_1 = 0,\ x_4 = 1$,
the subset of solutions for which $x_2 = 0,\ x_3 = 0,\ x_1 = 0,\ x_4 = 0$.

The union of these five subsets gives the set of solutions.

The manner in which we constructed the arborescence was purely arbitrary, and any other progression could have been chosen. In practice it is not always necessary to calculate the solutions for an arborescence with complete branches in order to obtain all the solutions; when a certain point has been reached it can often be shown that it is not necessary to proceed further.

Second Example

Let us consider the system of five Boolean equations with seven variables:

$$(13.59)\qquad \begin{aligned} &(1)\quad a+b = \bar{c}.\bar{d}+\bar{e}.\bar{f}.\bar{g},\\ &(2)\quad b+d = \bar{a}.\bar{c}.\bar{e},\\ &(3)\quad e = \bar{a}.\bar{b}.\bar{c}+\bar{f},\\ &(4)\quad f = \bar{b}.\bar{c}.\bar{d}+\bar{c}.\bar{g},\\ &(5)\quad g = \bar{a}.\bar{b}.\bar{c}.\bar{d}+\bar{e}. \end{aligned}$$

If we take an inventory of the variables and their complements we find that they occur as follows: c six times, b five times, a and e four times, f and g three times. Let us decide to begin with c.

a. $c = 1$. Then system (13.59) becomes

$$(13.60)\qquad \begin{aligned} &(1)\quad a+b = \bar{e}.\bar{f}.\bar{g},\\ &(2)\quad b+d = 0,\\ &(3)\quad e = \bar{f},\\ &(4)\quad f = 0,\\ &(5)\quad g = \bar{e}. \end{aligned}$$

By considering (2) we find that $(c = 1) \Rightarrow (b = 0)$ and $(d = 0)$. By considering (4) we see that $(x = 1) \Rightarrow (f = 0)$. This result, when substituted in (3) of (13.60), shows that $(c = 1) \Rightarrow (e = 1)$. This new result substituted in (5) gives

$(c = 1) \Rightarrow (g = 0)$. All these results when introduced into (1) give $(c = 1) \Rightarrow (a = 0)$. Hence we have obtained a solution,

$$[a, b, c, d, e, f, g] = [0, 0, 1, 0, 1, 0, 0]. \tag{13.61}$$

b. $c = 0$, $b = 1$. The system becomes

$$\begin{aligned} &(1) \quad 1 = \bar{d} + \bar{e}.\bar{f}.\bar{g}, \\ &(2) \quad 1 = \bar{a}.\bar{e}, \\ &(3) \quad e = \bar{f}, \\ &(4) \quad f = \bar{g}, \\ &(5) \quad g = \bar{e}. \end{aligned} \tag{13.62}$$

Equation (2) gives $a = 0$, $e = 0$; from this, in accordance with (3), we have $f = 1$, and in accordance with (4) we have $g = 0$ which would give $e = 1$. Thus (3) and (5) are incompatible for $c = 0$ and $d = 1$. Hence at this vertex of the branching there is no solution.

c. $c = 0$, $b = 0$, $e = 1$. We have

$$\begin{aligned} &(1) \quad a = \bar{d}, \\ &(2) \quad d = 0, \\ &(3) \quad 1 = \bar{a} + \bar{f}, \\ &(4) \quad f = \bar{d} + \bar{g}, \\ &(5) \quad g = \bar{a}.\bar{b}. \end{aligned} \tag{13.63}$$

We find successively that from (2) $d = 1$, from (5) $g = 0$, from (4) $f = 1$. But by substituting $\bar{a} = 0$ and $\bar{f} = 0$ in (3) we produce an impossibility, so that there is no solution.

d. $c = 0$, $b = 0$, $c = 0$, $a = 1$. We have

$$\begin{aligned} &(1) \quad 1 = \bar{d} + \bar{f}.\bar{g}, \\ &(2) \quad d = 0, \\ &(3) \quad 0 = \bar{f}, \\ &(4) \quad f = \bar{d} + \bar{g}, \\ &(5) \quad g = 1. \end{aligned} \tag{13.64}$$

We find successively that from (2) $d = 0$, from (3) $f = 1$, from (5) $g = 1$. If we now substitute these values in (1) and (4) we find that these equations are verified. Hence we have a solution,

$$[a,b,c,d,e,f,g] = [1,0,0,0,0,1,1]. \tag{13.65}$$

e. $c=0$, $b=0$, $e=0$, $a=0$, $d=1$. We have

$$\begin{aligned} &(1)\quad 0 = \bar{f}.\bar{g},\\ &(2)\quad 1 = 1,\\ &(3)\quad 0 = 1 \quad \text{impossible}. \end{aligned} \tag{13.66}$$

And it is useless to proceed further since relation (3) cannot be satisfied whatever the value of d.

Hence there are only two solutions,

$$[a,b,c,d,e,f,g] = [0,0,1,0,1,0,0], \tag{13.67}$$

$$[a,b,c,d,e,f,g] = [1,0,0,0,0,1,1]. \tag{13.68}$$

Important Observation

It is advisable to avoid complete enumeration and to use sifting procedures (sequential or arborescent elimination) since the number of cases to be examined increases exponentially as the powers of 2 with the number of variables. Let us observe a few revealing figures,

$$2^{10} = 1024 = 1.024\times 10^3$$

$$2^{100} \simeq 1.2677\times 10^{30} \qquad \text{a number of 31 digits!}$$

$$2^{1000} \simeq 1.0716\times 10^{300} \qquad \text{a number of 301 digits!}$$

These show the necessity of avoiding complete combinatorial enumeration.

The branching method can be employed for inequations, since these can be transformed into equations by means of properties (13.29), (13.30), (13.43), or (13.44). Let us take a brief look at the procedure, using example (13.7).

$$x_1 x_2 \bar{x}_3 + \bar{x}_1 x_2 \leqslant x_1 \bar{x}_2 x_3. \tag{13.69}$$

By making use of (13.29), we obtain

$$(x_1 x_2 \bar{x}_3 + \bar{x}_1 x_2 \leqslant x_1 \bar{x}_2 x_3) \Leftrightarrow (x_1 x_2 \bar{x}_3 + \bar{x}_1 x_2).\overline{(x_1 \bar{x}_2 x_3)} = 0. \tag{13.70}$$

The right member can be simplified:

$$\begin{aligned} &(x_1 x_2 \bar{x}_3 + \bar{x}_1 x_2).\overline{(x_1 \bar{x}_2 x_3)}\\ &\quad = (x_1 x_2 \bar{x}_3 + \bar{x}_1 x_2).(\bar{x}_1 + x_2 + \bar{x}_3) = x_2.(\bar{x}_1 + \bar{x}_3) = 0. \end{aligned} \tag{13.71}$$

To solve (13.71) we now need only employ the branching method, but for this very simple equation the five solutions (13.9) can be found by inspection.

5. Method of Families of Solutions

This method consists of two stages.

1. Reduce the equation, inequation, the system of equations and/or of inequations to a single equation of the form

$$f(x_1, x_2, \dots, x_n) = 1, \tag{13.72}$$

and express f in a disjunctive form that is not necessarily canonical or sole.

$$\varphi_1 \dotplus \varphi_2 \dotplus \dots \dotplus \varphi_r = 1. \tag{13.73}$$

2. Consider every function φ_i, $i = 1, 2, \dots, r$. The condition $\varphi = 1$ provides a family F_i of solutions and their set is given by

$$\mathsf{F} = \mathsf{F}_1 \cup \mathsf{F}_2 \cup \dots \cup \mathsf{F}_r. \tag{13.74}$$

Let us consider an example. Let

$$x_2\bar{x}_4 \dotplus x_1x_2x_5 \dotplus \bar{x}_3\bar{x}_4 \dotplus \bar{x}_2\bar{x}_3 \dotplus \bar{x}_3x_5 \dotplus \bar{x}_1\bar{x}_4\bar{x}_5 = 0. \tag{13.75}$$

By taking the left and right complement, it follows that

(13.76)

$$(\bar{x}_2 \dotplus x_4)(\bar{x}_1 \dotplus \bar{x}_2 \dotplus \bar{x}_5)(x_3 \dotplus x_4)(x_2 \dotplus x_3)(x_3 \dotplus \bar{x}_5)(x_1 \dotplus x_4 \dotplus x_5) = 1.$$

And after the multiplications, cancellations, and absorptions, we have

$$x_1\bar{x}_2x_3 \dotplus \bar{x}_2x_3x_4 \dotplus \bar{x}_2x_3x_5 \dotplus \bar{x}_1x_3x_4 \dotplus x_2x_4\bar{x}_5 \dotplus x_3x_4\bar{x}_5 = 1. \tag{13.77}$$

From this we obtain

$$\begin{aligned} &\varphi_1 = x_1\bar{x}_2x_3, \quad \varphi_2 = \bar{x}_2x_3x_4, \quad \varphi_3 = \bar{x}_2x_3x_5, \\ &\varphi_4 = \bar{x}_1x_3x_4, \quad \varphi_5 = x_2x_4\bar{x}_5, \quad \varphi_6 = x_3x_4\bar{x}_5. \end{aligned} \tag{13.78}$$

(13.79)

For φ_1, we have : $\mathsf{F}_1 = \{[x_1, x_2, x_3, x_4, x_5] | x_1 = 1, x_2 = 0, x_3 = 1\}$,

For φ_2, we have : $\mathsf{F}_2 = \{[x_1, x_2, x_3, x_4, x_5] | x_2 = 0, x_3 = 1, x_4 = 1\}$,

For φ_3, we have : $\mathsf{F}_3 = \{[x_1, x_2, x_3, x_4, x_5] | x_2 = 0 \ x_3 = 1, x_5 = 1\}$,

For φ_4, we have : $\mathsf{F}_4 = \{[x_1, x_2, x_3, x_4, x_5] | x_1 = 0, x_3 = 1, x_4 = 1\}$,

For φ_5, we have : $\mathsf{F}_5 = \{[x_1, x_2, x_3, x_4, x_5] | x_2 = 1, x_4 = 1, x_5 = 0\}$,

For φ_6, we have : $\mathsf{F}_6 = \{[x_1, x_2, x_3, x_4, x_5] | x_3 = 1, x_4 = 1, x_5 = 0\}$.

We then construct a table:

(13.80)

	x_1	x_2	x_3	x_4	x_5
F_1	1	0	1	x_4	x_5
F_2	x_1	0	1	1	x_5
F_3	x_1	0	1	x_4	1
F_4	0	x_2	1	1	x_5
F_5	x_1	1	x_3	1	0
F_6	x_1	x_2	1	1	0

And from this table, by assigning to x_i ($i = 1, 2, 3, 4, 5$) the values of 0 and 1, it follows that

(13.81)

	x_1	x_2	x_3	x_4	x_5
F_1	1	0	1	0	0
	1	0	1	0	1
	1	0	1	1	0
	1	0	1	1	1
F_2	0	0	1	1	0
	0	0	1	1	1
	1	0	1	1	0
	1	0	1	1	1
F_3	0	0	1	0	1
	0	0	1	1	1
	1	0	1	0	1
	1	0	1	1	1
F_4	0	0	1	1	0
	0	0	1	1	1
	0	1	1	1	0
	0	1	1	1	1
F_5	0	1	0	1	0
	0	1	1	1	0
	1	1	0	1	0
	1	1	1	1	0
F_6	0	0	1	1	0
	0	1	1	1	0
	1	0	1	1	0
	1	1	1	1	0

By effecting the union of these families we obtain the table of solutions:

(13.82)

	x_1	x_2	x_3	x_4	x_5
F	1	0	1	0	0
	1	0	1	0	1
	1	0	1	1	0
	1	0	1	1	1
	0	0	1	1	0
	0	0	1	1	1
	0	1	1	1	0
	0	1	1	1	1
	0	1	0	1	0
	1	1	0	1	0
	1	1	1	1	0
	0	0	1	0	1

6. Pseudo-Boolean Linear Equations[1]

Given an equation

(13.83) $$A_1x_1 + B_1\bar{x}_1 + A_2x_2 + B_2\bar{x}_2 + \dots + A_nx_n + B_n\bar{x}_n = K$$

where

$$x_i \in \{0,1\}, \quad i = 1, 2, \dots, n, \qquad A_i, B_i \in \mathbf{R}, \quad i = 1, 2, \dots, n,$$
$$\text{and} \quad K \in \mathbf{R}.$$

We assume $A_i \neq B_i$, $i = 1, 2, \dots, n$, otherwise $A_ix_i + B_i\bar{x}_i = A_i(x_i + \bar{x}_i) = A_i$, where the plus signs represent common algebra. An equation such as (13.83) is known as a pseudo-Boolean linear equation.

To solve such an equation, we proceed as follows:

Let us first assume

(13.84) $$y_i = x_i \quad \text{if} \quad A_i > B_i \qquad \text{and} \qquad y_i = \bar{x}_i \quad \text{if} \quad B_i > A_i.$$

[1] This subsection, like the preceding one, has been inspired by the work of P. Hammer and S. Rudeanu [K14].

From this we obtain

$$A_i x_i + B_i \bar{x}_i = |A_i - B_i|\ y_i + (A_i \wedge B_i), \qquad i = 1, 2, \dots, n, \tag{13.85}$$

where $|\alpha|$ indicates the absolute value of α and $\alpha \wedge \beta$ means the minimum of α and β.

Let us assume

$$C_i = |A_i - B_i|, \tag{13.86}$$

then (13.83) can be expressed as

$$\sum_{i=1}^{n} A_i x_i + B_i \bar{x}_i = \sum_{i=1}^{n} C_i y_i + \sum_{i=1}^{n} (A_i \wedge B_i) = K. \tag{13.87}$$

Let us assume further that

$$D = K - \sum_{i=1}^{n} (A_i \wedge B_i), \tag{13.88}$$

then (13.83) will become

$$C_1 y_1 + C_2 y_2 + \dots + C_n y_n = D. \tag{13.89}$$

Now let us arrange the C_i in their total natural order, which gives the C'_j:

$$C'_1 \geqslant C'_2 \geqslant \dots \geqslant C'_n \tag{13.90}$$

and let us make variables u_j correspond to the y_i. This finally gives

$$C'_1 u_1 + C'_2 u_2 + \dots + C'_n u_n = D. \tag{13.91}$$

It can be proved[1] that eight distinct cases can emerge as follows:

(13.92) (1) $D < 0$. There is no solution.

(13.93) (2) $D = 0$. The sole solution is $u_1 = u_2 = \dots = u_n = 0$.

(13.94) (3) $D > 0$, with $C'_1 \geqslant C'_2 \geqslant \dots \geqslant C'_p > D \geqslant C'_{p+1} \geqslant \dots > C'_n$.

The solutions (if any) satisfy

$$u_1 = u_2 = \cdots = u_p = 0 \quad \text{and} \quad \sum_{j=p+1}^{n} C'_j . u_j = D.$$

(13.95) (4) $D > 0$ with $C'_1 = C'_2 = \dots = C'_p = D \geqslant C'_{p+1} \geqslant \dots \geqslant C'_n$.

Then, (a) for every $k = 1, 2, \dots, p : u_k = 1$ and $u_1 = \dots = u_{k-1} = u_{k+1} = \dots = u_n = 0$ is a solution; (b) the other solutions (if any) satisfy

$$u_1 = \dots = u_p = 0 \quad \text{and} \quad \sum_{j=p+1}^{n} C'_j u_j = D.$$

[1] See [K14], page 50 of the English edition of this work or page 59 of the French translation.

(13.96) (5) $D > 0, \quad C_i' < D, \quad i = 1, 2, \ldots, n \quad \text{and} \quad \sum_{i=1}^{n} C_i' < D.$

No solution.

(13.97) (6) $D > 0, \quad C_i' < D, \quad i = 1, 2, \ldots, n \quad \text{and} \quad \sum_{i=1}^{n} C_i' = D.$

The sole solution is $u_1 = u_2 = \ldots = u_n = 1.$

(13.98) (7) $D > 0, \quad C_i' < D, \quad i = 1, 2, \ldots, n \quad \text{and} \quad \sum_{i=1}^{n} C_i' > D$

$\text{and} \quad \sum_{j=2}^{n} C_j' < D.$

The solutions (if any) satisfy

$$u_1 = 1 \quad \text{and} \quad \sum_{j=2}^{n} C_j' u_j = D - C_1'.$$

(13.99) (8) $D > 0, \quad C_i' < D, \quad i = 1, 2, \ldots, n \quad \text{and} \quad \sum_{i=1}^{n} C_i' > D$

$\text{and} \quad \sum_{j=2}^{n} C_j' \geqslant D.$

The solutions (if any) satisfy

$$\begin{aligned} &\text{given } u_1 = 1 \quad \text{and} \quad \sum_{j=2}^{n} C_j' u_j = D - C_1', \\ &\text{given } u_1 = 0 \quad \text{and} \quad \sum_{j=2}^{n} C_j' u_j = D. \end{aligned} \tag{13.100}$$

To enumerate without omission or repetition all the solutions of (13.89) and then those of (13.83) we shall proceed by the method of forks or branchings. One example will suffice to demonstrate the procedure. Let

$$3x_1 + 2\bar{x}_1 + 7x_2 - \bar{x}_2 + 4x_3 + 8\bar{x}_4 = 10. \tag{13.101}$$

Let us first carry out the conversion given in (13.86) and (13.88).

$$\begin{aligned} C_1 &= |3-2| = 1, \qquad C_2 = |7+1| = 8, \\ C_3 &= |4-0| = 4, \qquad C_4 = |0-8| = 8. \end{aligned} \tag{13.102}$$

$$\begin{aligned} A_1 \wedge B_1 &= 3 \wedge 2 = 2, \qquad A_2 \wedge B_2 = 7 \wedge (-1) = -1, \\ A_3 \wedge B_3 &= 4 \wedge 0 = 0, \qquad A_4 \wedge B_4 = 0 \wedge 8 = 0. \end{aligned} \tag{13.103}$$

Then (13.101) becomes

$$y_1 + 8y_2 + 4y_3 + 8y_4 = 10 - (2 - 1 + 0 + 0) = 9. \tag{13.104}$$

Or again, by assuming

(13.105)

$$u_1 = y_2 = x_2\,, \quad u_2 = y_4 = \bar{x}_4\,, \quad u_3 = y_3 = x_3\,, \quad u_4 = y_1 = x_1\,,$$

(13.106) $$8u_1+8u_2+4u_3+u_4 = 9\,.$$

We shall solve (13.106) by considering which case will provide the conditions laid down in (13.92)–(13.99).

We find ourselves in case (8), that is to say (13.99), and we have

$$D = 9>0\,,\ \ C_1' = 8<D\,,\ \ C_2' = 8<D\,,\ \ C_3' = 4<D\,,\ \ C_4' = 1<D\,.$$

(13.107)

$$\sum_{i=1}^{4} C_i' = 8+8+4+1 = 21>D\,, \qquad \sum_{j=2}^{4} C_j' = 8+4+1 = 13>D\,.$$

Hence we shall state,

(13.108) $$u_1 = 1\,,$$

whence

(13.109) $$8u_2+4u_3+u_4 = 1\,.$$

Or

(13.110) $$u_1 = 0\,,$$

whence

(13.111) $$8u_2+4u_3+u_4 = 9\,.$$

The successive branchings will be shown in Fig. 13.2.

Let us now consider (13.109) where we are in case (3), namely (13.94); indeed,

(13.112) $$D' = 1>0\,, \qquad C_2' > C_3' > D' = C_4'\,.$$

Hence we shall state,

(13.113) $$u_2 = u_3 = 0 \quad \text{and} \quad u_4 = 1\,.$$

Let us now consider (13.111) where we are in case (7), namely (13.98); indeed,

(13.114)

$$D'' = 9 > 0\,, \qquad C_2' < D''\,, \qquad C_3' < D''\,, \qquad C_4' < D''\,,$$

$$\sum_{i=2}^{n} C_i' = 8+4+1 = 13 < D''\,,$$

$$\sum_{i=3}^{4} C_i' = 4+1 = 5 < D''\,.$$

Hence we shall state,

(13.115) $$u_2 = 1$$

and

(13.116) $$4u_3 + u_4 = 1\,.$$

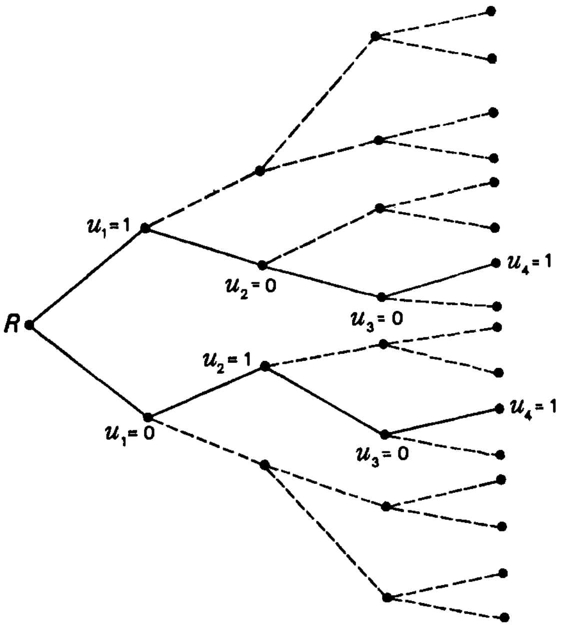


FIG. 13.2

Let us consider (13.116) where we are in case (3), namely (12.94), and indeed

(13.117) $$D''' = 4 > 0\,, \qquad C_3' > D''' = C_4'\,.$$

Hence we can state,

(13.118) $$u_3 = 0 \quad \text{and} \quad u_4 = 1\,.$$

We have finally found two solutions,

(13.119) $$[u_1, u_2, u_3, u_4] = [1,0,0,1]\,,$$

and

(13.120) $$[u_1, u_2, u_3, u_4] = [0,1,0,1]\,.$$

Passing to the initial variables x_i, $i = 1, 2, 3, 4$, and making use of (13.105), we obtain as a solution of (13.101)

(13.121) $$[x_1, x_2, x_3, x_4] = [1, 1, 0, 1],$$

and

(13.122) $$[x_1, x_2, x_3, x_4] = [1, 0, 0, 0].$$

It must be observed that in such a simple example the solutions could have been obtained by inspection, but the procedure becomes useful and indeed indispensable, in the absence of a better one, as soon as the number of variables increases. The corresponding table of enumeration is given in (13.123).

(13.123)

x_1	x_2	x_3	x_4	$3x_1$	$2\bar{x}_1$	$7x_2$	$-\bar{x}_2$	$4x_3$	$8\bar{x}_4$	Total	
0	0	0	0	0	2	0	-1	0	8	9	
0	0	0	1	0	2	0	-1	0	0	1	
0	0	1	0	0	2	0	-1	4	8	13	
0	0	1	1	0	2	0	-1	4	0	5	
0	1	0	0	0	2	7	0	0	8	17	
0	1	0	1	0	2	7	0	0	0	9	
0	1	1	0	0	2	7	0	4	8	21	
0	1	1	1	0	2	7	0	4	0	13	
1	0	0	0	3	0	0	-1	0	8	10	←
1	0	0	1	3	0	0	-1	0	0	2	
1	0	1	0	3	0	0	-1	4	8	14	
1	0	1	1	3	0	0	-1	4	0	6	
1	1	0	0	3	0	7	0	0	8	18	
1	1	0	1	3	0	7	0	0	0	10	←
1	1	1	0	3	0	7	0	4	8	22	
1	1	1	1	3	0	7	0	4	0	14	

Space is lacking in this work to treat pseudo-Boolean linear inequalities or, in a more general way, systems of Boolean linear equations and/or inequations, but the reader is referred to the work by Hammer and Rudéanu [K14], so often mentioned and utilized here, in which all the appropriate methods are explained and developed with remarkable clarity and detail.

Section 14. **Mathematical Properties of Programming with Integers**

1. General Observations

In this and the following sections we shall pursue two objectives. First we shall trace the historical development of the methods (the *dual-simplex* method, Dantzig–Manne's and Gomory's procedures, asymptomatic programming, and so on). Secondly we shall endeavor to improve the readers' knowledge of the geometry of polyhedrons, linear programming, and group theory, so that they will be better able to comprehend the more fundamental aspects of the methods.

Nevertheless the knowledge acquired by the readers of the first two volumes of this work, with respect to linear programming in Volume 1 and dynamic programming in Volume 2 should enable them to understand what is discussed in the present and subsequent sections. It is this instructional purpose that has led us, for instance, to modify Gomory's explanation of asymptotic programming since it requires elements of modern algebra that would be new to some of our older readers (such as the concepts of isomorphism, homomorphism, and the group), and so we reduce it to Smith's less abstract form presented in this chapter. We believe that what of a general nature will be lost by this change will be made up for by the gain in simplicity.

The methods for solving integer programs by direct search have been given in the first part, since they required only an elementary knowledge of mathematics. Here we shall concentrate with strict formality on the methods needed to produce *cuts*.

For lack of space a third group of recent methods, that of the *cut and search*, to which the main contributors were Glover [K39] and Balas [K27], has not been included, not because less importance was attributed to them but because it would have been necessary to extract the essential elements of instruction contained in them. Moreover, no sooner were the proofs of this work corrected, than new methods appeared. Let us recall that the volumes in this work are not intended as treatises but as a means of acquiring knowledge and practical methods for readers absorbed in economic life.

Having first made clear the practical importance of problems of programming in a cone, and having referred to integer programs, we shall devote Sections 16–22 to original material concerned with the theory directly involved.

To be sure, some of the more difficult passages will demand a considerable mental effort on the part of our readers, but any acquisition of knowledge, even if facilitated by preliminary instruction, requires this. Our aim has been to reduce this effort as much as possible.

2. Convex Subsets and Polyhedrons

In this section we shall consider the set product $\mathbf{R}''$ having the well-known properties of a vectorial space[1]:

(14.1) $$\mathbf{R}^n = \mathbf{R} \times \mathbf{R} \times \ldots \times \mathbf{R}$$

where $\mathbf{R}$ is the set of real numbers.

A *convex subset* $\mathbf{X}$ of $\mathbf{R}''$ is a subset such that, if

(14.2) $$[x^{(1)}] = \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ \vdots \\ x_n^{(1)} \end{bmatrix}$$

and

(14.3) $$[x^{(2)}] = \begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \\ \vdots \\ x_n^{(2)} \end{bmatrix}$$

are any two elements of $\mathbf{X}$, then every *element* or *point* such that

(14.4) $$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \lambda \begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ \vdots \\ x_n^{(1)} \end{bmatrix} + (1-\lambda) \begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \\ \vdots \\ x_n^{(2)} \end{bmatrix},$$

that can also be expressed as

(14.5) $$[x] = \lambda[x^{(1)}] + (1-\lambda)\,[x^{(2)}],$$

also belongs to $\mathbf{X}$ if $0 \leqslant \lambda \leqslant 1$.

We also state that if $[x]$ belongs to the segment that joins $[x^{(1)}]$ to $[x^{(2)}]$, it belongs to $\mathbf{X}$ if $\mathbf{X}$ is convex for any pair of elements $[x^{(1)}]$ and $[x^{(2)}]$ belonging to $\mathbf{X}$.

Let us first consider an example in $\mathbf{R}^2 = \mathbf{R} \times \mathbf{R}$ (Figs. 14.1 and 14.2). The subset $\mathbf{A} \subset \mathbf{R}^2$ shown in Fig. 14.1 is convex. Whichever pair of points $[x^{(1)}]$ and $[x^{(2)}]$ is chosen in $\mathbf{A}$, all the points of the segment connecting them belong to $\mathbf{A}$.

[1] See page 195.

[1] Let us recall that a vectorial space such as $\mathbf{R}^n$ is defined by the following properties. If

$$[x] = \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} \quad \text{and} \quad [y] = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_n \end{bmatrix}$$

are any two elements of $\mathbf{R}^n$, it is usual to refer to them as *vectors*. The following properties can then be verified:

(1)
$$[x]+[y] = \begin{bmatrix} x_1+y_1 \\ x_2+y_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n+y_n \end{bmatrix} = [x+y].$$

(2) If λ is a scalar : $\lambda \in \mathbf{R}$:

$$\lambda[x] = \begin{bmatrix} \lambda x_1 \\ \lambda x_2 \\ \cdot \\ \cdot \\ \cdot \\ \lambda x_n \end{bmatrix} = [\lambda x].$$

Mathematicians, however, give a much more general sense to the concept of vectorial space. We describe it for the benefit of readers with a more advanced knowledge of modern mathematics, adding the necessary explanations.

Let us consider a set $\mathbf{K}$ with a bodily structure that satisfies the two internal laws represented by $+$ and $\bullet$, namely $(\mathbf{K}, +, \bullet)$. On this set, called the *scalar body*, it is possible to perform operations similar to common addition and multiplication on the one hand and to subtraction and division on the other, such as we carry out for the set $\mathbf{R}$ of real numbers. A second set $\mathbf{V}$ provided with a structure having a commutative grouping for a law indicated by $*$, namely $(\mathbf{V}, *)$ is also considered. We say that $\mathbf{V}$ is a *vectorial space* if a law of external arrangement exists for $\mathbf{K}$ indicated by $\circ$, that is to say that by means of $\circ$ we arrange an element $a \in \mathbf{K}$ and an element $V \in \mathbf{V}$ that can be expressed

$$a \circ V = U, \qquad \forall a \in \mathbf{K},\ \forall V,\ U \in \mathbf{V}.$$

This external law should verify the following axioms: $\forall V$, $U \in \mathbf{V}$, and $\forall a$, $b \in \mathbf{K}$,

(1) $(a * b) \circ V = (a \circ V) * (b \circ V)$,
(2) $a \circ (V * U) = (a \circ V) * (a \circ U)$,
(3) $a \circ (b \circ V) = (a \circ b) \circ V$,
(4) $e \circ V = V$, where e is the unit of $\mathbf{K} - 0$ for the law indicated by $\circ$ of $\mathbf{K}$ and 0 is the unit of $\mathbf{K}$ for the law $*$.

The elements of $\mathbf{V}$ are called *vectors*. It should be noted that, in practice, we use the same symbol for $+$ and $*$ on the one hand and for $\bullet$ and $\circ$ on the other, but this can lead to confusion for certain vectorial spaces.

In the special case of vectorial space considered in this and the following sections we take $\mathbf{K} = \mathbf{R}$ and $\mathbf{V} = \mathbf{R}^n$, $* = +$ and $\circ = \bullet$; this permits some simplification and allows us to define this special but most often used space without the need for being too axiomatic. Let us also observe that this vectorial space defined in $\mathbf{R}^n$ is termed an *affinity* if no metric is introduced.

For fuller details the reader should consult one of our references such as [K45].

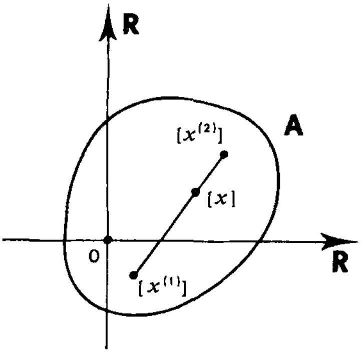


FIG. 14.1

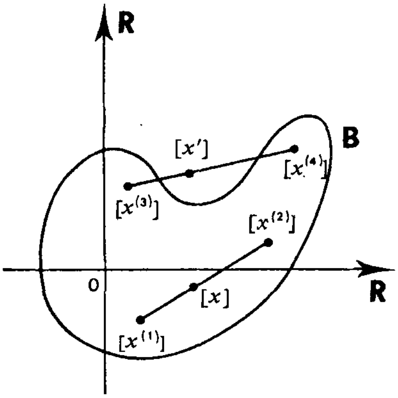


FIG. 14.2

This is no longer true in the example given in Fig. 14.2, where $[x^{(3)}]$ and $[x^{(4)}]$ clearly belong to **B**, but there is at least one point $[x']$ situated on the segment joining these two points of **B** that does not belong to **B**.

Let us consider another instructional example.

Let there be a straight line

$$(14.6) \qquad [x] = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \mu \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}, \qquad \mu \in \mathbf{R}, \quad \mu \geqslant 0,$$

which therefore passes through the origin,

$$[0] = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \text{and the point} \quad [a] = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix},$$

in such a way that point $[a]$ defines the direction of the straight line. Let us verify that if we consider two points $[x^{(1)}] = \mu_1 . [a]$ and $[x^{(2)}] = \mu_2 . [a]$ of this straight line, the points on the segment joining them belong to this straight line, and it is therefore a convex subset. This is expressed

$$\begin{aligned} [x] &= \lambda . [x^{(1)}] + (1-\lambda) . [x^{(2)}] \\ (14.7) \qquad &= \lambda \mu_1 . [a] + (1-\lambda)\, \mu_2 . [a] \\ &= (\lambda \mu_1 + (1-\lambda)\, \mu_2) . [a] \\ &= \nu . [a], \end{aligned}$$

where

(14.8) $$\nu = \lambda\mu_1 + (1-\lambda)\,\mu_2 \geqslant 0\,.$$

Let us consider another example.

Given

(14.9) $$[a] = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix}$$

and

(14.10) $$[x] = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix},$$

we have [1]

(14.11) $$[a]'_{1\times n}\cdot[x]_{n\times 1} = a_1x_1+a_2x_2+\ldots+a_nx_n\;.$$

Then

(14.12) $$[a]'_{1\times n}\cdot[x]_{n\times 1} = \beta\,,$$

where $\beta \in \mathbf{R}$, defines a plane (more accurately termed the hyperplane when it is not specified that $n = 3$). It is clear that such a plane is a convex subset demarcating two convex subsets,

(14.13) $$[a]'\,.\,[x] < \beta$$

and

(14.14) $$[a]'\,.\,[x] > \beta\,.$$

Let us verify that (14.13), for example, is a convex subset.

Given

(14.15) $$[x^{(1)}] \text{ and } [x^{(2)}] \in \{[x]\,|\,[a]'\,.\,[x] < \beta\}$$

and

(14.16) $$[x^{(3)}] = \lambda\,.\,[x^{(1)}] + (1-\lambda)\,.\,[x^{(2)}]\,, \qquad 0 \leqslant \lambda \leqslant 1\,.$$

[1] As in the first two volumes, we use the notation $[a]'$ to indicate the transpose of $[a]$. Also, if a matrix has m lines and n columns, and if this is useful, we represent it as $[a]_{m\times n}$.

It follows

(14.17)

$$[a]'.[x^{(3)}] = \lambda[a]'.[x^{(1)}] + (1-\lambda).[a]'.[x^{(2)}] < \lambda\beta + (1-\lambda)\beta,$$

that is to say,

$$(14.18) \qquad [a]'.[x^{(3)}] < \beta,$$

which proves that every $[x^{(3)}]$ such as (14.16) certainly belongs to the subset of $\mathbf{R}^n$ defined by (14.13).

Lemma 14.I

The intersection of several convex subsets of $\mathbf{R}^n$ is convex.

Proof

Let us consider two subsets $\mathbf{X}_1$ and $\mathbf{X}_2$ of $\mathbf{R}^n$ with $[x^{(1)}]$, $[x^{(2)}] \in \mathbf{X}_1 \cap \mathbf{X}_2$; then

$$(14.19) \qquad [x] = \lambda - [x^{(1)}] + (1-\lambda).[x^{(2)}]$$

belongs to $\mathbf{X}_1$ since $[x^{(1)}]$ and $[x^{(2)}]$ both belong to $\mathbf{X}_1$, which is convex. For similar reasons $[x]$ belongs to $\mathbf{X}_2$. Hence $[x]$ belongs to the intersection of $\mathbf{X}_1$ and $\mathbf{X}_2$.

This can at once be expressed in a general form for the intersection of r convex subsets of $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_r$ and $\mathbf{R}^n$, $r = 1, 2, 3, 4, \ldots$.

Corollary 14.II

The subset defined by the intersection of m half-spaces

$$(14.20) \qquad \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leqslant b_1, \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leqslant b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leqslant b_m, \end{array}$$

that can also be expressed as

$$(14.21) \qquad [A].[x] \leqslant [b],$$

where

$$(14.22) \qquad [A] = \begin{bmatrix} a_{11}\, a_{12} \cdots a_{1n} \\ a_{21}\, a_{22} \cdots a_{2n} \\ \ldots\ldots\ldots \\ a_{m1}\, a_{m2} \cdots a_{mn} \end{bmatrix},$$

$$[x] = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \tag{14.23}$$

$$[b] = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \tag{14.24}$$

is a convex subset. This results from Lemma 14.I. Each subset such as (14.20) is convex.

Convex Polyhedron

The m inequations of (14.20) can be expressed as[1]

$$\begin{aligned} &[A]_1 \cdot [x] \leqslant b_1, \\ &[A]_2 \cdot [x] \leqslant b_2, \\ &\dots\dots\dots\dots\dots \\ &[A]_m \cdot [x] \leqslant b_m, \end{aligned} \tag{14.25}$$

where

$$[A]_i = [a_{i1}\ a_{i2} \dots a_{in}]. \tag{14.26}$$

These define what is termed a *convex polyhedron.*

Figure 14.3 represents a convex polyhedron in $\mathbf{R}^2$.

Such a polyhedron can be reduced to a single element or even to no element of $\mathbf{R}^n$; thus $\varnothing$ is a convex polyhedron of $\mathbf{R}^n$.

3. Cones

We shall define what is meant by a cone and will afterward give some properties of cones. We shall discover that these properties play an important part in various questions of optimization.

$$[A] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \tag{14.27}$$

[1] We indicate the ith line of a matrix $[A]$ by $[A]_i$. It should not be confused with the notation $[A]_{m \times n}$ that indicates the number m of lines and n of columns.

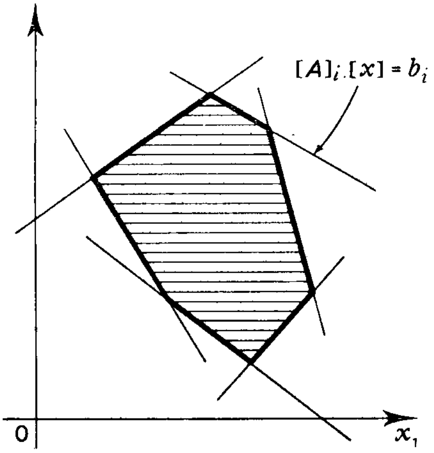


FIG. 14.3

and

(14.28) $$[b] = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

then the set of points $[x] \in \mathbf{R}^n$ such that

(14.29) $$\{[x] | [A] . [x] \leqslant [b]\},$$

is called a *convex polyhedral cone* (CPC) if there is a point $[x^{(0)}]$ such that

(14.30) $$[A] . [x^{(0)}] = [b] .$$

Figure 14.4 gives an example of a convex polyhedral cone in $\mathbf{R}^2$. Let us take another example, this time in $\mathbf{R}^3$. Let us take three half-spaces

(14.31) $$\begin{array}{ll} (1) & -8x_1 - 4x_2 - 3x_3 \leqslant -24, \\ (2) & -20x_1 + 12x_2 - 9x_3 \leqslant 0, \\ (3) & x_3 \leqslant 4. \end{array}$$

Or, in matrical form,

(14.32) $$\begin{bmatrix} -8 & -4 & -3 \\ -20 & 12 & -9 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \leqslant \begin{bmatrix} -24 \\ 0 \\ 4 \end{bmatrix} .$$

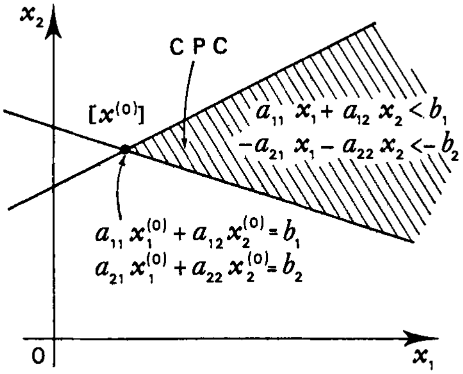

x_2
CPC
$[x^{(0)}]$
$a_{11}\, x_1 + a_{12}\, x_2 < b_1$
$-a_{21}\, x_1 - a_{22}\, x_2 <- b_2$
$a_{11}\, x_1^{(0)} + a_{12} x_2^{(0)} = b_1$
$a_{21}\, x_1^{(0)} + a_{22}\, x_2^{(0)} = b_2$
0
x_1

FIG. 14.4

We can verify that a point exists

$$(14.33) \qquad [x^{(0)}] = \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ x_3^{(0)} \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix},$$

that satisfies $[A]\,[x] = [b]$, namely,

$$(14.34) \qquad \begin{bmatrix} -8 & -4 & -3 \\ -20 & 12 & -9 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} -24 \\ 0 \\ 4 \end{bmatrix}.$$

In Fig. 14.5 the three straight lines that define the CPC corresponding to (14.32) are shown by heavy lines.

There are other relations besides (14.29) and (14.30) for defining a CPC, but they must be such as can be reduced to this form.

Ridge. Vertex. Edge. Ray

The point $[x^{(0)}]$ that satisfies (14.29) and (14.30) may or may not be the sole solution, and we propose to consider this important question.

We give the term *ridge* of the CPC to the set of points

$$(14.35) \qquad \{[x^{(0)}] \,|\, [A] . [x^{(0)}] = [b]\}.$$

By definition this set is not void since we are concerned with a CPC.

In particular, if the rank of matrix $[A]$ is equal to m, that is to say, possesses the order of $[A]$, the solution of the matrical equation

$$(14.36) \qquad [A] . [x^{(0)}] = [b]$$

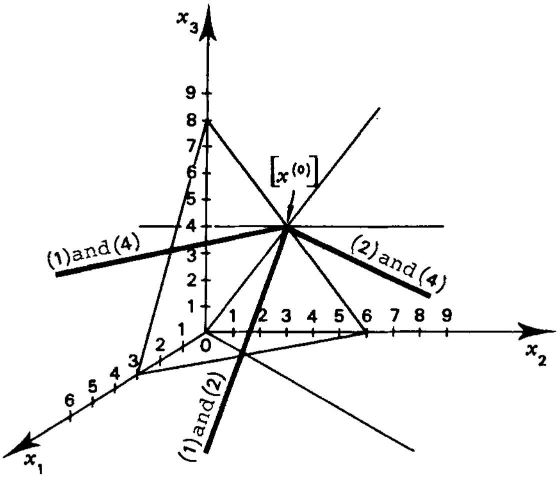


FIG. 14.5

is the sole one (arising from the well-known conditions laid down by Cramer). We shall then define $[x^{(0)}]$ by the name *vertex* of the CPC.

Figures (14.4) and (14.5) show examples where the ridges of the respective cones are vertices. Let us examine a case where the ridge cannot be reduced to a vertex.

In $\mathbf{R}^3$ let us consider the following example:

$$\text{(14.36a)}\qquad \begin{aligned} &(1)\quad 3x_1+3x_2+2x_3 \leqslant 18,\\ &(2)\quad 6x_1+2x_2+3x_3 \leqslant 18,\\ &(3)\quad -3x_1+x_2-x_3 \leqslant 0,\end{aligned}$$

that in its matrical form is

$$\text{(14.37)}\qquad \begin{bmatrix} 3 & 3 & 2\\ 6 & 2 & 3\\ -3 & 1 & -1\end{bmatrix} \cdot \begin{bmatrix} x_1\\ x_2\\ x_3\end{bmatrix} \leqslant \begin{bmatrix} 18\\ 18\\ 0\end{bmatrix}.$$

The determinant of the matrix of coefficients has as its value

$$\text{(14.38)}\qquad |A| = \begin{vmatrix} 3 & 3 & 2\\ 6 & 2 & 3\\ -3 & 1 & -1\end{vmatrix} = 3\begin{vmatrix} 2 & 3\\ 1 & -1\end{vmatrix} - 6\begin{vmatrix} 3 & 2\\ 1 & -1\end{vmatrix} - 3\begin{vmatrix} 3 & 2\\ 2 & 3\end{vmatrix}$$

$$= 3(-2-3)-6(-3-2)-3(9-4)$$

$$= -15+30-15 = 0.$$

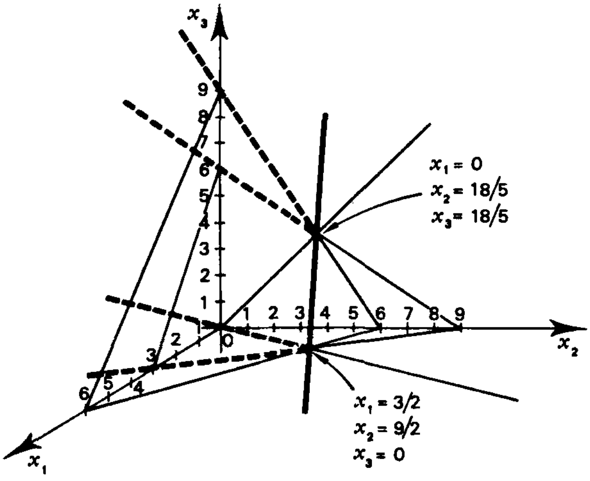


FIG. 14.6

Hence the rank of $[A]$ is less than 3 and is equal to 2. It can easily be verified that there is a point $[x^{(0)}]$ such that

$$(14.39) \qquad [A].[x^{(0)}] = [b],$$

but this is no longer the sole point. A straight line exists (see Fig. 14.6) that runs through the points

$$(14.40) \qquad [x^{(1)}] = \begin{bmatrix} 3/2 \\ 9/2 \\ 0 \end{bmatrix}$$

and

$$(14.41) \qquad [x^{(2)}] = \begin{bmatrix} 0 \\ 18/5 \\ 18/5 \end{bmatrix}$$

and that is a ridge of the CPC. This ridge is shown by a heavy line, whereas the broken lines have only been included in order to visualize the cone without delimiting it. The CPC is delimited by the planes

$$(14.42) \qquad \begin{aligned} 3x_1+3x_2+2x_3 &= 18, \\ 6x_1+2x_2+3x_3 &= 18, \end{aligned}$$

and the plane

$$(14.43) \qquad -3x_1+x_2-x_3 = 0$$

that passes through their intersection and the point of origin of the coordinates.

Let us now consider what is meant by *edge* and *ray* of a CPC. In fact these concepts are the same if the CPC is nondegenerate, that is to say, if A is of rank m; but these concepts usually differ where we are concerned with a polyhedron rather than with a cone.

The term *edge* of a CPC is used for the set of points of the CPC forming the intersection of $n-1$ of the n hyperplanes delimiting it. Thus in Fig. 14.5 the three half-lines in heavy type are the edges of the CPC shown in the figure. In this example (14.32) the matrix is of rank n, so that the intersection of any two planes gives a straight line passing through the vertex. But owing to the inequality between the left and the right members shown in (14.32) the set of points belonging to the cone is limited to a half-line.

The *ray* of a CPC is an edge that includes an infinite point. It is evident that, in the case of a nondegenerate CPC, all the edges are half-lines (we should really say *half-hyper lines*) passing through the vertex. All the points on them belong to the CPC, including the points in infinity. In the case of a CPC all the edges are rays.

The *edge of a convex polyhedron* is the set of points of the polyhedron forming the intersection of $n-1$ of the m hyperplanes delimiting it. Thus in Fig. 14.3, the segments of straight lines shown in heavy type are the edges of the convex polyhedron, in this case a polygon.

The term *ray* of a convex polyhedron is given to an edge that includes an infinite point. If the polyhedron includes at least one ray we say that it is *nonbounded.* We shall discover in Section 16 that a linear program containing, as the convex polyhedron of the constraints, a nonbounded polyhedron, can have, in certain cases, an economic function of infinite value.

We define the *direction of the ray* as a vector $[V]$ that has the same direction as the ray. This means that the points of the ray have the form

$$(14.44) \qquad [x]_{n\times 1} = [x^{(0)}]_{n\times 1} + \theta [V]_{n\times 1}, \qquad \theta \geqslant 0,$$

where $[x^{(0)}]$ is an extreme point[1] of the convex polyhedron and θ is a nonnegative scalar. Let us note that if the convex polyhedron is a CPC, $[x^{(0)}]$ is the extreme point of the cone, that is to say, its vertex.

Nondegenerate Convex Polyhedral Cone

The set of elements or points that satisfy

$$(14.45) \qquad [A].[x] \leqslant [b]$$

is called a *nondegenerate convex polyhedral cone*, if and only if, the columns of $[A]$ are linearly independent or, which amounts to the same, the rank of $[A]$ is equal to the number of columns or lines of $[A]$, that is to say, is of the order of $[A]$. In this case, as we have seen, the ridge of the cone is reduced to the vertex.

[1] This is defined in the following paragraph.

In the case of a nondegenerate CPC we can give the following explicit expression for the cone:

Given a CPC defined by

$$(14.46) \qquad [A]_{n\times n}\cdot[x]_{n\times 1} \leqslant [b]_{n\times 1}\,.$$

Let us transform this matrical inequation into a matrical equation by introducing a *deviation vector*:

$$(14.47) \qquad [u]_{n\times 1} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}, \qquad u_i \geqslant 0, \quad i = 1, 2, \dots, n,$$

in (14.46) which becomes

$$(14.48) \qquad [A]_{n\times n}\cdot[x]_{n\times 1} + [u]_{n\times 1} = [b]_{n\times 1}\,.$$

Since $[A]_{n\times n}$ is by hypothesis a regular matrix, we have

$$(14.49) \qquad [x]_{n\times 1} = [A]^{-1}_{n\times n}\cdot[b]_{n\times 1} - [A]^{-1}_{n\times n}\cdot[u]_{n\times 1}\,.$$

If we make

$$(14.50) \qquad [u]_{n\times 1} = [0]_{n\times 1}, \qquad \text{with } u_i = 0, \quad i = 1, 2, \dots, n,$$

in (14.49), we obtain the vertex of the cone.

Let us consider a very simple example. Let

$$(14.51) \qquad \begin{bmatrix} -1 & 1 \\ 1 & 2 \end{bmatrix}\cdot\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leqslant \begin{bmatrix} 3 \\ 18 \end{bmatrix}.$$

be a nondegenerate CPC (Fig. 14.7).

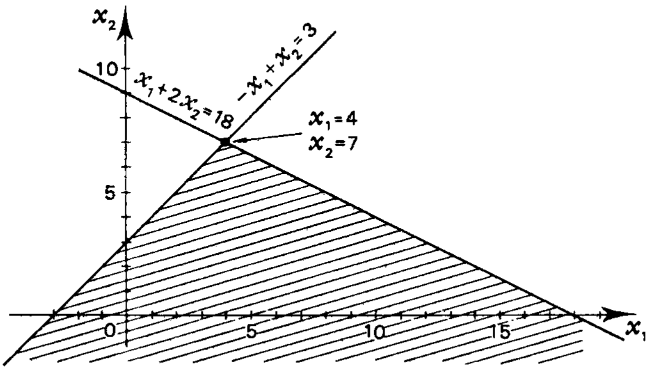


FIG. 14.7

Since

$$[A] = \begin{bmatrix} -1 & 1 \\ 1 & 2 \end{bmatrix}, \tag{14.52}$$

we have

$$[A]^{-1} = \begin{bmatrix} -2/3 & 1/3 \\ 1/3 & 1/3 \end{bmatrix}. \tag{14.53}$$

We first have

$$[x^{(0)}] = \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \end{bmatrix} = [A]^{-1}.[b] = \begin{bmatrix} -2/3 & 1/3 \\ 1/3 & 1/3 \end{bmatrix}.\begin{bmatrix} 3 \\ 18 \end{bmatrix} = \begin{bmatrix} 4 \\ 7 \end{bmatrix}. \tag{14.54}$$

Hence

$$x_1^{(0)} = 4 \quad \text{and} \quad x_2^{(0)} = 7. \tag{14.55}$$

Let us express the relation corresponding to (14.49)

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -2/3 & 1/3 \\ 1/3 & 1/3 \end{bmatrix}.\begin{bmatrix} 3 \\ 18 \end{bmatrix} - \begin{bmatrix} -2/3 & 1/3 \\ 1/3 & 1/3 \end{bmatrix}.\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}. \tag{14.56}$$

That is

$$x_1 = 4+\tfrac{2}{3}u_1-\tfrac{1}{3}u_2, \tag{14.57}$$

$$x_2 = 7-\tfrac{1}{3}u_1-\tfrac{1}{3}u_2. \tag{14.58}$$

$$u_1, u_2 \geqslant 0.$$

If we make, for example, $u_2 = 0$ in (14.57) and (14.58), it follows that

$$x_1=4+\tfrac{2}{3}u_1, \tag{14.59}$$

$$x_2 = 7-\tfrac{1}{3}u_1, \tag{14.60}$$

$$u_1 \geqslant 0.$$

This is the parametral equation in relation to u_1 of the half-line of $x_1+2x_2 = 18$ beginning with $x_1 = 4$, $x_2 = 7$ that delimits the cone; such a half-line is a *ray* of the cone. In the same way, if we make $u_1 = 0$ in (14.57) and (14.58) we find

$$x_1 = 4-\tfrac{1}{3}u_2, \tag{14.61}$$

$$x_2 = 7-\tfrac{1}{3}u_2, \tag{14.62}$$

$$u_2 \geqslant 0.$$

We then find the parametral equations of the other half-line which is also a ray of the cone.

Relations (14.57) and (14.58) constitute the parametral expressions of the nondegerate CPC.

4. Extreme Points of a Convex Subset

An extreme point of a convex subset $\mathbf{X} \subset \mathbf{R}^n$ is a point $[x^*]$ such that

(14.69)[1]
$$\forall [x^{(1)}], [x^{(2)}] \in \mathbf{X}, \quad \forall \lambda \in [0,1] :$$
$$[x^*] = \lambda [x^{(1)}] + (1-\lambda)\,[x^{(2)}] \Rightarrow [x^*] = [x^{(1)}] = [x^{(2)}].$$

This definition in general and applies equally to all convex subsets and to polyhedral subsets.

Figures 14.8 and 14.9 represent convex subsets in which the points indicated by $[x^*]$ are extreme points

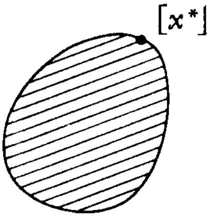


FIG. 14.8

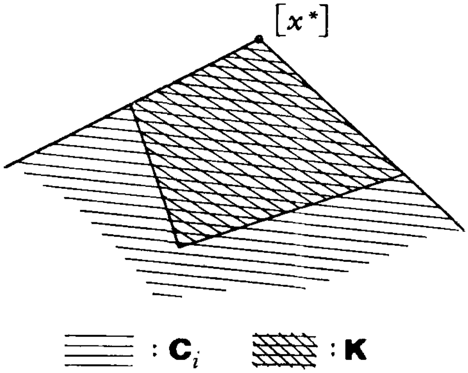


FIG. 14.9

Let us consider in particular a convex polyhedron $\mathbf{K} \subset \mathbf{R}^n$ defined by the matrical relation

(14.70) $$[A]_{m\times n} \cdot [x]_{n\times 1} \leqslant [b]_{m\times 1}$$

where $[A]_{m\times n}$ is a matrix $m \times n$ with $m \geqslant n$.

In choosing a submatrix $n \times n$ of $[A]$, namely $[B]$, we are defining a convex cone

(14.71) $$[B]_{n\times n} \cdot [x]_{n\times 1} \leqslant [b_B]_{n\times 1},$$

where $[b_n]$ is a matrical column taken from $[b]$ and corresponding to $[B]$.

We will assume that all the C_m^n (the number of combinations of m objects

[1] Equation numbers (14.63)–(14.68) omitted from the French edition.

$n \times n$) cones that can be defined in this manner are nondegenerate. Expressed differently, the C_m^n submatrices $n \times n$ are regular. Clearly, these cones C_i, $i = 1, 2, \ldots, C_m^n$, are obtained by n hyperplanes from among the m defined by (14.70) and each contains **K**, that is to say,

$$(14.72)\qquad \mathbf{C}_i \supset \mathbf{K}, \qquad i = 1, 2, \ldots, C_m^n.$$

For the theory of linear programming it is not necessary to assume that these C_m^n cones are nondegenerate, but we shall use this property here.

Let us show that the vertices of the cones C_i are extreme points of **K**. Indeed, these vertices give all solutions of the C_m^n equations that are assumed to be sole and distinct:

$$(14.73)\qquad [B]_{n\times n}\cdot[x]_{n\times 1} = [b_B]_{n\times 1}$$

taken from

$$(14.74)\qquad [A]_{m\times n}\cdot[x]_{n\times 1} = [b]_{m\times 1}\,.$$

Let us use a proof by absurdity.

If $[x^{(0)}]$ is a vertex of a cone C_i and is not an extreme point of **K** we produce an absurdity. Indeed, let there be two other points $[x^{(1)}] \neq [x^{(0)}]$ and $[x^{(2)}] \neq [x^{(0)}]$ such that

$$(14.75)\qquad [x^{(0)}] = \lambda[x^{(1)}] + (1-\lambda)\,[x^{(2)}], \qquad 0 < \lambda < 1\,.$$

(we have $\lambda \neq 0$ and $\lambda \neq 1$, otherwise we could have $[x^{(0)}] = [x^{(1)}]$ or $[x^{(0)}] = [x^{(2)}]$.) Then there exists at least one of the inequalities (14.71) that is strictly satisfied by $[x^{(1)}]$ since $[x^{(0)}]$, assumed to be the sole vertex, is the only one that satisfies all of them, that is,

$$(14.76)\qquad ([B]_j)_{1\times n}\cdot[x^{(1)}]_{n\times 1} < [b_{B_j}]_{1\times 1},$$

whereas

$$(14.77)\qquad [B]_{n\times n}\cdot[x^{(0)}]_{n\times 1} = [b_B]_{n\times 1}\,.$$

For the other point $[x^{(2)}]$, we have

$$(14.78)\qquad ([B]_j)_{1\times n}\cdot[x^{(2)}]_{n\times 1} \leqslant [b_{B_j}]_{1\times 1}\,.$$

By combining (14.76) and (14.78) and by taking into account that $\lambda > 0$ and $1-\lambda > 0$,

$$(14.79)\qquad \begin{aligned}&\lambda([B]_j)_{1\times n}\cdot[x^{(1)}]_{n\times 1} + (1-\lambda)\,[B_j]_{1\times n}\cdot[x^{(2)}]_{n\times 1}\\ &< \lambda[b_{B_j}]_{1\times 1} + (1-\lambda)\,[b_{B_j}]_{1\times 1} = [b_{B_j}]_{1\times 1},\end{aligned}$$

that is,

$$(14.80)\qquad ([B]_j)_{1\times n}\cdot[x^{(0)}]_{n\times 1} < [b_{B_j}]_{1\times 1},$$

which contradicts (14.77) characterizing $[x^{(0)}]$.

To illustrate this proof let us consider the example in Fig. 14.10.

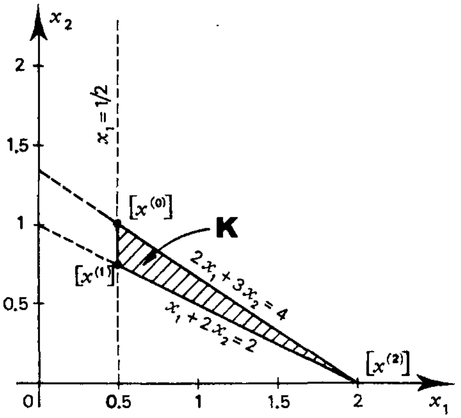


FIG. 14.10

Let there be a convex polyhedron in $\mathbf{R}^2$ defined by

$$(14.81)\qquad \begin{aligned} &(1)\quad -x_1 \leqslant -1/2, \\ &(2)\quad -x_1-2x_2 \leqslant -2, \\ &(3)\quad 2x_1+3x_2 \leqslant 4. \end{aligned}$$

Let us consider these inequations. By solving (1) and (2) we find that $[x^{(1)}] = [1/2\ 3/4]$. By solving (1) and (3) we find $[x^{(0)}] = [1/2\ 1]$. By solving (2) and (3) it follows that $[x^{(2)}] = [2\ 0]$. We see how the vertices of the three cones form the three extreme points of $\mathbf{K}$.

Important Observation

In practice there are generally less than C_m^u extreme points in a polyhedron $\mathbf{K}$ since, when we choose any n inequations and solve

$$(14.82)\qquad [B]_{n\times n}\cdot[x]_{n\times 1} = [b_B]_{n\times 1},$$

there is no guarantee that the $m-n$ remaining inequations will be satisfied by the solution of (14.82).

This equation is important because the optimum for certain problems of mathematical programming occurs at an extremity, and we could eventually enumerate all the extreme points of a convex polyhedron $\mathbf{K}$, evaluate the economic function for each, and then choose the optimal point (or points) for this function.

However, if C_n^m is very large, which is often the case, we are excluded from enumerating all the extremities even with the use of the most powerful computers. Certain methods exist that avoid the need for this total enumeration (see [K51]).

5. Geometrical Interpretation of Gauss–Jordan's Pivoting

If we return to example (14.81) and introduce three deviation variables u_1, u_2, and u_3 this system of equations becomes

$$\begin{array}{lll} (1) & -x_1+u_1 = -1/2, & \\ (2) & -x_1-2x_2+u_2 = -2, & u_1, u_2, u_3 \geqslant 0, \\ (3) & 2x_1+3x_2+u_3 = 4. & \end{array} \tag{14.83}$$

Let us examine Fig. 14.10. Vertex $[x^{(0)}]$ corresponds to $u_1 = u_3 = 0$, vertex $[x^{(1)}]$ to $u_1 = u_2 = 0$, and vertex $[x^{(2)}]$ to $u_2 = u_3 = 0$.

In (14.83) if we express, for example, x_1, x_2, and u_3 in relation to u_1 and u_2, it follows that

$$\begin{array}{lll} (1) & x_1 = \frac{1}{2}+u_1, & \\ (2) & x_2 = \dfrac{3}{4}-\dfrac{u_1}{2}+\dfrac{u_2}{2}, & u_1, u_2 \geqslant 0, \\ (3) & z_3 = \frac{3}{4}-2u_1-\frac{3}{2}u_2. & \end{array} \tag{14.84}$$

which, in its matrical form, is

$$\begin{bmatrix} x_1 \\ x_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1/2 \\ 3/4 \\ 3/4 \end{bmatrix} - \begin{bmatrix} -1 & 0 \\ 1/2 & -1/2 \\ 2 & 3/2 \end{bmatrix} . \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \qquad \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \geqslant \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{14.85}$$

When u_1 and u_2 assume their values in the interval $[0, \infty]$, the point $[x] = [x_1 . x_2]$ traverses an edge of the cone

$$\begin{array}{l} -x_1 \leqslant -1/2, \\ -x_1-2x_2 \leqslant 2, \end{array} \tag{14.86}$$

from $[x^{(1)}]$ to $[x^{(2)}]$.

Let us now consider the meaning of *Gauss–Jordan's pivoting method.* This consists in replacing one of the basic variables by one that is not in the basis. This is precisely the same method as we used for the simplex method in Volume 1, Sections 58 and 59. In the theory of matrical calculation it bears the above name.

Using example (14.83) we shall give another illustration of this method adapted to the needs of the present volume.

If we consider (14.85) we find that x_1, x_2, and u_3 belong to the basis, whereas u_1 and u_2 do not. Let us remove u_3 from the basis to make room for u_2; to do this let us use (3) in (14.84) where u_2 will be expressed as a function of u_1 and u_3, the result then being substituted in (1) and (3) of (14.84). This gives

$$(14.87) \qquad \begin{array}{ll} (1) & x_1 = \tfrac{1}{2}+u_1 , \\ (2) & x_2 = 1-\tfrac{2}{3}u_1-\tfrac{1}{3}u_3 , \qquad u_1 , u_3 \geqslant 0 . \\ (3) & u_3 = \tfrac{1}{2}-\tfrac{1}{3}u_1-\tfrac{2}{3}u_3 . \end{array}$$

When u_1 and u_3 take their values in the interval $[0, \infty]$ point $[x] = [x_1 \ x_2]$ traverses an edge of the cone

$$(14.88) \qquad -x_1 \leqslant -1/2 , \qquad 2x_1+3x_2 \leqslant 4 ,$$

from $[x^{(0)}]$ toward $[x^{(2)}]$.

Finally, if we made u_1 enter the basis to replace u_3, we should obtain the third edge that runs from $[x^{(3)}]$ to $[x^{(0)}]$.

We thus have the three extremities:

$$(14.89) \qquad \begin{array}{l} [x^{(0)}] \text{ corresponding to } u_1 = u_3 = 0 , \\ [x^{(1)}] \text{ corresponding to } u_1 = u_2 = 0 , \\ [x^{(2)}] \text{ corresponding to } u_2 = u_3 = 0 . \end{array}$$

As we can see, as often as we include a u_i in the basis in place of a u_j ($j \neq i$), we pass from an extremity to another adjacent to it (in our rather slight example, since there are only three extremities, thay are all adjacent, two by two, but this would clearly not be the case for more complex polyhedrons).

The simplex method and that of some of its variants consists, as we should recall, in passing from one extreme point to an adjacent one by improving the economic function and by making sure at each step that the variables x_i and u_j retain nonnegative values. As we have seen in Volume 1, this is achieved by the use of two of Dantzig's criteria. We shall return to certain aspects of these considerations shortly.

Extreme Rays

Let us consider an example of a convex polyhedron in $\mathbf{R}^2$ (Fig. 14.11) the inequations of which are

$$(14.90) \qquad \begin{array}{ll} (1) & x_2 \leqslant 2 , \\ (2) & -x_1-2x_2 \leqslant -2 , \\ (3) & -2x_1-x_2 \leqslant -2 . \end{array}$$

Let us introduce the deviation variables u_1, u_2, and u_3, which gives

$$
(14.91)\qquad
\begin{aligned}
&(1)\quad x_2+u_1 = 2,\\
&(2)\quad -x_1-2x_2+u_2 = -2, \qquad u_1, u_2, u_3 \geqslant 0.\\
&(3)\quad -2x_1-x_2+u_3 = -2,
\end{aligned}
$$

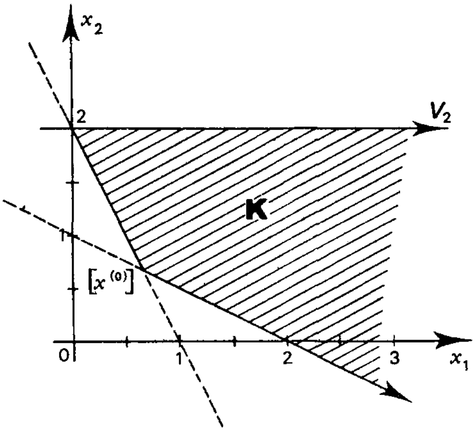


FIG. 14.11

If we take in the basis x_1, x_2, and u_1, it follows that

$$
(14.92)\qquad
\begin{aligned}
&(1)\quad x_1 = \tfrac{2}{3} - \tfrac{1}{3}u_2 + \tfrac{2}{3}u_3,\\
&(2)\quad x_2 = \tfrac{2}{3} + \tfrac{2}{3}u_2 - \tfrac{1}{3}u_3, \qquad u_1, u_2, u_3 \geqslant 0,\\
&(3)\quad u_1 = \tfrac{8}{3} + \tfrac{2}{3}u_2 - \tfrac{1}{3}u_3.
\end{aligned}
$$

In (14.92) if we take $u_2 = 0$ and $u_3 \to \infty$, we see that the point $[x] = [x_1\ x_2]$ will traverse the half-line V_1, the equation of which is

$$(14.93)\qquad x_1+2x_2 = 2;$$

this from the extreme point $[x^{(0)}] = [2/3\ 2/3]$ to infinity, the length of this half-line. Such a half-line of the convex polyhedron is termed the *extreme ray*. We can see that another extreme ray V_2 exists corresponding to $x_2 = 2$.

We shall make use of these concepts in illustrating the dual-simplex method and various methods of partition (Sections 16 and 21).

Finally, before concluding this section, let us define what is termed a *convex combination* of points and set forth two important theorems.

6. Convex Combination

Let there be r points $[x^{(i)}] \in \mathbf{X}$ a convex subset of $\mathbf{R}^n$, $i = 1, 2, ..., r$, such that

$$(14.94) \qquad \lambda_i \geqslant 0\,, \quad i = 1, 2, \ldots, r\,, \quad \text{with} \sum_{i=1}^{r} \lambda_i = 1\,.$$

Then, the point $[x]$ such that

$$(14.95) \qquad [x] = \sum_{i=1}^{r} \lambda_i . [x^{(i)}]$$

is called the *convex combination* of the points $[x^{(i)}]$, $i = 1, 2, ..., r$.

Theorem 14.III

If $[x]$ is a *convex combination* of r points $[x^{(i)}]$ of a convex set $\mathbf{X}$, then $[x]$ belongs to $\mathbf{X}$.

Proof

This is purely and simply the generalization of (14.5). We need only make a convex combination of $[x^{(2)}]$, $[x^{(1)}]$, then a convex combination of the result with $[x^{(3)}]$ and so on, with the requirement of taking the weight λ_1 for $[x^{(1)}]$, then a weight λ_2 for $[x^{(2)}]$, and so forth.

Theorem 14.IV

Given a convex closed set $\mathbf{X} \subset \mathbf{R}^n$, that is to say, such that all its points have finite components, then every point $[x] \in \mathbf{X}$ is a convex combination of the extreme points $[x^{*(i)}]$.

The proof of this theorem is a long and difficult one. It is an important theorem, but we have avoided having to use it for the logical procedure in the questions that are treated here. The reader who wishes to learn this proof can consult [K3], though it should be observed that its significance is of an intuitive nature.

Section 15. **Properties of the Optimums of Convex and Concave Functions**

1. Convex Functions and Concave Functions

It is now time to turn our attention, in an economic program, not only to the domain in which the variables assume values corresponding to solutions, but also to the economic function that enables us to select the optimal solutions (or solutions).

If $[x]$ is a point of $\mathbf{X} \subset \mathbf{R}^n$ we shall indicate the function taking its values in $\mathbf{X}$ by $f([x])$. The distance between two points $[x]$ and $[x']$ will be represented as $|[x]-[x']|$ and will be equal to

$$\sqrt{(x_1-x_1')^2+(x_2-x_2')^2+\ldots+(x_n-x_n')^2}.$$

Let us first consider the concept of a *local minimum.* A function $f([x])$ has a local minimum for $[x^{(0)}]$ if there is a $\varepsilon > 0$ such that $f([x^{(0)}]) \leqslant f([x])$ for each $|[x]-[x^{(0)}]| < \varepsilon$. In the same way we define a *local maximum* if there is an $\varepsilon > 0$ such that $f([x^{(0)}]) \geqslant f([x])$ for all $|[x]-[x^{(0)}]| < \varepsilon$.

If we have $f([x^{(0)}]) < f([x])$ or, respectively, $f([x^{(0)}]) > f([x])$, we say that we have, respectively, a *strict local minimum* and a *strict local maximum.*

Let us now consider another concept, that of the *global minimum.* A function $f([x])$, $[x] \in \mathbf{X}$ has a global minimum for $[a] \in \mathbf{X}$, if $f([a]) \leqslant f([x])$ for every $[x] \in \mathbf{X}$. We can similarly define a *global maximum* by changing the direction of this inequality.

It should be observed that there may be more than one global minimum or maximum.

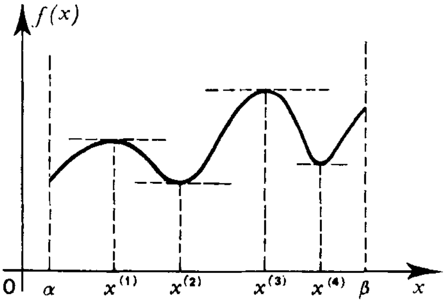


FIG. 15.1

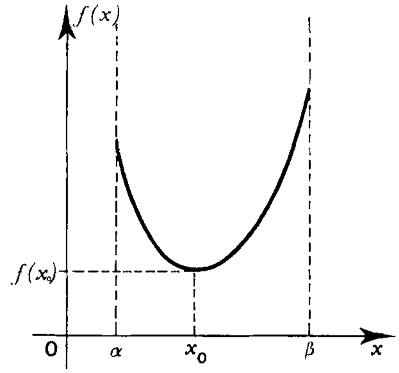


FIG. 15.2

In Fig. 15.1 we have shown an example where $\mathbf{X} \subset \mathbf{R}$. The function $f([x])$, which can be expressed here as $f(x)$ without ambiguity, assumes its values in the interval $[\alpha, \beta]$. Points $x=\alpha$, $x=x^{(2)}$, $x=x^{(4)}$ are local minimums. Points $x=x^{(1)}$, $x=x^{(3)}$, $x=\beta$ are local maximums. Point $x=x^{(2)}$ is a global minimum and point $x=x^{(3)}$ a global maximum.

For convenience and to save space we shall henceforward generally refer to the minimum, only transposing to the concept of the maximum when this word is placed in parentheses after the other.

In general, a local minimum (maximum) is not always a global minimum (maximum). For convex functions a local minimum is a global minimum, the same being true for the maximum.

In Fig. 15.2 we have shown a *convex function*; its sole local minimum in the interval $[\alpha, \beta]$ is a global minimum that can easily be transposed for the maximum.

Let us now leave this example in $\mathbf{R}$ to give a strict definition of the same concepts in $\mathbf{R}^n$, $n = 1, 2, 3, 4, \ldots$.

A function $f([x])$ defined for a convex subset $\mathbf{X} \subset \mathbf{R}^n$ is *convex* if and only if

$$(15.1) \qquad \forall [x^{(1)}], [x^{(2)}] \in \mathbf{X} :$$

$$f(\lambda [x^{(1)}] + (1-\lambda)[x^{(2)}]) \leqslant \lambda f([x^{(1)}]) + (1-\lambda) f([x^{(2)}]),$$

$$0 \leqslant \lambda \leqslant 1 .$$

Let us give this formula a strict geometrical definition. Let us assume

$$(15.2) \qquad z = f([x]) .$$

Let us then consider the point $[x_1\, x_2 \cdots x_n\, z]$ in the space $\mathbf{R}^{n+1}$. In this space, (15.2) represents a surface. If we take any two points in it $[x_1^{(1)}\, x_2^{(1)} \cdots x_n^{(1)}\, z^{(1)}]$ and $[x_1^{(2)}\, x_2^{(2)} \cdots x_n^{(2)}\, z^{(2)}]$, then relation (15.1) shows that if we join them by a segment, every point in this segment is *above* the surface.

Let us state,

$$(15.3) \qquad [x] = \lambda [x^{(1)}] + (1-\lambda)[x^{(2)}], \qquad [x^{(1)}], [x^{(2)}] \in \mathbf{X},$$

$$(15.4) \qquad z = f([x]),$$

$$(15.5) \qquad z^{(1)} = f([x^{(1)}]),$$

$$(15.6) \qquad z^{(2)} = f([x^{(2)}]),$$

$$(15.7) \qquad w = \lambda z^{(1)} + (1-\lambda) z^{(2)}.$$

Relation (15.1) shows

$$(15.8) \qquad z \leqslant w .$$

Figures 15.3 and 15.4, respectively, represent convex functions defined for convex subsets, the first for $\mathbf{X} \subset \mathbf{R}$, the second for $\mathbf{X} \subset \mathbf{R}^2$.

In the same way, let us define a concave function.

A function $f([x])$ defined for a convex subset $\mathbf{X} \subset \mathbf{R}^n$ is *concave* if and only if

$$\forall [x^{(1)}], [x^{(2)}] \in \mathbf{X} :$$

$$(15.9) \qquad f(\lambda [x^{(1)}] + (1-\lambda)[x^{(2)}]) \geqslant \lambda f([x^{(1)}]) + (1-\lambda) f([x^{(2)}]),$$

$$0 \leqslant \lambda \leqslant 1,$$

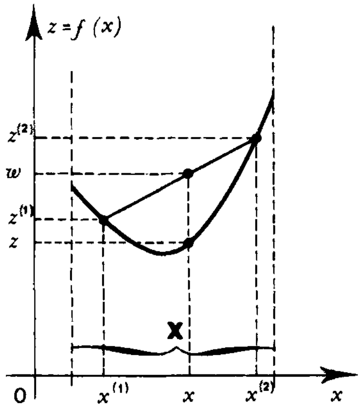


FIG. 15.3

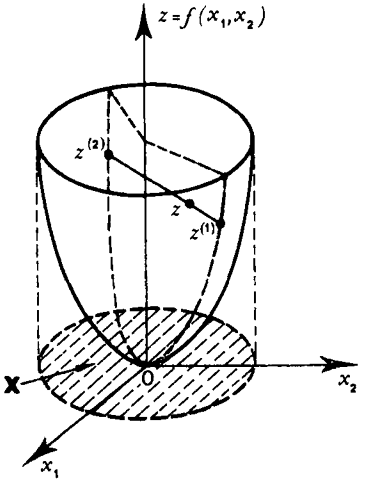


FIG. 15.4

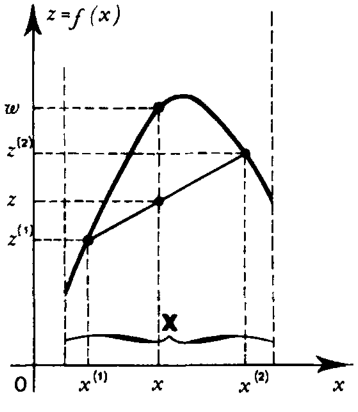


FIG. 15.5

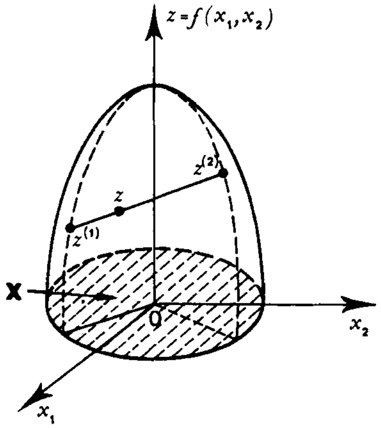


FIG. 15.6

that is to say, with the notation specified in (15.3)–(15.7),

$$z \geqslant w. \tag{15.10}$$

Figures 15.5 and 15.6, respectively, illustrate concave functions defined for convex subsets, the first for $\mathbf{X} \subset \mathbf{R}$, the second for $\mathbf{X} \subset \mathbf{R}^2$.

A particularly important case concerns functions $f([x])$ that can be differentiated. In this case, we can define the convexity and concavity in a more convenient form. Here we shall consider the case of a convex domain $\mathbf{X} \subset \mathbf{R}$ but, in a next volume, when studying an important theorem of Kuhn and Tucker we shall extend these considerations to the case where $\mathbf{X} \subset \mathbf{R}^n$.

Given a convex function $f(x)$, we can state in accordance with (15.1),

$$f(\lambda x_1 + (1-\lambda)\, x_2) \leqslant \lambda f(x_1) + (1-\lambda) f(x_2), \tag{15.11}$$

or again

$$f((1-\lambda)\, x_1 + \lambda x_2) \leqslant (1-\lambda) f(x_1) + \lambda f(x_2), \tag{15.12}$$

that is to say,

$$f(x_1 + \lambda(x_2 - x_1)) \leqslant f(x_1) + \lambda(f(x_2) - f(x_1)), \tag{15.13}$$

or

$$\frac{f(x_1 + \lambda(x_2 - x_1)) - f(x_1)}{\lambda} \leqslant f(x_2) - f(x_1). \tag{15.14}$$

Let us divide the two members of (15.14) by $(x_2 - x_1)$. It follows that

$$\frac{f(x_1 + \lambda(x_2 - x_1)) - f(x_1)}{\lambda(x_2 - x_1)} \leqslant \frac{f(x_2) - f(x_1)}{(x_2 - x_1)}. \tag{15.15}$$

Assuming

$$\Delta x_1 = \lambda(x_2 - x_1), \tag{15.16}$$

it follows that

$$\frac{f(x_1 + \Delta x_1) - f(x_1)}{\Delta x_1} \leqslant \frac{f(x_2) - f(x_1)}{(x_2 - x_1)}. \tag{15.17}$$

Let us make $\lambda \to 0$, that is, $\Delta x_1 \to 0$; we then have

$$\lim_{\Delta x_1 \to 0} \frac{f(x_1 + \Delta x_1) - f(x_1)}{\Delta x_1} = f'(x_1) \tag{15.18}$$

and relation (15.17) can be expressed

$$f'(x_1) \leqslant \frac{f(x_2) - f(x_1)}{x_2 - x_1}. \tag{15.19}$$

For a concave function, in the same conditions, we shall write

$$f'(x_1) \geqslant \frac{f(x_2) - f(x_1)}{x_2 - x_1}. \tag{15.20}$$

In Fig. 15.7 we have illustrated (15.19) by an example. Function $f(x)$ is convex; if, from any point x_1, we draw a straight line passing through another point x_2, the slope of this line is always greater than that of the derivative of $f(x)$ in x_1. The concave property of (15.20) could be similarly shown.

Let us now consider the important case of the linear functions $f([x])$.

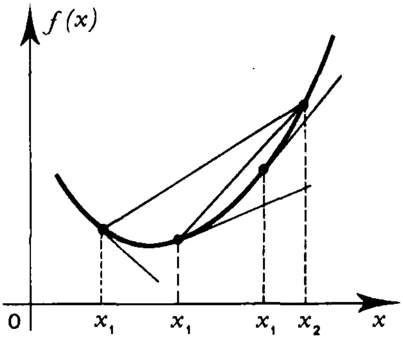


FIG. 15.7

Theorem 15.I

If a function $f([x])$ is linear and is defined for a convex set $\mathbf{X} \subset \mathbf{R}^n$, then

$$\forall [x^{(1)}], [x^{(2)}] \in \mathbf{X}:$$

$$f(\lambda[x^{(1)}] + (1-\lambda)[x^{(2)}]) = \lambda f([x^{(1)}]) + (1-\lambda) f([x^{(2)}]), \tag{15.21}$$

$$0 \leqslant \lambda \leqslant 1,$$

which shows that every such linear function that is defined for a convex set is both a convex and concave function defined for a convex domain.

Proof

If a function $f([x])$ is linear, we have, by hypothesis,

$$f(k[x]) = kf([x]), \tag{15.22}$$

$$f([x] + [x']) = f([x]) + f([x']), \tag{15.23}$$

which enables us to state further that

$$\begin{aligned} f(k[x] + k'[x']) &= f(k[x]) + f(k'[x']) \\ &= kf([x] + k'f([x']). \end{aligned} \tag{15.24}$$

By taking the special case where $k+k' = 1$, $k, k' \geqslant 0$ we shall discover (15.21).

2. Optimum of a Convex Function in a Convex Domain

Let us state two theorems.

Theorem 15.II

If $f([x])$ is a convex function defined for a convex set $\mathbf{X} \subset \mathbf{R}^n$, then a local minimum (maximum) of $f(x)$ is a global minimum (maximum) of $f(x)$.

Proof

If we give the proof for the minimum this can at once be transposed for the maximum.

If $[x^{(0)}]$ is a local minimum of $f([x])$ there is an adjacent area to $[x^{(0)}]$ such that $f([x^{(0)} \leqslant f([x])$. Let us therefore assume that there is a point $[a] \in \mathbf{X}$ such that

$$f([a]) < f([x^{(0)}]). \tag{15.25}$$

All the points

$$[\bar{x}] = \lambda[a] + (1-\lambda)\,[x^{(0)}], \qquad 0 \leqslant \lambda \leqslant 1, \tag{15.26}$$

belong to **X** since it is convex. Let us take λ small enough to be in the neighborhood of $[x^{(0)}]$ but differing from zero so that $[\bar{x}] \neq [x^{(0)}]$. We have

$$f([\bar{x}]) \geqslant f([x^{(0)}]), \tag{15.27}$$

since $[x^{(0)}]$ is a local minimum.

We can also state

$$f([\bar{x}]) \leqslant \lambda f([a] + (1-\lambda)f([x^{(0)}]), \tag{15.28}$$

since $f([x])$ is convex. In addition, by starting from (15.27) and considering (15.25), we can say after observing that $f[x^{(0)}] = \lambda f[x^{(0)}] + (1-\lambda)f[x^{(0)}]$ and that $\lambda \neq 0$,

$$f([\bar{x}]) > \lambda f([a]) + (1-\lambda)f([x^{(0)}]). \tag{15.29}$$

If we compare (15.28) and (15.29) we discover a contradiction. Hence, the assumption that there is a point $[a] \in \mathbf{X}$ differing from $[x^{(0)}]$, such that $[a]$ is a global minimum, leads to an absurdity. The theorem is thus proved by contradiction.

Theorem 15.III

The set of local minimums (maximums) of a convex function that are also global minimums (maximums) in accordance with Theorem 15.II is a convex set.

Proof

It is sufficient to show that the points of the segment joining two local minimums (maximums), namely $[a]$ and $[b]$, are global minimums (maximums).

Let

$$[x] = \lambda[a] + (1-\lambda)\,[b]. \tag{15.30}$$

By hypothesis, we have

$$f([a]) = f([b]), \tag{15.31}$$

since $[a]$ and $[b]$ are global minimums (maximums).

In addition,

$$f([x]) \geqslant f([a]), \tag{15.32}$$

since $[a]$ is a global minimum (or maximum if we replace $\geqslant$ by $\leqslant$).

But we also have

$$f([x]) \leqslant \lambda f([a]) + (1-\lambda) f([b]) = f([a]). \tag{15.33}$$

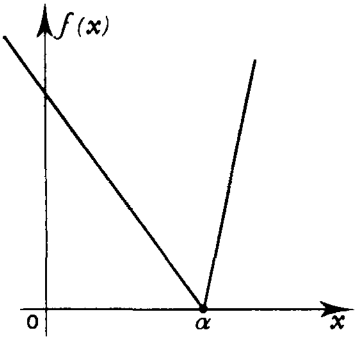
(we replace $\leqslant$ by $\geqslant$ if we are considering a maximum).

From (15.32) and (15.33) we obtain

$$f([x]) = f([a]); \tag{15.34}$$

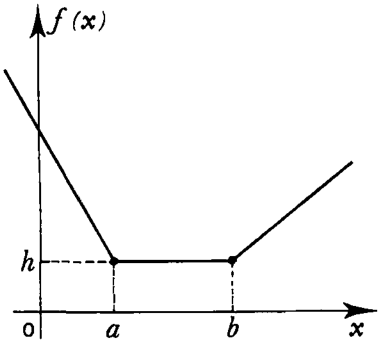
which shows that $[x]$, situated on the segment joining $[a]$ and $[b]$ is also a global minimum (maximum).

Figures 15.3–15.6 and 15.8 serve to illustrate Theorem 15.II, while Theorem 15.III is illustrated by Fig. 15.9.



$$\begin{aligned} f(x) &= k_1(\alpha - x), \quad k_1>0, \quad x \leqslant \alpha, \\ &= k_2(x-\alpha), \quad k_2>0, \quad x \geqslant \alpha. \end{aligned}$$

Fig. 15.8



$$\begin{aligned} f(x) &= k_1(a-x)+h, \quad k_1>0, x \leqslant a, \\ &= h, \quad a \leqslant x \leqslant b, \\ &= k_2(x-b)+h, \quad k_2>0, \ x \geqslant b. \end{aligned}$$

Fig. 15.9

3. Optimum of a Concave Function in a Convex Domain

Let us first enunciate a theorem.

Theorem 15.IV

A strict local minimum of a concave function $f([x])$ in a convex domain $\mathbf{X} \subset \mathbf{R}^n$ corresponds to an extreme point of $\mathbf{X}$.

This theorem is a fortiori true for a strict global minimum.

Proof

If $[x^{(0)}]$ is not an extreme point of $\mathbf{X}$ we have $[x^{(0)}] = \lambda'[x'_1] + (1-\lambda')[x'_2]$, with $[x'_1]$ and/or $[x'_2]$ differing from $[x^{(0)}]$ and $\lambda \in [0, 1]$.

Since domain **X** is convex we can find two points $[x_1]$ and $[x_2]$ on the segment joining $[x'_1]$ and $[x'_2]$ such that

$$[x^{(0)}] = \lambda[x_1] + (1-\lambda)\,[x_2], \qquad 0 \leqslant \lambda \leqslant 1, \tag{15.35}$$

with

$$|[x^{(0)}] - [x_1]| \leqslant \varepsilon, \qquad \varepsilon \geqslant 0, \tag{15.36}$$

and

$$|[x^{(0)}] - [x_2]| \leqslant \varepsilon, \qquad \varepsilon \geqslant 0, \tag{15.37}$$

in which $[x_1]$ and/or $[x_2]$ differ from $[x^{(0)}]$, that is to say, $\lambda \neq 0$ and/or $\lambda \neq 1$ in (15.35). Let us arbitrarily assume that this point is $[x_1]$. Since $f([x])$ is concave, we have

$$f([x^{(0)}]) \geqslant \lambda f([x_1]) + (1-\lambda)\, f([x_2]). \tag{15.38}$$

We now have

$$f([x^{(0)}]) < f([x_1]), \tag{15.39}$$

since $[x^{(0)}]$ is a strict local minimum and $[x_1] \neq [x^{(0)}]$. In addition we have

$$f([x^{(0)}] \leqslant f\,[(x_2]). \tag{15.40}$$

By multiplying the two members of (15.39) by λ and those of (15.40) by $(1-\lambda)$ with $\lambda \neq 0$, and by adding the resulting inequalities, we obtain a contradiction with (15.38).

This proof by absurdity can easily be transposed for the case of the maximum.

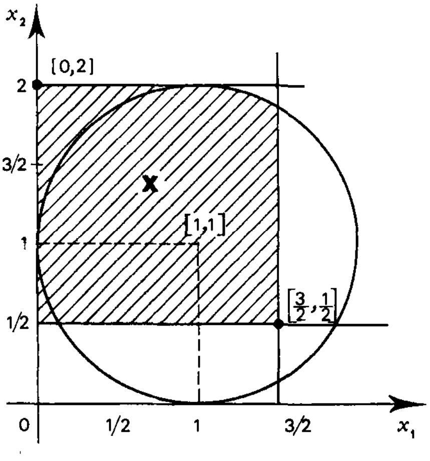


FIG. 15.10

Let us consider an example, and let us take the following program:

$$(15.41)\qquad \begin{aligned} &(1)\ \ [\mathrm{MIN}]\ z = -(x_1-1)^2-(x_2-1)^2,\\ &(2)\ \ 0 \leqslant x_1 \leqslant 3/2,\\ &(3)\ \ 1/2 \leqslant x_2 \leqslant 2. \end{aligned}$$

It is evident that $\mathbf{X}$ defined by (2) and (3) is a convex set since it is a square in $\mathbf{R}^2$. It is also evident that (1) represents a concave function. We shall use Fig. 15.10 for our explanations.

We can verify that points $[x_1^{(0)}\ x_2^{(0)}] = [3/2\ 1/2]$ and $[x_1'^{(0)}\ x_2'^{(0)}] = [0\ 2]$ are local minimums. It can be shown that any point adjacent to them gives a greater value than z. Point $[3/2\ 1/2]$ is a local minimum but not a global minimum; by contrast, $[0\ 2]$ is a global minimum. For the former we have $z=-1/2$ and for the latter $z=-1$. We can easily verify that $[0\ 2]$ is an extreme point of $\mathbf{X}$.

We find here one of the peculiarities of the minimization (maximization) of concave (convex) functions. While it is easy to obtain a local minimization (maximization) we cannot affirm that it is global, as was the case for convex (concave) functions. The difference between the value of z in a global optimum and a local optimum may be considerable in relation to the value of z. We can see how useful it is to find the global minimum.

4. Relation between Convex and Concave Functions. Conversion of Maximums into Minimums and Conversely

Let us consider the program,

$$(15.42)\qquad \begin{aligned} &(1)\ \ [\mathrm{MIN}]\ z = f([x]),\\ &(2)\ \ [x] \in \mathbf{X} \subset \mathbf{R}^n. \end{aligned}$$

Let us suppose that $\mathbf{X}$ is any closed domain and that z has a global minimum $[a]$. By definition, we have

$$(15.43)\qquad \forall [x] \in \mathbf{X}: \qquad f([x]) \geqslant f([a]),$$

hence

$$(15.44)\qquad \forall [x] \in \mathbf{X}: \qquad -f([x]) \leqslant -f([a]).$$

Hence, if $[a]$ is a global minimum of $f([x])$, then it is also a global maximum of $-f([x])$ in the same domain $\mathbf{X}$. This is expressed

$$[\mathrm{MIN}]\, f([x]) = -[\mathrm{MAX}]\,(-f([x])).$$

It is agreed that the symbols [MIN] and [MAX], respectively, represent the search for global minimums and maximums.

In addition, if $f([x])$ is convex, $-f([x])$ is concave, and hence we can transpose the results of Theorem 15.II for the case of convex functions. We shall give the results obtained from these theorems in the form of table (15.49) at the end of this section.

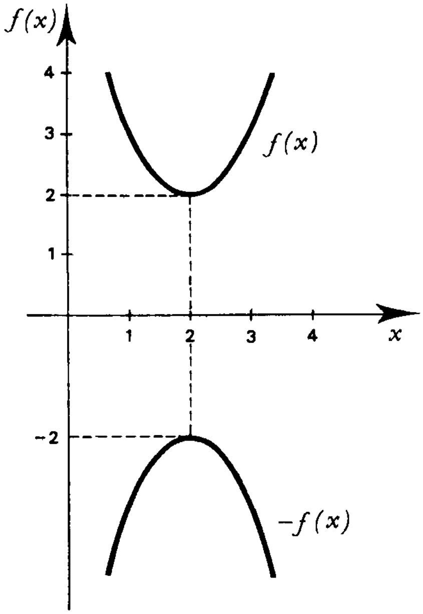


FIG. 15.11

Let us consider an example. Let

$$(15.45)\quad \begin{aligned}&(1)\quad [\text{MIN}]\ z = f(x) = 2 + (x-2)^2,\\ &(2)\quad x \in \mathbf{R}.\end{aligned}$$

Corresponding to it is

$$(15.46)\quad \begin{aligned}&(1)\quad [\text{MAX}]\ z = -f(x) = -2 - (x-2)^2,\\ &(2)\quad x \in \mathbf{R}.\end{aligned}$$

By elementary calculation we obtain

$$(15.47)\qquad \min f(x) = f(2) = 2.$$

and

$$(15.48)\qquad \max(-f(x)) = -f(2) = -2.$$

This maximum and this minimum are reached at the same point $x = 2$.

The results previously obtained can be conveniently displayed in the form of table (15.49).

(15.49)

$f([x])$ is convex	$f([x])$ is concave
A local minimum is a global minimum.	A local maximum is a global maximum.
A strict local maximum can only be reached at an extreme point.	A strict local minimum can only be reached at an extreme point.

The results shown in table (15.49) are true for every convex domain and hence they apply to the special case where $\mathbf{X}$ is a convex polyhedron.

Section 16. Complements on the Theory of Linear Programming

1. Complementary Properties of Primal and Dual Solutions

We propose to expand and complete various concepts introduced in the second part of Volume 1 concerning linear programming and, in particular, the question of duality that was explained very briefly in Section 66 of that volume.

Given a linear program

$$(16.1)\qquad \begin{aligned} &(1)\quad [\mathrm{MAX}]\ g = \sum_{j=1}^{n} c_j x_j, \\ &(2)\quad \sum_{j=1}^{n} a_{ij}\cdot x_j \leqslant b_i, \qquad i = 1, 2, \ldots, m, \\ &(3)\quad x_j \in \mathbf{R}, \qquad x_j \geqslant 0, \qquad j = 1, 2, \ldots, n. \end{aligned}$$

Or, in matrical notation,

$$(16.2)\qquad \begin{aligned} &(1)\quad [\mathrm{MAX}]\ g = [c]'_{1\times n}\cdot[x]_{n\times 1}, \\ &(2)\quad [a]_{m\times n}\cdot[x]_{n\times 1} \leqslant [b]_{m\times 1}, \\ &(3)\quad [x]_{n\times 1} \in \mathbf{R}^n, \qquad [x]_{n\times 1} \geqslant [0]_{n\times 1}. \end{aligned}$$

Let us then introduce m deviation variables u_i, $i = 1, 2, \ldots, m$, to transform the inequations (2) of (16.2) into equations; it follows that, by indicating the matrix column of u_i as $[u]_{m\times 1}$,

$$
(16.4)^1 \qquad
\begin{aligned}
&(1)\quad [\mathrm{MAX}]\ g = [c]'_{1\times n}\cdot[x]_{n\times 1},\\
&(2)\quad [a]_{m\times n}\cdot[x]_{n\times 1} + [1]_{m\times m}\cdot[u]_{m\times 1} = [b]_{m\times 1},\\
&(3)\quad [x]_{n\times 1}\in\mathbf{R}^n, \qquad [u]_{m\times 1}\in\mathbf{R}^m,\\
&\qquad\ \ [x]_{n\times 1}\geqslant[0]_{n\times 1}, \qquad [u]_{m\times 1}\geqslant[0]_{m\times 1}.
\end{aligned}
$$

Let us suppose that we find ourselves, after iteration k of the simplex method, at an extreme point of the convex polyhedron of the constraints (2) and (3) of (16.4). This point is determined by choosing n of the $n+m$ variables x_j and u_i in such a manner that they are null. In effect, this corresponds to the intersection of n of the $n+m$ hyperplanes $[a]_{m\times n}\cdot[x]_{n\times 1}=[b]_{m\times 1}$ and $[x]_{n\times 1}=[0]_{n\times 1}$. The n zero variables do not belong to the basis.

Let us indicate by $[x_B]_{m\times 1}$ the matrix column of the variables corresponding to the basis, these variables being selected from the n variables x_j and the m variables u_i, and $[x_N]_{n\times 1}$ indicating those of the variables that do not belong to the basis. Let us indicate by $[B]_{m\times 1}$ the columns of the matrix formed by $[a]_{m\times n}$ and $[1]_{m\times m}$, namely $[[a]\ [1]]$, which does not belong to the basis. Let us specify by $[N]_{m\times n}$ the matrix formed with the other columns. In the same way, in function g, let us indicate by $[C_B]_{1\times n}$ the coefficients of the variables that belong to the basis and the remainder by $[C_N]_{1\times n}$.

Using the above notation, let us now express the set of relations (1) and (2) of (16.4),

(16.5)

$$
\begin{bmatrix}
[1]_{1\times 1} & -[c_B]'_{1\times m} & -[c_N]'_{1\times n} & [0]_{1\times m}\\
[0]_{m\times 1} & [B]_{m\times m} & [N]_{m\times n} & [1]_{m\times m}
\end{bmatrix}
\cdot
\begin{bmatrix}
[g]_{1\times 1}\\
[x_B]_{m\times 1}\\
[x_N]_{n\times 1}\\
[\varphi]_{m\times 1}
\end{bmatrix}
=
\begin{bmatrix}
[0]_{1\times 1}\\
[b]_{m\times 1}
\end{bmatrix},
$$

where we have added matrices $[0]_{1\times m}$, $[1]_{m\times m}$ (the unit matrix of order m) and $[\varphi]_{m\times 1}$ in order later to produce, in the corresponding simplex table that will be used, at iteration k, the inverse $[B]^{-1}_{m\times m}$ of the basis matrix $[B]$. This notation does not in any way affect the set of the solutions of (16.4) but greatly assists the explanations, as will appear. Matrix $[\varphi]_{m\times 1}$ will have as its elements φ_i, $i=1, 2, \ldots, m$. We might interpret the φ_i terms as artificial variables that should disappear as soon as we obtain a solution for (16.4).

Let us now consider the following square submatrix $(m+1)(m+1)$, removed

[1] Equation number (16.3) omitted in the French edition.

from the left member of (16.5):

$$(16.6)\qquad \begin{bmatrix} [1]_{1\times 1} & -[c_B]'_{1\times m} \\ [0]_{m\times 1} & [B]_{m\times m} \end{bmatrix}_{(m+1)\times(m+1)}$$

This matrix is clearly regular and therefore possesses an inverse; indeed $[B]_{m\times m}$is, by hypothesis, regular since it forms a basis.

The inverse matrix of (16.6) is

$$(16.7)\qquad \begin{bmatrix} [1]_{1\times 1} & [c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m} \\ [0]_{m\times 1} & [B]^{-1}_{m\times m} \end{bmatrix}_{(m+1)\times(m+1)} .$$

Let us then premultiply the two members of (16.5) by matrix (16.7); it follows that

(16.8)

$$\begin{bmatrix} [1]_{1\times 1} & [0]_{1\times m} & -[c_N]'_{1\times n}+[c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[N]_{m\times n} & [c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m} \\ [0]_{m\times 1} & [1]_{m\times m} & [B]^{-1}_{m\times m}\cdot[N]_{m\times n} & [B]^{-1}_{m\times m} \end{bmatrix}$$

$$\cdot\begin{bmatrix} [g]_{1\times 1} \\ [x_B]_{m\times 1} \\ [x_N]_{n\times 1} \\ [\varphi]_{m\times 1} \end{bmatrix} = \begin{bmatrix} [c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[b]_{m\times 1} \\ [B]_{m\times m}^{-1}\cdot[b]_{m\times 1} \end{bmatrix}.$$

The matrical relation (16.8) will be called the *simplex table* corresponding to the choice of basis $[B]_{m\times m}$. Let us observe, finally consulting Section 14.3, that we can obtain from it the relation

$$(16.9)\qquad [x_B]_{m\times 1} = [B]^{-1}_{m\times m}\cdot[b]_{m\times 1} - [B]^{-1}_{m\times m}\cdot[N]_{m\times n}\cdot[x_N]_{n\times 1}$$

$$\text{for } [\varphi]_{m\times 1} = [0]_{m\times 1},$$

which is the explicit equation of the polyhedral cone with vertex $[x_B]$ having as edges the vector columns of the matrix $[B]^{-1}.[N]$.

Let us now enunciate a theorem.

Theorem 16.1

The matrical relation (16.8) corresponds to the global maximum if

$$(16.10)\qquad (1)\quad [B]^{-1}_{m\times m}\cdot[b]_{m\times 1} \geqslant [0]_{m\times 1},$$

$$(16.11)\qquad (2)\quad -[c_N]'_{1\times n}+[c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[N]_{m\times n} \geqslant [0]_{1\times n},$$

$$(16.12)\qquad (3)\quad [c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m} \geqslant [0]_{1\times m}.$$

The optimal solution of (16.4) will be indicated by $[\hat{\xi}]_{(n+m)\times 1} = [[\hat{x}]_{n\times 1}\,[\hat{u}]_{m\times 1}]$ and $[\hat{X}_B]_{m\times 1}$ will indicate those variables the values of which are given by the left member of (16.10) corresponding to the basis, and $[\hat{X}_N]_{n\times 1}$ will indicate the others.

Proof

In every realizable solution we must have $[\varphi]_{m\times 1} = [0]_{m\times 1}$ and, expanding (16.8), we then obtain for the point $[\hat{\xi}]$

$$(16.13) \qquad [g]_{1\times 1} = [c_B]'_{1\times m} \cdot [B]^{-1}_{m\times m} \cdot [b]_{m\times 1}$$
$$+ ([c_N]'_{1\times m} - [c_B]'_{1\times m} \cdot [B]^{-1}_{m\times m} \cdot [N]_{m\times m}) \cdot [\hat{x}_N]_{n\times 1},$$

$$(16.14) \qquad [\hat{x}_B]_{m\times 1} = [B]^{-1}_{m\times m} \cdot [b]_{m\times 1} - [B]_{m\times m} \cdot [N]_{m\times m} \cdot [\hat{x}_N]_{n\times 1}.$$

To verify (16.10) note that $[\hat{x}_N]$ must be a null vector (whose components are not basis variables). Replacing $[\hat{x}_N]$ in (16.14) by $[0]$ (and as $[\hat{x}_B]$ must be nonnegative for the solution to be realizable), we obtain

$$(16.15) \qquad [B]^{-1}_{m\times m} \cdot [b]_{m\times 1} \geqslant [0]_{m\times 1},$$

which is none other than the sought condition (16.10).

To verify (16.11), we consider the relation (16.13).

As $[\hat{x}_N]_{n\times 1} \geqslant [0]_{m\times 1}$, we can have a maximum[1] only if

$$(16.16) \qquad [c_N]'_{1\times n} - [c_B]'_{1\times m} \cdot [B]^{-1}_{m\times m} \cdot [N]_{m\times m} \cdot [0]_{1\times n},$$

which gives the relation (16.11) by multiplying (16.16) by -1 and changing the sense of the inequality.

To prove (16.12) is a little more complicated, and for that we are going to use relation (16.11). Let u_i be the slack variable in one of the inequalities (16.4). Two cases are possible: u_i is or is not a basis variable in the optimal solution.

(*i*) *First case: u_i in the basis.*

We suppose that u_i has the index $_{B_i}$ (which we can always obtain by reordering the columns of $[B]$). Then the ith column of $[B]$ is

$$\begin{matrix} 0 \\ 0 \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ 0 \end{matrix} \qquad \text{and the } i\text{th column of } [B]^{-1} \text{ is} \qquad \begin{matrix} 0 \\ 0 \\ \cdot \\ 1 \\ \cdot \\ \cdot \\ 0 \end{matrix}\ .$$

[1] The expression $2+3x_1-5x_2-6x_3$ is not the maximum for $x_1 = x_2 = x_3 = 0$ since the value increases when x_1 has a positive value with x_2 and x_3 equal to zero.

In contrast, $2-3x_1-5x_2-6x_3$ has a global maximum equal to 2 for $x_1 = x_2 = x_3 = 0$, when x_1, x_2, x_3 can only take nonnegative values.

With these conventions, we have $c_{B_i} = 0$ since the cost of u_i is zero in the economic function[1] of program (16.4). The ith component of the row vector $[c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}$ is written

$$(16.17)\qquad [([c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m})]\quad \text{and also } [c_B]'_{1\times m}\cdot[B^{-1}]^i,$$

or even c_{B_i}—following the expression for the column $[B^{-1}]^i$. The ith component of the row vector $[c_B]_{1\times m}\cdot[B^{-1}]_{m\times m}$ is then zero in this case.

(*ii*) *Second case: u_i is not in the basis.*

Let us suppose that u_i has index ${}_{N_i}$ (which we may always obtain by reordering the columns of the matrix $[N]_{m\times n}$). The ith component of the row vector (16.11) may be written

$$(16.18)\qquad [(-[c_N]'_{1\times n}+[c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[N]_{m\times n})]^i,$$

and also

$$(16.19)\qquad -c_{N_i}+[c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[N]^i_{m\times 1}.$$

We have $c_{N_i} = 0$ (the cost of u_i is zero), and $[N]^i_{1\times m}$ is a column vector of zeros with a 1 in the ith row. We can then write (16.19): $[([c_B]'_{1\times m}[B^{-1}]_{m\times m})]^i$ which must be greater than or equal to zero since (16.18) is a component of the row vector (16.11), which is always greater than or equal to zero, according to what we have previously shown.

Then $[c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}\geqslant[0]_{1\times m}$ since all the components of this row vector are nonnegative in all cases.

We shall now illustrate by means of a numerical example, as usual with an instructional purpose, how the simplex table defined earlier is decomposed into submatrices. Let us take the program

$$(16.20)\qquad \begin{aligned} &(1)\quad [\text{MAX}]\ g = x_1+3x_2,\\ &(2)\quad -x_1+x_2 \leqslant 3,\\ &(3)\quad x_1+2x_2 \leqslant 18,\\ &\qquad\ \ x_1,\, x_2 \geqslant 0. \end{aligned}$$

This program is shown in Fig. 16.1.

If we add the deviation variables u_1 and u_2 to the linear program of (16.20), constraints (2) and (3) become

$$(16.21)\qquad \begin{aligned} &(2')\quad -x_1+x_2+u_1 = 3,\\ &(3')\quad x_1+2x_2+u_2 = 18. \end{aligned}$$

For the sake of an example, let us show how we obtain the simplex table (16.8) in its matrical form relative to the extreme point P_3 of the polyhedron

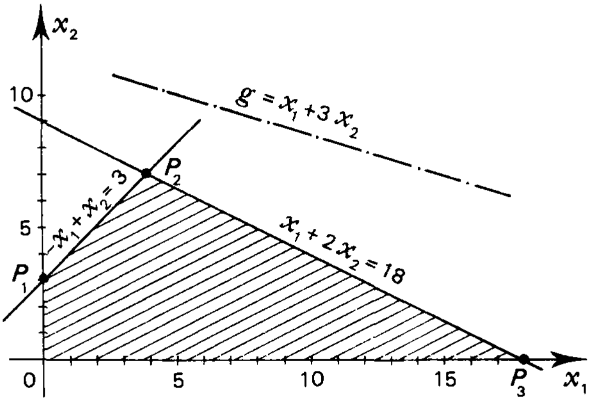


FIG. 16.1

of the constraints. At this point we have

$$x_2 = u_2 = 0\,. \tag{16.22}$$

The nonnull variables of the basis are x_1 and u_1; the corresponding basis matrix $[B]_{2\times 2}$ is

$$[B] = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}, \tag{16.23}$$

and the corresponding vectors $[x_B]_{2\times 1}$, $[x_N]_{2\times 1}$, $[c_B]_{2\times 1}$, $[c_N]_{2\times 1}$ are

$$[x_B] = \begin{bmatrix} x_1 \\ u_1 \end{bmatrix}, \tag{16.24}$$

$$[x_N] = \begin{bmatrix} x_2 \\ u_2 \end{bmatrix}, \tag{16.25}$$

$$[c_B] = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{16.26}$$

$$[c_N] = \begin{bmatrix} 3 \\ 0 \end{bmatrix}. \tag{16.27}$$

Whence, for this program, the corresponding initial program (16.5) is

(16.28)

$$\begin{bmatrix} [1] & -\overbrace{[\ 1 \quad 0]}^{-[c_B]'} & -\overbrace{[3 \quad 0]}^{-[c_N]'} & [0 \quad 0] \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \underbrace{\begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}}_{B} & \underbrace{\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}}_{N} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{bmatrix} \cdot \begin{bmatrix} [g] \\ \begin{bmatrix} x_1 \\ u_1 \end{bmatrix} \\ \begin{bmatrix} x_2 \\ u_2 \end{bmatrix} \\ \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} [0] \\ \begin{bmatrix} 3 \\ 18 \end{bmatrix} \end{bmatrix} \Big\} b.$$

We easily obtain

$$(16.29) \qquad [B]^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix},$$

and thence the simplex table corresponding to basis $[B]$, that is to say, to the extremity P_3.

(16.30)

$$\begin{bmatrix} [1] & [0 \quad 0] & -[3 \quad 0]+[1 \quad 0].\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}\cdot\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} & [1 \quad 0].\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}\cdot\begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} & \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \end{bmatrix}$$

$$\cdot \begin{bmatrix} [g] \\ \begin{bmatrix} x_1 \\ u_1 \end{bmatrix} \\ \begin{bmatrix} x_2 \\ u_2 \end{bmatrix} \\ \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} [1 \quad 0].\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}\cdot\begin{bmatrix} 3 \\ 18 \end{bmatrix} \\ \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}\cdot\begin{bmatrix} 3 \\ 18 \end{bmatrix} \end{bmatrix},$$

which gives, after all the calculations have been made,

$$\begin{bmatrix} [1] & [0\ \ 0] & [-1\ \ 1] & [0\ \ 1] \\ \begin{bmatrix}0\\0\end{bmatrix} & \begin{bmatrix}1&0\\0&1\end{bmatrix} & \begin{bmatrix}2&1\\3&1\end{bmatrix} & \begin{bmatrix}0&1\\1&1\end{bmatrix}\end{bmatrix} \cdot \begin{bmatrix}[g]\\ \begin{bmatrix}x_1\\u_1\end{bmatrix}\\ \begin{bmatrix}x_2\\u_2\end{bmatrix}\\ \begin{bmatrix}\varphi_1\\\varphi_2\end{bmatrix}\end{bmatrix} = \begin{bmatrix}18\\18\\21\end{bmatrix}. \tag{16.31}$$

We verify that, for

$$x_2 = u_2 = \varphi_1 = \varphi_2 = 0, \tag{16.32}$$

table (16.31) gives

$$x_1 = 18, \qquad u_1 = 21; \qquad g = 18. \tag{16.33}$$

Now let us consider the dual program of (16.2):

$$\begin{aligned} &(1)\quad [\mathrm{MIN}]\, f = [b]'_{1\times m}\cdot[y]_{m\times 1}, \\ &(2)\quad [a]'_{n\times m}\cdot[y]_{m\times 1} \geqslant [c]_{n\times 1}, \\ &(3)\quad [y]\in \mathbf{R}^m, \qquad [y]_{m\times 1} \geqslant [0]_{m\times 1}. \end{aligned} \tag{16.34}$$

Let $[\hat{y}]_{m\times 1}$ represent an optimal solution of (16.34). We shall prove that

$$[\hat{y}]_{\mathrm{m}\times 1} = ([B]^{-1}_{m\times m})'\cdot[c_B]_{m\times 1}. \tag{16.35}$$

The right member appears in the form of its transpose in (16.8) above and to the right of the left member; it is also the vector of the marginal costs of the m artificial variables φ_i.

First of all,

$$([B]^{-1})'_{m\times m}\cdot[c_B]_{m\times 1} \geqslant [0]_{m\times 1}, \tag{16.36}$$

in accordance with (16.12).

Again, (16.34) can be expressed as follows by introducing the matrix of the deviation variables $[v]_{n\times 1}$:

$$\begin{aligned} &(1)\quad [\mathrm{MIN}]\, f = [b]'_{1\times m}\cdot[y]_{m\times 1}, \\ &(2)\quad [a]'_{n\times m}\cdot[y]_{m\times 1} - [1]_{n\times n}\cdot[v]_{n\times 1} = [c]_{n\times 1}, \\ &(3)\quad [y]_{m\times 1}\in \mathbf{R}^m, \qquad [v]_{n\times 1}\in \mathbf{R}^n, \\ &\qquad\ [y]_{m\times 1} \geqslant [0]_{m\times 1}, \qquad [v]_{n\times 1} \geqslant [0]_{n\times 1}. \end{aligned} \tag{16.37}$$

We recall how we obtained $[B]_{m\times m}$ and $[N]_{m\times n}$ by a suitable choice of columns in $[[a]\ [I]]_{m\times(m+n)}$ (see what was done immediately before (16.16)).

Let us then calculate

(16.38)

$$\begin{bmatrix}[B]' \\ [N]'\end{bmatrix}_{(n+\mathrm{m})\times m} \cdot([B]^{-1})'_{m\times m}\cdot[c_B]_{m\times 1} = \begin{bmatrix}[c_B] \\ [N]'\cdot([B]^{-1})'\cdot[c_B]\end{bmatrix}_{(n+m)\times 1} .$$

In addition, by the use of (16.11) and by taking the transpose, we can say

$$(16.39)\qquad [N]'_{n\times m}\cdot([B]^{-1})'_{m\times m}\cdot[c_B]_{m\times 1} \geqslant [c_N]_{n\times 1} .$$

By combining (16.39) with (16.38) we obtain $(m+n)$ inequalities

$$(16.40)\qquad \begin{bmatrix}[B]' \\ [N]'\end{bmatrix}_{(n\times m)\times m}\cdot[\hat{y}]_{m\times 1} \geqslant \begin{bmatrix}[c_B] \\ [c_N]\end{bmatrix}_{(m+n)\times 1} ,$$

from which we can again obtain m inequalities that we rearrange, recalling that $[[B]\,[N]]$ is obtained by a permutation of the columns of matrix $[[a]\ [I]]$, and $[[c_B]\ [c_N]]$ by the same permutation of the vector $[[c]\ [0]]$. We obtain

$$(16.41)\qquad [a]'\cdot[\hat{y}] \geqslant [c] .$$

This shows that, in accordance with (16.34) and (16.36), $[\hat{Y}]_{m\times 1}$, obtained from (16.35) is a possible solution of the dual program.

In addition,

$$\begin{aligned}(16.42)^1\qquad \min f = [b]'_{1\times m}\cdot[\hat{y}]_{m\times 1} &= [b]'_{1\times m}\cdot([B]^{-1})'_{m\times m}\cdot[c_B]_{m\times 1} \\ &= ([c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[b])'_{m\times 1} \\ &= ([c_B]'\cdot[\hat{x}_B])' \\ &\quad \text{(taking the transpose of the result),}\end{aligned}$$

where $[\hat{x}_B]$ is the vector of the optimal base variables of program (16.1) in accordance with Theorem 16.I. But

$$\begin{aligned}(16.43)^1\qquad ([c_B]'\cdot[\hat{x}_B])' &= [c_B]'\cdot[\hat{x}_B] \\ &= \max([c]'\cdot[x]) = \max g .\end{aligned}$$

Again, let us consider any possible solutions $[X]_{n\times 1}$ of program (16.1) and

[1] Let us recall that the symbol [MAX] Φ indicates the search for the maximum of Φ, whereas max Φ means that we are concerned with the maximum itself. The same notation applies to [MIN] and min.

$[Y]_{m\times 1}$ of (16.34). We can say

$$[b]'_{1\times m}\cdot[y]_{m\times 1} \geqslant ([a]_{m\times n}\cdot[x]_{n\times 1})'\cdot[y]_{m\times 1} \quad \text{from (16.2)}. \tag{16.44}$$

Or again,

$$[b]'_{1\times m}\cdot[y]_{m\times 1} \geqslant [x]'_{1\times n}\cdot[a]'_{n\times m}\cdot[y]_{m\times 1}\,. \tag{16.45}$$

But, from (16.34), we have

$$[a]'_{n\times m}\cdot[y]_{m\times 1} \geqslant [c]_{n\times 1}\,. \tag{16.46}$$

Combining (16.46) with (16.45) we obtain

$$[b]'_{1\times m}\cdot[y]_{m\times 1} \geqslant [x]'_{1\times n}\cdot[c]_{n\times 1}\,; \tag{16.47}$$

that is to say again,

$$f \geqslant g\,. \tag{16.48}$$

Two solutions $([\hat{x}]_{n\times 1}, [\hat{y}]_{m\times 1})$, where $[\hat{x}]$ is the solution of program (16.1) and $[\hat{y}]$ is that of (16.34), are such that

$$f = g\,, \tag{16.49}$$

in accordance with (16.42) and (16.43).

Without making use on this occasion of Theorem 16.I, we have now proved that $[\hat{x}]$ is an optimal solution of program (16.1), since function g is always less than f in accordance with (16.48) and is only equal to f for this point $[\hat{x}]$ (see (16.49)).

Observations

For every vector $[x]_{n\times 1}$ and for every vector $[y]_{m\times 1}$, and in particular for $[\hat{x}]_{n\times 1}$ and $[\hat{y}]_{m\times 1}$, we have

$$[b]_{m\times 1} - [a]_{m\times n}\cdot[x]_{n\times 1} \geqslant [0]_{m\times 1}\,, \tag{16.50}$$

$$[a]'_{n\times m}\cdot[y]_{m\times 1} - [c]_{n\times 1} \geqslant [0]_{n\times 1}\,. \tag{16.51}$$

Since the vectors $[x]$ and $[y]$ are nonnegative, we can state

$$([b] - [a]\cdot[x])'_{1\times m}\cdot[y]_{m\times 1} \geqslant 0 \tag{16.52}$$

and

$$([a]'\cdot[y] - [c])'_{1\times n}\cdot[x]_{n\times 1} \geqslant 0\,. \tag{16.53}$$

Hence we still have

$$([b]-[a]\cdot[x])'_{1\times m}\cdot[y]_{m\times 1} + ([a]'\cdot[y]-[c]')_{1\times n}\cdot[x]_{n\times 1} \geqslant 0\,. \tag{16.54}$$

But, as we shall now show, the expression (16.54) is identically null for $[x] = [\hat{x}]$ and $[y] = [\hat{y}]$.

By expanding (16.54), it follows that

$$(16.55)\qquad [b]'_{1\times m}\cdot[\hat{y}]_{m\times 1} - [\hat{x}]'_{1\times n}\cdot[a]'_{n\times m}\cdot[\hat{y}]_{m\times 1} + [\hat{y}]'_{1\times m}\cdot[a]_{m\times n}\cdot[\hat{x}]_{n\times 1} - [c]'_{1\times n}\cdot[\hat{x}]_{n\times 1} \geqslant 0.$$

Let us observe that

$$(16.56)\qquad [\hat{x}]'_{1\times n}\cdot[a]'_{n\times m}\cdot[\hat{y}]_{m\times 1} = [\hat{y}]'_{1\times m}\cdot[a]_{m\times n}\cdot[\hat{x}]_{n\times 1}$$

and that

$$(16.57)\qquad [b]'_{1\times m}\cdot[\hat{y}]_{m\times 1} = \min f = \max g = [c]_{1\times n}\cdot[\hat{x}]_{n\times 1}.$$

Hence the left member of (16.55) is equal to 0. This enables us to prove Theorem 16.II that follows.

Theorem 16.II

This theorem is generally known as the *fundamental theorem of duality.*

For a primal-dual optimal pair ($[\hat{x}]$, $[\hat{y}]$), we have

$$(16.58)\qquad ([b]-[a]\cdot[\hat{x}])'_{1\times m}\cdot[\hat{y}]_{m\times 1} = 0$$

and

$$(16.59)\qquad ([a]'\cdot[\hat{y}]-[c])'_{1\times n}\cdot[\hat{x}]_{n\times 1} = 0.$$

Proof

The expression (16.55), that is identically null for $[x] = [\hat{x}]$ and $[y] = [\hat{y}]$, is the sum of the two nonnegative terms (16.52) and (16.53) combined in (16.54). These two terms must be nonnull, whence we have (16.58) and (16.59).

We shall now interpret this theorem so as to give the reader a better understanding of these important properties than we provided in Section 66 of the first volume.

Let $\hat{x}_i$ be a variable of positive basis in the primal program. To satisfy (16.59) it is necessary that

$$(16.60)\qquad ([a]^i)'_{1\times m}\cdot[\hat{y}]_{m\times 1} - c_i = 0,$$

where $[a]^i_{m\times 1}$ is the ith column of the matrix $[a]$. This means that the ith constraint of the dual program (16.34) is strictly verified for the optimum.

Symmetrically, if the jth constraint of the primal program (16.1) is not strictly verified for the optimum, that is to say, if

$$(16.61)\qquad b_j - ([a]_j)_{1\times n}\cdot[\hat{x}]_{n\times 1} > 0,$$

where $([a]_j)_{1\times n}$ is the jth line of $[a]_{m\times n}$, then, to have (16.58) it means that, if $\hat{y}_j$ is the jth variable of the dual program (16.34), $\hat{y}_j = 0$.

The above properties, often called the *properties of complementation of* *primal and dual solutions* are summarized as follows:

(16.62) $\hat{x}_i$ is a basic variable of the primal $\Rightarrow$
$$([a]^i)'_{1\times m}\cdot[\hat{y}]_{m\times 1} - c_i = 0.$$

(16.63) $\hat{x}_i$ is not a basic variable of the primal $\Rightarrow$
$$([a]^i)'_{1\times m}\cdot[\hat{y}]_{m\times 1} - c_i \geqslant 0.$$

(16.64) $b_j - ([a]_j)_{1\times n}\cdot[\hat{x}]_{n\times 1} > 0 \Rightarrow$
$\hat{y}_j = 0$ and $\hat{y}_j$ is not a basic variable of the dual.

(16.65) $b_j - ([a]_j)_{1\times n}\cdot[\hat{x}]_{n\times 1} = 0 \Rightarrow$
$\hat{y}_j \geqslant 0$ and $\hat{y}_j$ is a basic variable of the dual.

2. Dual-Simplex Method [K58]

Let us summarize what we have proved by taking the two following programs, each of which is the dual of the other:

(16.66)
(1) $[\text{MAX}]\ g = [c]'_{1\times n}\cdot[x]_{n\times 1}$,
(2) $[a]_{m\times n}\cdot[x]_{n\times 1} \leqslant [b]_{m\times 1}$,
(3) $[x]_{n\times 1} \in \mathbf{R}^n$, $[x]_{n\times 1} \geqslant [0]_{n\times 1}$.

and

(16.67)
(1) $[\text{MIN}]\ f = [b]'_{1\times m}\cdot[y]_{m\times 1}$.,
(2) $[a]'_{n\times m}\cdot[y]_{m\times 1} \geqslant [c]_{n\times 1}$,
(3) $[y]_{m\times 1} \in \mathbf{R}^m$, $[y]_{m\times 1} \geqslant [0]_{m\times 1}$;

then g will be maximal and f minimal if the three following conditions are all satisfied in table (16.8):

(16.68) $$[B]^{-1}_{m\times m}\cdot[b]_{m\times 1} \geqslant [0]_{m\times 1},$$

(16.69) $$[c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[N]_{m\times n} - [c_N]'_{1\times n} \geqslant [0]_{1\times n},$$

(16.70) $$[c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m} \geqslant [0]_{1\times m}.$$

We shall now utilize these results in a method originated by Lemke [K58] and called the *dual-simplex method.*

In the simplex method explained in Volume 1 the optimal solution of the primal problem is reached by a path from extreme points to adjacent ones, that is to say, that the right members of the different tables (16.8) obtained during the iterations are nonnegative. By contrast, in the dual-simplex method this right member may be nonpositive (that is, it may include certain negative components), but the first line of the rectangular matrices of (16.8) must remain nonnegative, that is to say, if we will refer to (16.8), that we must have

$$(16.71) \qquad -[c_N]'_{1\times n} + [c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[N]_{m\times n} \geqslant [0]_{1\times n}$$

and

$$(16.72) \qquad [c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m} \geqslant [0]_{1\times m}.$$

With the dual-simplex method the right member of (16.8) will only be nonnegative for the optimum, whereas, in the classic simplex method (sometimes by analogy referred to as the *primal-simplex method*), (16.71) and (16.72) are only satisfied for the optimum.

Let us assume

$$(16.73) \qquad \bar{a}_{ij} = ([B]_i^{-1})_{1\times m}\cdot([N]^j)_{m\times 1},$$

where $[B]_i^{-1}$ is the ith line of matrix $[B]^{-1}$ and $[N]^j$ is the jth column of $[N]$.

Let us also assume

(16.74)

$$[\bar{c}]_{1\times(2m+n+1)} = [[1]_{1\times 1}\cdot[0]_{1\times m} \quad -[c_N]'_{1\times n} + [c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[N]_{m\times n} \\ \cdot[c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}]$$

and take $\bar{c}_j$ for the jth element of $[\bar{c}]_{1\times(2m+n+1)}$.

We shall now explain the dual-simplex method.

We assume that table (16.8) has been obtained after k iterations and (16.71) and (16.72) are satisfied. In this case, we say that the table provides a solution for the dual program (16.34) since, by taking

$$(16.75) \qquad [y]_{m\times 1} = ([B]^{-1})'_{m\times m}\cdot[c_B]_{m\times 1} \geqslant [0]_{m\times 1},$$

all the dual constraints are satisfied, as was proved in (16.41).

Let us now introduce a vector

$$(16.76) \qquad [\bar{b}]_{m\times 1} = [B]^{-1}_{m\times m}\cdot[b]_{m\times 1}.$$

If $[\bar{b}] \geqslant [0]$, then the table provides a solution,

$$(16.77) \qquad [x]_{m\times 1} = [\bar{b}]_{m\times 1}$$

of the primal program (16.1). In this case, all the conditions of optimality of (16.68)–(16.70) are satisfied. We have obtained the pair of optimal solutions $[\hat{x}]$, $[\hat{y}]$ of the primal and dual programs, and we now end the procedure.

Let us now suppose that the condition

$$(16.78) \qquad [\bar{b}]_{m\times 1} \geqslant [0]_{m\times 1}$$

is not satisfied. In that case, we shall carry out operations known as *pivoting* that are in all respects similar to those employed in Section 59 of Volume 1. The sole difference will consist in choosing the pivotal element to maintain the first line as nonnegative, whereas in the classic simplex method the aim was to keep the second member nonnegative.

Let r be an index such that

(16.79) $\quad \bar{b}_r < 0,$

where $\bar{b}_r$ is the element of the rth line of $[\bar{b}]$. In practice, we shall take the element $\bar{b}_r$ that is the most negative of the negative elements of $[\bar{b}]$.

Now, let s be the index such that

(16.80) $\quad (1)\ \bar{a}_{rs} < 0,$

where $\bar{a}_{rs}$ is defined by means of (16.73)

(16.81) $\quad (2)\ \underset{j}{\text{MIN}} \dfrac{\bar{c}_j}{-\bar{a}_{rj}} = \dfrac{\bar{c}_s}{-\bar{a}_{rs}}$

where $\min_j$ is selected from the j's such that $\bar{a}_{rj} < 0$.

Let us recall that the φ_i, $i = 1, 2, \ldots, m$, are artificial variables that must not enter the basis. As a result, the choice of r in (16.80) must be made from the nonartificial variables, that is, $r = 1, 2, \ldots, m+n+1$ (instead of $2m+n+1$). Similarly in (16.81), j cannot be chosen from the artificial variables and is selected so that $a_{rj} < 0$ with $j = 1, 2, \ldots, m+n+1$,

Choosing $\bar{a}_{rs}$ as a pivot, we then obtain

(16.82) $\quad \bar{b}_i^* = \bar{b}_i - \bar{a}_{is}.\bar{b}_r/\bar{a}_{rs}, \qquad i \neq r, \qquad i = 1, 2, \ldots, m;$

(16.83) $\quad \bar{b}_r^* = \bar{b}_r/\bar{a}_{rs};$

(16.84) $\quad \bar{a}_{ij}^* = \bar{a}_{ij} - \bar{a}_{is}.\bar{a}_{rj}/\bar{a}_{rs}, \qquad i \neq r, \qquad i = 1, 2, \ldots, m, \quad j = 1, 2, \ldots, 2m+n+1;$

(16.85) $\quad \bar{a}_{rj}^* = \bar{a}_{rj}/\bar{a}_{rs};$

(16.86) $\quad \bar{c}_j^* = \bar{c}_j - \bar{c}_s.\bar{a}_{rj}/\bar{a}_{rs}, \qquad j = 1, 2, \ldots, 2m+n+1.$

We can at once verify that the choice of the above pivot $\bar{a}_{rs}$, as shown by (16.80) and (16.81), results in

$$\bar{c}_j^* \geqslant 0, \qquad j = 1, 2, \ldots, 2m+n+1.$$

The new table that is obtained also corresponds to a solution of the dual problem that we have defined as beloning to a table for which $[\bar{c}]_{1\times(2m+n+1)} \geqslant [0]_{1\times(2m+n+1)}$ by taking the $\bar{c}_j^*$ as elements of $[\bar{c}]$.

This method of procedure justifies its name of *dual-simplex.*

There remains the case where a negative $\bar{a}_{rs}$ cannot be found, and this will be considered in Section 16.3.

Let us now consider an example that requires for its treatment the use of the results of Section 15.5 where the search for the maximum is transposed for the minimum and conversely.

Let us solve the program

$$(16.87)\qquad \begin{aligned} &(1)\quad [\mathrm{MIN}]\ z = 2x_2+3x_3\,,\\ &(2)\quad x_1-3x_2 \geqslant -4\,,\\ &(3)\quad x_2+x_3 \geqslant 3\,,\\ &\qquad\ \ x_1\,, x_2\,, x_3 \geqslant 0\,. \end{aligned}$$

Let us transform this into a program for maximization. Let us suppose

$$(16.88)\qquad g = -z\,.$$

It follows that

$$(16.89)\qquad [\mathrm{MIN}]\ z = -[\mathrm{MAX}]\ g\,,$$

that is,

$$(16.90)\qquad \begin{aligned} &(1)\quad [\mathrm{MAX}]\ g = -2x_2-3x_3\,,\\ &(2)\quad x_1-3x_2 \geqslant -4\,,\\ &(3)\quad x_2+x_3 \geqslant 3\,,\\ &(4)\quad x_1\,, x_2\,, x_3 \geqslant 0\,. \end{aligned}$$

Let us express this program in its standard form (16.5) after adding the deviation variables u_1 and u_2. We shall take x_ι and u_2 as the initial basic variables.

(16.91)

$$\begin{bmatrix} [1] & -[0\ \ 0] & -[-2\ \ -3\ \ \ 0] & [0\ \ 0] \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & \begin{bmatrix} -3 & 0 & -1 \\ -1 & -1 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{bmatrix} \cdot \begin{bmatrix} [g] \\ \begin{bmatrix} x_1 \\ u_2 \end{bmatrix} \\ \begin{bmatrix} x_2 \\ x_3 \\ u_1 \end{bmatrix} \\ \begin{bmatrix} \varphi_1 \\ \varphi_2 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} [0] \\ \begin{bmatrix} -4 \\ -3 \end{bmatrix} \end{bmatrix}.$$

For convenience, the right member of (16.91) will be shown as the first column of a table and the column of the variables will be given as a line above the table. In a similar way the basic variables are shown to the left of the table so that we can recall which we must use in the iteration we are considering.

(16.92)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
			g	x_1	u_2	x_2	x_3	u_1	φ_1	φ_2
(0)	g	0	1	0	0	2	3	0	0	0
(1)	x_1	-4	0	1	0	-3	0	$\textcircled{-1}$	1	0
(2)	u_2	-3	0	0	1	-1	-1	0	0	1

← This line represents $[\bar{c}]$

↑ Column giving the point $x_1 = -4$, $x_2 = -3$, $g = 0$.

Let us observe that line (0) of (16.92) is nonnegative as long as the conditions of (16.71) and (16.72) are satisfied. Hence we can commence dual iterations.

We have

$$[\bar{b}] = \begin{bmatrix} -4 \\ -3 \end{bmatrix}. \tag{16.93}$$

So $x_1 = -4$ and $u_2 = -3$ does not provide a solution of the primal program (16.90). We shall now perform a dual iteration taking line (1) as the line for finding a pivot, since -4 is more negative than -3. Let us next look for the column of this pivot. In line (1) the elements $\bar{a}_{14} = -3$ (column (4)) and $\bar{a}_{16} = -1$ (column (6)) are candidates since they are negative. If we look for the pivot with the help of (16.81), we find

$$\min\left(\frac{\bar{c}_4}{-\bar{a}_{14}}, \frac{\bar{c}_6}{-\bar{a}_{16}}\right) = \min(2/3, 0/1) = 0, \tag{16.94}$$

and $\bar{a}_{16}$ will be chosen, being circled in table (16.92).

By applying rules (16.82)–(16.86) we obtain the following new table where x_1 leaves the basis and u_1 enters it.

(16.95)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
			g	x_1	u_2	x_2	x_3	u_1	φ_1	φ_2
(0)	g	0	1	0	0	2	3	0	0	0
(1)	u_1	4	0	-1	0	3	0	1	-1	0
(2)	u_2	-3	0	0	1	$\textcircled{-1}$	-1	0	0	1

↑ Column giving the point $u_1 = 4$, $u_2 = -3$, $g = 0$.

The point $u_1 = 4$, $u_2 = -3$, $g = 0$ is not a solution. We shall take line (2) as the pivot line as it contains -3 and, by applying rule (16.81), we find that $\bar{a}_{24} = -1$ must be chosen as the pivot. Table (16.96) and (16.97) are given below, and the reader can find the results as an exercise.

(16.96)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
			g	x_1	u_2	x_2	x_3	u_1	φ_1	φ_2
(0)	g	-6	1	0	2	0	1	0	0	2
(1)	u_1	-5	0	$\textcircled{-1}$	3	0	-3	1	-1	3
(2)	x_2	3	0	0	-1	1	1	0	0	-1

↑

Column giving the point $u_1 = -5$, $x_2 = 3$, $g = -6$.

(16.97)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
			g	x_1	u_2	x_2	x_3	u_1	φ_1	φ_2
(0)	g	-6	1	0	2	0	1	0	0	2
(1)	x_1	5	0	1	-3	0	3	-1	1	-3
(2)	x_2	3	0	0	-1	1	1	0	0	-1

↑

Column giving the point $x_1 = 5$, $x_2 = 3$, $g = -6$.

Table (16.97) shows the presence of a solution

(16.98) $$x_1 = 5, \quad x_2 = 3, \quad g = -6,$$

which is the optimal solution of (16.90) in accordance with what we have just proved.

Finally let us return to the initial program (16.87) where we have

(16.99) $$\min z = -\max g = 6$$

(16.100) $$x_1 = 5, \quad x_2 = 3, \quad x_3 = 0, \quad u_1 = 0, \quad u_2 = 0.$$

The reader can check that the vector of the marginal costs of the artificial variables φ_1 and φ_2, which is [0 2], appearing on the right of line (0) in (16.97), is the optimum for the dual program of (16.90), namely,

(16.101)
(1) $[\text{MIN}]\, f = 4y_1 - 3y_2$,

(2) $-y_1 \geqslant 0$,

(3) $3y_1 - y_2 \geqslant -2$,

$$(4)\quad -y_2 \geqslant -3,$$

$$(5)\quad y_1, y_2 \geqslant 0.$$

The solution of this dual program is

$$(16.102)\qquad y_1 = 0, \qquad y_2 = 2, \qquad \min f = -6.$$

To obtain (16.101) the reader should transpose the inequalities of (16.90); this will give the program

$$(16.103)\qquad \begin{aligned} &(1)\quad [\text{MAX}]\ g = -2x_2 - 3x_3, \\ &(2)\quad -x_1 + 3x_2 \leqslant 4, \\ &(3)\quad -x_2 - x_3 \leqslant -3, \\ &(4)\quad x_1, x_2, x_3 \geqslant 0. \end{aligned}$$

3. Observations on the Impossible Case for the Primal Problem

We shall now examine the case where, using the dual-simplex method, it is not possible to find in the line of index r such that $\bar{b}_r < 0$, an element $\bar{a}_{rs} < 0$. Our reason for dwelling on this aspect here is that we shall make use of it in Section 21 when considering Benders's method.

To prove that there is no primal solution in this case we shall remove the rth line from table (16.8) and write the equivalent equation.

$$(16.104)\qquad \begin{aligned} 0.g + 1.x_{B_r} &+ ([B]_r^{-1})_{1\times m}.[N]_{m\times n}.[x_N]_{n\times 1} \\ &+ ([B]_r^{-1})_{1\times m}.[\varphi]_{m\times 1} = ([B]_r^{-1})_{1\times m}.[b]_{m\times 1} \end{aligned}$$

where x_{B_r} is the basis variable with index r of the vector $[x_B]$ and $[B]_r^{-1}$ is the rth line of $[B]^{-1}$.

By observing that $[\varphi]$ must be identically null, since the variables φ_i cannot enter the basis, and by using the notation of (16.73), (16.79), and (16.80), Eq. (16.104) is expressed

$$(16.105)\qquad x_{B_r} = \bar{b}_r - \sum_{j=m+1}^{m+n+1} \bar{a}_{rj}.x_{N_j},$$

where x_{N_j} is the variable outside the basis of index j of the vector $[x_N]_{n\times 1}$.

If all the $\bar{a}_{rj} > 0$, the x_{N_j} being positive or null and $\bar{b}_r$ being negative, it follows that x_{B_r} can never be nonnegative; hence the primal program (16.66) has no solution since we can only obtain $x_{B_r} \geqslant 0$.

We shall now show that if we encounter the case where, beginning with table (16.8) corresponding to the basis $[B]$, there is no solution of the primal program (16.66), this means that the convex polyhedron of the constraints of the dual program (16.67) possesses a ray (see the definition on page 204) the direction of which can be given by equation (16.104). Let us therefore enunciate a theorem.

Theorem 16.III

The vector $[B]_r^{-1}$, which, we should recall, represents the rth line of $[B]^{-1}$, is a direction of a ray of the dual-convex polyhedron expressed by the constraints (2) and (3) of (16.67), provided $\bar{b}_r < 0$ and provided the primal program is impossible.

Proof

Let us consider table (16.8) and state,

$$(16.107)^1 \qquad [y^*]_{m\times 1} = ([B]^{-1})'_{m\times m}\cdot[c_B]_{m\times 1}\,.$$

This point of $\mathbf{R}^m$ is a solution of the dual program (16.67).

Let us assume

$$(16.108) \qquad [V]_{m\times 1} = ([B]_r^{-1})'_{m\times 1}\,.$$

If $[y]_{m\times 1}$ is any point of the half-line of $\mathbf{R}^m$ that passes through the point $[y^*]_{m\times 1}$ having the direction of the vector $[V]_{m\times 1}$, we can then say,

$$(16.109) \qquad [y]_{m\times 1} = [y^*]_{m\times 1} + \theta.[V]_{m\times 1}, \qquad \theta \geqslant 0,$$

where θ is therefore a nonnegative scalar.

We shall now show that all the points on the half-line (16.109) belong to the dual-convex polyhedron formed by constraints (2) and (3) of (16.67).

Before proceeding further with the proof, let us observe that

$$(16.110) \qquad \begin{aligned}[B]'_{m\times m}\cdot[V]_{m\times 1} &= [B]'_{m\times m}\cdot([B]_r^{-1})'_{m\times 1}\\ &= ([B]_r^{-1}\cdot[B])_{m\times 1}\\ &= \begin{bmatrix}0\\0\\ \vdots\\1\\ \vdots\\0\end{bmatrix}_{m\times 1} \quad (1 \text{ in position } r).\end{aligned}$$

Let us then assume

$$(16.111) \qquad [e_r] = \begin{bmatrix}0\\0\\ \vdots\\1\\ \vdots\\0\end{bmatrix}_{m\times 1} \quad (1 \text{ in position } r).$$

[1] Equation number (16.106) omitted from the French edition.

By employing a method similar to (16.38) we obtain

(16.112)

$$\begin{bmatrix} [B]' \\ [N]' \end{bmatrix}_{(m+n)\times m} \cdot [y]_{m\times 1} = \begin{bmatrix} [B]'_{m\times m} \\ [N]'_{n\times m} \end{bmatrix}_{(m+n)\times m} \cdot ([y^*]_{m\times 1} + \theta . [V]_{m\times 1}).$$

By substituting (16.107) in the above and expanding it, we obtain

$$(16.113)\qquad \begin{bmatrix} [B]' \\ [N]' \end{bmatrix}_{(m+n)\times m} \cdot [y]_{m\times 1}$$

$$= \begin{bmatrix} [c_B]_{m\times 1} + \theta . [B]'_{m\times m} \cdot [V]_{m\times 1} \\ [N]'_{n\times m} \cdot [B]^{-1}_{m\times m} \cdot [c_B]_{m\times 1} + \theta . [N]'_{n\times m} \cdot [V]_{m\times 1} \end{bmatrix}_{(m+n)\times 1}$$

By using (16.110), (16.111), and (16.71) we obtain

$$(16.114)\qquad \begin{bmatrix} [B]' \\ [N]' \end{bmatrix}_{(m+n)\times m} \cdot [y]_{m\times 1} \geqslant \begin{bmatrix} [c_B]_{m\times 1} \\ [c_N]_{n\times 1} \end{bmatrix}_{(m+n)\times 1} + \theta \begin{bmatrix} [e_r]_{m\times 1} \\ [N]'_{n\times m} \cdot [V]_{m\times 1} \end{bmatrix}_{(m+n)\times 1}$$

Thus the primal program (16.66) has no solution beginning with table (16.8), since $\bar{b}_r < 0$ and $a_{r_j} \geqslant 0$ for $j = (m+1), \ldots, (m+n+1)$. As a result we can say

$$(16.115)\qquad [N]' . [V] = [N]' . ([B]_r^{-1})' = \begin{bmatrix} \bar{a}_{rm+1} \\ \bar{a}_{rm+2} \\ \vdots \\ \bar{a}_{rm+n+1} \end{bmatrix} \geqslant \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} .$$

Hence, since

(16.116) $\quad [e_r]_{m\times 1} \geqslant [0]_{m\times 1}$ (from 16.111),

(16.117) $\quad [N]'_{n\times m} \cdot [V]_{m\times 1} \geqslant [0]_{n\times 1}$ (from 16.115),

(16.118) $\quad \theta \geqslant 0$ (from 16.109), we can state by the use of (16.114),

(16.119)

$$\begin{bmatrix}[B]' \\ {[N]'}\end{bmatrix}_{(m+n)\times m}\cdot[y]_{m\times 1} \geqslant \begin{bmatrix}[c_B]_{m\times 1} \\ {[c_N]_{n\times 1}}\end{bmatrix}_{(m+n)\times 1} + \theta\begin{bmatrix}[e_r]_{m\times 1} \\ {[N]'_{n\times m}\cdot[V]_{m\times 1}}\end{bmatrix}_{(m+n)\times 1}$$

$$\geqslant \begin{bmatrix}[c_B]_{m\times 1} \\ {[c_N]_{n\times 1}}\end{bmatrix}_{(m+n)\times 1}.$$

Recalling that matrix $[[B]\,[N]]$ is obtained by a permutation of the columns of matrix $[[a]\,[I]]$, and vector $[[c_B]\,[c_N]]$ by the same permutation of vector $[[c]\,[0]]$, we deduce (from 16.119) that $[a]'.[y] \geqslant [c]$.

Hence $[y]_{m\times 1}$ satisfies the constraints of the dual problem whatever the value of $\theta \geqslant 0$. As a result, the half-line (16.109) of $\mathbf{R}^m$ having the direction of the vector $[V]_{m\times 1}$ is contained in the convex polyhedron of its constraints.

Now, the direction of $[V]_{m\times 1}$ is determined by the intersection of $(m-1)$ hyperplanes that delimit the convex polyhedron of the constraints of the dual program in accordance with the definition of the ray of a convex polyhedron given earlier.

Let us therefore consider the points $[y]$ of the ray (16.109) and let us express the value of the economic function f of the dual program for these different points. We have

(16.120)
$$\begin{aligned} f &= [b]'_{1\times m}\cdot[y]_{m\times 1} \\ &= [b]'_{1\times m}\cdot([B]^{-1}_{m\times m}\cdot[c_B]_{m\times 1} + \theta[V]_{m\times 1}) \\ &= [b]'_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[c_B]_{m\times 1} + \theta.[b]'_{1\times m}\cdot[V]_{m\times 1} \\ &= [b]'_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[c_B]_{m\times 1} + \theta.\bar{b}_r, \end{aligned}$$

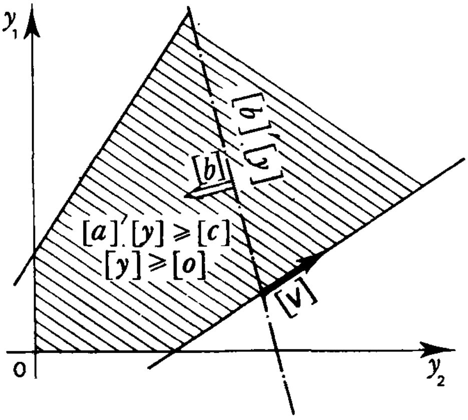


FIG. 16.2

where $\bar{b}_r$ is defined by (16.76) and (16.79).

For $\theta \to \infty$, since $\bar{b}_r < 0$ in accordance with (16.79), $f \to (-\infty)$ for $\theta \to \infty$. And so the dual program allows a minimum that can have as large negative values as we desire.

To sum up, there are four possible cases for the respective solutions of primal and dual programs, as shown in (16.121).

(16.121)

Program of maximization or primal program	Program of minimization or dual program	Observations
max g = min f	min f = max g	Both primal and dual programs have solutions
no solution	min $f \to -\infty$	
max $g \to \infty$	no solution	
no solution	no solution	This case can occur even if it does not have any practical interest

In the case given in which the primal program has no solution, we can, by using the definition of $\bar{b}_r$ (16.79), say

$$\bar{b}_r = ([B]_r^{-1})_{1\times m}\cdot[b]_{m\times 1}, \tag{16.122}$$

which, by the use of definition (16.108) and by transposing (16.122), becomes

$$\bar{b}_r = [b]'_{1\times m}\cdot[V]_{m\times 1} \tag{16.123}$$

Hence, we have

$$[b]'_{1\times m}\cdot[V]_{m\times 1} < 0. \tag{16.124}$$

Let us give a geometrical interpretation of (16.24) in $\mathbf{R}^2$. The slope $[b]'_{1\times m}$ of the economic function of the dual program has a negative scalar product with the direction $[V]_{m\times 1}$ of a ray of the convex polyhedron of the constraints of (16.67). When we minimize the economic function that is shown by a thick arrow, we obtain an infinite value, since there is a direction of ray $[V]$ that has a negative scalar product with this slope.

Section 17. **Programming Method of Dantzig and Manne**

1. Principle of the Method

To understand the principle of this method let us take the following integer program:

$$(17.1)\qquad \begin{array}{ll} (1) & [\mathrm{MIN}]\, f = [c]'_{1\times n}\cdot[x]_{n\times 1}\,, \\ (2) & [a]_{m\times n}\cdot[x]_{n\times 1} \geqslant [b]_{m\times 1}\,, \\ (3) & [x]_{n\times 1} \geqslant [0]_{n\times 1}\,, \\ (4)^1 & [x]_{n\times 1} \in \mathbf{Z}^n. \end{array}$$

The vertices of the cones of $\mathbf{R}^n$ that surround the polyhedron of the constraints are determined by the intersection of n separate hyperplanes taken from the $m+n$ that limit the domain of possible solutions, that is, m hyperplanes given by $[a].[x] = [b]$ and n given by $[x] = [0]$. We shall call

$$(17.2)\qquad s_1\,, s_2\,, \ldots, s_m\,, s_{m+1}\,, \ldots, s_{m+n}\,, \quad s_i \in \mathbf{R}^+\,, \quad i = 1, 2, \ldots, m+n\,,$$

the $m+n$ deviation variables transforming the inequalities (2) and (3) of (17.1) into equations.

Before proceeding further let us demonstrate this by an example intended only to illustrate the procedure:

$$(17.3)\qquad \begin{array}{ll} (1) & [\mathrm{MIN}]\, f = 2x_1 - 3x_2 + 11x_3\,, \\ (2) & x_1 - 2x_2 + 8x_3 \geqslant 10\,, \\ (3) & x_1 + x_2 - 3x_3 \geqslant -2\,, \\ (4) & x_1\,, x_2\,, x_3 \geqslant 0\,, \\ (5) & x_1\,, x_2\,, x_3 \in \mathbf{Z}^3\,. \end{array}$$

By incorporating the deviation variables s_1 to s_5, this program becomes

$$(17.4)\qquad \begin{array}{ll} (1) & [\mathrm{MIN}]\, f = 2x_1 - 3x_2 + 11x_3\,, \\ (2) & x_1 - 2x_2 + 8x_3 - s_1 = 10\,, \\ (3) & x_1 + x_2 - 3x_3 - s_2 = -2\,, \\ (4) & x_1 - s_3 = 0\,, \\ (5) & x_2 - s_4 = 0\,, \end{array}$$

[1] Let us recall that in the usual notation $\mathbf{R}$ is the set of real numbers, $\mathbf{Z}$ that of the related integers, and $\mathbf{N}$ that of the nonnegative natural or integer numbers. $\mathbf{R}^+$ is the set of nonnegative real numbers and $\mathbf{R}_0^*$ that of the positive real numbers.

(6) $x_3 - s_5 = 0,$

(7) $x_1, x_2, x_3 \in \mathbf{N}, \qquad s_1, s_2 \; s_3, s_4, s_5 \in \mathbf{R}^+ .$

Let us assume that the elements of $[a]_{m\times n}$ and $[b]_{m\times 1}$ are integers. Since the variables of $[x]_{n\times 1}$ are integers, the deviation variables of $[s] = [s_1, s_2, \ldots, s_{m\times n}]$ must also be positive or null in accordance with (17.2), so that

$$(17.5) \qquad s_i \in \mathbf{N}, \qquad i = 1, 2, \ldots, m+n .$$

That is $[x]_{n\times 1} \geqslant [0]_{n\times 1}$ which satisfies constraint (2) of (17.1) and is an extreme point of the convex polyhedron defined by

$$(17.6) \qquad [a]_{m\times n} \cdot [x]_{n\times 1} \geqslant [b]_{m\times 1},$$

$$(17.7) \qquad [x]_{n\times 1} \geqslant [0]_{n\times 1} \cdot$$

As we saw earlier, an extreme point in $\mathbf{R}^n$ is the intersection of n hyperplanes in it. Hence, n of the $n+m$ constraints (17.6) and (17.7) are fully satisfied by $[x]$; that is to say, that n of the $n+m$ variables s_i are null. Let $i_1, i_2, \ldots, i_\alpha, \ldots, i_n$ be the indices of these n null s_i variables.

If $[x]$ includes at least one noninteger element, the extreme point $[x]$ is not a solution of (17.1). Hence another point of the convex polyhedron is needed, and for this point at least one of the n inequations (17.6) and (17.7) that were fully satisfied by $[x]$ will no longer be so. At least one of the variables s_{i_α}, $\alpha = 1, 2, \ldots, n$ that were null for this point $[x]$ must be positive. Since the s_i, $i = 1, 2, \ldots, m+n$ are positive or null integers, we must have

$$(17.8) \qquad s_{i_1} + s_{i_2} + \ldots + s_{i_n} \geqslant 1, \qquad s_{i_\alpha} \geqslant 0, \alpha = 1, 2, \ldots, n,$$

that is to say that there is at least one $s_{i_\alpha} > 0$.

The new constraint (17.8) that we shall add to the constraints of the program, if point $[x]_{n\times 1}$ does not belong to $\mathbf{N}^n$, is called the *Dantzig–Manne constraint*. We thus obtain a more constrained linear program that differs from (17.1) but allows the same integer solutions.

The *Dantzig–Manne method* uses the simplex procedure to solve this new program by removing constraint (4) of (17.1) and by including constraint (17.8), and so on.

However, this method, of great historical importance in the solution of integer programs, since it was the first introduced for this purpose in 1956, has now been discarded for others, mainly because the algorithm attached to it does not always lead to convergence.

Nevertheless, the two necessary (but unfortunately insufficient) conditions have been proved by R. E. Gomory and A. J. Hoffmann [K43], and we shall now examine them.

For a point $[x]_{n\times 1}$, the solution of (17.6) and (17.7), there are $m+n$ deviation variables depending on $[x]$, that is to say, for an $[x]$ that satisfies the above conditions; they can be called $s_i[x]$, $i=1,2,\ldots,m+n$, defining in this way their dependence on $[x]$.

In addition, for each new constraint such as (17.8) we must also add a deviation variable depending on $[x]$; we shall use the notation[1] $t_p[x]$, $p=1,2,\ldots$ for the deviation variable of the pth Dantzig–Manne constraint that we can therefore express as

$$(17.9) \qquad s_{i_1}^{(p)}[x] + s_{i_2}^{(p)}[x] + \ldots + s_{i_n}^{(p)}[x] - t_p[x] = 1\,.$$

The upper index (p) in $s_{in}^{(p)}$ shows that it is the Dantzig–Manne constraint introduced at the pth iteration.

2. Conditions Needed for Convergence

Let us take a point $[x]_{n\times 1}$ that satisfies constraints (17.6) and (17.7); for this point we can calculate the value of the $m+n$ deviation variables $s_i[\bar{x}]$, $i=1,2,\ldots,m+n$. Also let $s_{\lambda_k}^{(1)}[\bar{x}]$, $k=1,2,\ldots,n-1,n,n+1,\ldots,n+m$, be the same deviation variables reindexed so as to appear in their increasing numerical order,

$$(17.10) \qquad s_{\lambda_1}^{(1)}[\bar{x}] \leqslant s_{\lambda_2}^{(1)}[\bar{x}] \leqslant \ldots \leqslant s_{\lambda_{n-1}}^{(1)}[\bar{x}] \leqslant s_{\lambda_n}^{(1)}[\bar{x}] \leqslant s_{\lambda_{n+1}}^{(1)}[\bar{x}] \\ \ldots \leqslant s_{\lambda_{n+m}}^{(1)}[\bar{x}]\,.$$

We then have

$$(17.11) \qquad \left(\sum_{k=1}^{n-1} s_{\lambda_k}^{(1)}[\bar{x}] \geqslant 1\right) \Rightarrow (t_p[\bar{x}] \geqslant s_{\lambda_{n-1}}^{(1)}[\bar{x}]), \qquad p = 1,2,3,\ldots\,.$$

Proof

Let us suppose that in solving the linear program (17.1) in which we did not take into account condition (4), we had obtained a noninteger solution indicated by $[\tilde{x}]$ (which is, let us remember, an extreme point of the convex polyhedron defined by (17.6) and (17.7)). This point corresponds to the intersection of n hyperplanes of which the deviation variables $s_{i_\alpha}^{(1)}[\tilde{x}]$, $\alpha=1,2,\ldots,n$, are null for this point.

We shall add the first Dantzig–Manne constraint

$$(17.12) \qquad t_1[x] = -1 + \sum_{\alpha=1}^{n} s_{i_\alpha}^{(1)}[x], \qquad t_1[x] \geqslant 0\,.$$

In particular, for $[x]=[\tilde{x}]$, we have

$$(17.13) \qquad t_1[\tilde{x}] = -1\,,$$

[1] In this section we shall indicate by s_i and t_p the deviation variables of the inequalities in order to abbreviate the notations and proofs. In the later sections we shall reintroduce the notation u_i.

hence a negative number, which indicates that constraint (17.12) is not satisfied by $[\tilde{x}]$. The value of $t_1[x]$ for another point $[x] = [\tilde{x}]$ is

$$(17.14) \qquad t_1[\bar{x}] = -1 + \sum_{\alpha=1}^{n} s_{i_\alpha}^{(1)}[\bar{x}].$$

Let us now reindex the n deviation variables $s_{i_\alpha}^{(1)}[x]$ in such a way that their total order is their natural increasing order, namely,

$$(17.15) \qquad s_{j_1}^{(1)}[\bar{x}] \leqslant s_{j_2}^{(1)}[\bar{x}] \leqslant \dots \leqslant s_{j_{n-1}}^{(1)}[\bar{x}] \leqslant s_{j_n}^{(1)}[\bar{x}].$$

Let us observe that the ordered set (17.15) is a subset of set (17.10) the order of which is induced by that set.

Hence we can deduce that

$$(17.16) \qquad s_{j_n}^{(1)}[\bar{x}] \geqslant s_{\lambda_{n-1}}^{(1)}[\bar{x}]$$

and that:

$$(17.17)^1 \qquad \sum_{\beta=1}^{n-1} s_{j_\beta}^{(1)}[\bar{x}] \geqslant \sum_{k=1}^{n-1} s_{\lambda_k}^{(1)}[\bar{x}].$$

By using the reindexing j_β, $\beta = 1, 2, \dots, n$, we can express (17.14) in the form

$$(17.18) \qquad t_1[\bar{x}] = -1 + \sum_{\beta=1}^{n} s_{j_\beta}^{(1)}[\bar{x}],$$

or again,

$$(17.19) \qquad t_1[\bar{x}] = s_{j_n}^{(1)}[\bar{x}] - 1 + \sum_{\beta=1}^{n-1} s_{j_\beta}^{(1)}[\bar{x}].$$

Substituting (17.16) in the first term of the right member of (17.17), and (17.19) in the third term of the right member of (17.19), we obtain

$$(17.20) \qquad t_1[\bar{x}] \geqslant s_{\lambda_{n-1}}^{(1)}[\bar{x}] - 1 + \sum_{k=1}^{n-1} s_{\lambda_k}^{(1)}[\bar{x}].$$

[1] For the purpose of instruction let us take a numerical example to illustrate this. If $n = 5$ and $n+m = 8$, the following table shows eight totally ordered numbers corresponding to $s_{\lambda_k}^{(1)}[\bar{x}]$:

	1	2	3	4	5	6	7	8	
(1)	0.80	0.95	1.20	1.20	2.63	3.15	6.44	9.06	indices λ_k

Let there be a subset with five elements of this ordered set.

		1		2	3	4	5		
(2)	/	0.95	/	1.20	2.63	3.15	6.44	/	indices j_β

(17.16) indicates that the fifth element of (2), namely 6.44, is greater than or equal to the fifth element of (1). Also, (17.17) indicates that the sum of the first four elements of (2), namely 7.93, is greater than or equal to the sum of the first four elements of (1), namely 4.15.

Let us suppose that we have

$$(17.21) \qquad \sum_{k=1}^{n-1} s_{\lambda_k}^{(1)}[\bar{x}] \geqslant 1 .$$

If we substitute (17.21) we obtain

$$(17.22) \qquad t_1[\bar{x}] \geqslant s_{\lambda_{n-1}}^{(1)}[\bar{x}] ,$$

which proves (17.11) for $p = 1$.

To the set of constraints (17.6) and (17.7), let us add the constraint (17.12) of which the value of the deviation variable for $[\bar{x}]$ is $t_1[\bar{x}]$.

From (17.22) the $(n-1)$ smallest values of the $n+m+1$ deviation variables of the new polyhedron include the value of the right member of constraint (17.12); they are the same as before. The property given by (17.22) will thus apply by recurrence to $p = 2$, namely,

$$(17.23) \qquad t_2[\bar{x}] \geqslant s_{\lambda_{n-1}}^{(2)}[\bar{x}] .$$

And by using a similar procedure for 3, 4, ..., p, the theorem is thus proved.

Theorem 17.I

If $[\bar{x}]_{n\times 1}$ is a point corresponding to an optimal integer solution of program (17.1), then the Dantzig–Manne method can only converge if the $(n-1)$ smallest values of the deviation variables $s_i[\bar{x}]$, $i = 1, 2, ..., n+m$, are null.

Proof

If $[\bar{x}]_{n\times 1}$ is to be an optimal solution of the linear program obtained by ignoring constraint (4) of (17.1) and by adding the Dantzig–Manne constraints (17.12), then the point $[\bar{x}]$ must be a vertex of the convex polyhedron obtained by cutting the convex polyhedron defined by (17.6) and (17.7) by the constraints defined by (17.12). Let us take the following convex polyhedron:

$$(17.24) \qquad [a]_{m\times n} \cdot [x]_{n\times 1} \geqslant [b]_{m\times 1} ,$$

$$(17.25) \qquad [x]_{n\times 1} \geqslant [0]_{n\times 1} ,$$

$$(17.26) \qquad \left(\sum_{\alpha=1}^{n} s_{i_\alpha}^{(p)}[x]\right) - t_p[x] = 1 , \qquad p = 1, 2, \ldots, r ,$$

where r is the number of the Dantzig–Manne constraints.

A vertex of this polyhedron is defined by n null deviation variables.

By hypothesis, point $[\bar{x}]$ constitutes an optimal solution with integer values, the values of the deviation variables $s_i[\bar{x}]$, $i = 1, 2, ..., n+m$, being integer. If the $(n-1)$ smallest values of the deviation variables $s_{\lambda_k}^{(1)}[\bar{x}]$, $k = 1, 2, ..., n-1$, are not null, we have

$$(17.27) \qquad \sum_{k=1}^{n-1} s_{\lambda_k}^{(1)}[\bar{x}] \geqslant 1 .$$

The largest of the $(n-1)$ smallest values of these deviation variables, $s^{(1)}_{\lambda_{n-1}}[\bar{x}]$ is greater than or equal to 1.

Using (17.27) and the lemma expressed by (17.11), we have

$$t_p[\bar{x}] \geqslant s^{(1)}_{\lambda_{n-1}}[\bar{x}] \geqslant 1. \tag{17.28}$$

As a result the value of the deviation variables of the r Dantzig–Manne constraints cannot be null, whatever the value of r. Hence we have

$$(s^{(1)}_{\lambda_{n-1}}[\bar{x}] \geqslant 1) \Rightarrow (s^{(1)}_{\lambda_n}[\bar{x}] \geqslant 1) \Rightarrow \dots \Rightarrow (s^{(1)}_{\lambda_{n+m}}[\bar{x}] \geqslant 1). \tag{17.29}$$

And, in accordance with (17.18) and (17.29) only the value of the $n-2$ deviation variables $s^{(1)}_{\lambda_k}[\bar{x}]$, $k = 1, 2, \dots, n-2$, can eventually be null. Hence point $[\bar{x}]$ will never be the intersection of n hyperplanes among those defined by (17.24)–(17.26).

We shall give an example further on.

It is possible to provide a more geometric illustration of this theorem. If the Dantzig–Manne method is to produce convergence, the $(n-1)$ smallest values of the deviation variables evaluated for the optimal point $[\bar{x}]$ must all be null; that is to say, $[\bar{x}]$ must be at the intersection of $(n-1)$ hyperplanes of the convex polyhedron defined by (17.6) and (17.7).

A special case of some importance occurs where the necessary condition for this theorem is always satisfied, namely, when the n variables x_i of $[x]$ can only assume the values of 0 or 1. In this case only the vertices of a hypercube of $\mathbf{R}^n$ can be solutions, that is to say, when we have the assurance that the optimal point $[\bar{x}]$ is the intersection of n hyperplanes delimiting this cube and must therefore, a fortiori, be at the intersection of $(n-1)$ of them.

Theorem 17.II

Let us take a point $[x^*]$ belonging to the convex polyhedron,

$$[a]_{m\times n}\cdot[x]_{n\times 1} \geqslant [b]_{m\times 1}, \tag{17.30}$$

$$[x]_{n\times 1} \geqslant [0]_{n\times 1}, \tag{17.31}$$

but where we make it an assumed condition that the components of this point must contain at least one that is noninteger. We also assume that the point is such that

$$[c]'.[x^*] < [c]'.[\bar{x}], \tag{17.32}$$

where $[x^*]$ corresponds to a minimal integer solution of program (17.1).

In that case another necessary condition for convergence with the Dantzig–Manne method is

$$\sum_{k=1}^{n-1} s^{(1)}_{\lambda_k}[x^*] < 1. \tag{17.33}$$

Proof

By hypothesis $[x^*]$ is not integer and includes at least one noninteger component. Since the point corresponds to a value of the economic function f in (1) of (17.1) that is less than the one taken for the optimal solution corresponding to $[\bar{x}]$, and also by taking (17.32) into account, there must be at least one of the Dantzig–Manne constraints (17.26) not satisfied by $[x^*]$. Hence there is a value p in (17.26) such that $t_p[x^*] < 0$.

Let us suppose that (17.33) is not satisfied and that we have

$$(17.34) \qquad \sum_{k=1}^{n-1} s_{\lambda_k}^{(1)}[x^*] \geqslant 1\,;$$

this results in $s_{\lambda_{n-1}}^{(1)}[x^*] > 0$, remembering that the $s^{(1)}[x^*]$ are indexed in their increasing order of values.

In accordance with lemma (17.11) we have

$$(17.35) \qquad t_p[x^*] \geqslant s_{\lambda_{n-1}}^{(1)}[x^*], \qquad p = 1, 2, \ldots, r,$$

and since $s_{\lambda_{n-1}}^{(1)}[x^*] > 0$, then:

$$(17.36) \qquad t_p[x^*] > 0, \qquad p = 1, 2, \ldots, r,$$

which contradicts the hypothesis formulated immediately before (17.34), namely, that there is a value of p in (17.26) such that $t_p[x^*] < 0$.

Observation

All our explanations so far in this section have been based on a program such as (17.1) where we are seeking a minimization. All the considerations mentioned can apply equally to a search for the maximum with the sole proviso that we must invert condition (17.32) by

$$(17.37) \qquad [c]'.[x^*] > [c]'.[\bar{x}]$$

and must replace *minimal* and *smaller* by *maximal* and *greater* at the appropriate place in the statements and proofs.

3. Examples

We shall now present two examples. The first will illustrate the Dantzig–Manne method and make use of Theorem 17.I. The second will reveal the presence of nonconvergence when the conditions of Theorem 17.II are not satisfied. On account of its small number of variables the first will be shown graphically, the second by means of simplex tables.

First Example

$$(17.38) \qquad (1)\ \ [\text{MAX}]\ g = x_1 + x_2,$$

$$(2)\ \ 2x_1 \leqslant 3.$$

(3) $2x_2 \leqslant 3$,

(4) $x_1, x_2 \in \mathbf{N}$.

Let us replace condition (4) of (17.38) by

(17.39) $$x_1 \geqslant 0, \qquad x_2 \geqslant 0, \qquad x_1, x_2 \in \mathbf{R},$$

and let us then consider the linear program:

(17.40)
(1) $[\text{MAX}]\ g = x_1 + x_2$,

(2) $2x_1 + s_1 = 3$,

(3) $2x_2 + s_2 = 3$,

(4) $x_1 - s_3 = 0$,

(5) $x_2 - s_4 = 0$,

(6) $x_1, x_2, s_1, s_2, s_3, s_4 \geqslant 0, \qquad x_1, x_2, s_1, s_2, s_3, s_4 \in \mathbf{R}$.

From a brief consideration of Fig. 17.1a we can see that the point $[x^{(1)}] = [3/2\ 3/2]$ that gives the optimal solution is not an integer solution. This point is the vertex corresponding to the deviation variables $s_1 = s_2 = 0$. We now add the Dantzig–Manne constraint,

(17.41) $$s_1 + s_2 - t_1 = 1, \qquad t_1 \geqslant 0.$$

Substituting (2) and (3) of (17.40) in (17.41) we obtain

(17.42) $$2x_1 + 2x_2 + t_1 = 5, \qquad t_1 \geqslant 0.$$

This constraint has been added in Fig. 17.1b and produces a new convex polyhedron.

With such a simple polyhedron we can easily see the minimum, observing at the same time that all the points of the edge that connects points [1 3/2] and [3/2 1] are also optimal solutions, since the straight lines $2x_1 + 2x_2 = 5$ and $g = x_1 + x_2$ are parallel.

Point $[x^{(2)}]$ does not provide an integer solution and corresponds to the deviation variables $t_1 = s_2 = 0$. We now add the Dantzig–Manne constraint,

(17.43) $$t_1 + s_2 - t_2 = 1, \qquad t_2 \geqslant 0.$$

If we substitute the relations (3) of (17.40) on the one hand and (17.41) on the other, in (17.43), we obtain

(17.44) $$2x_1 + 4x_2 + t_2 = 7, \qquad t_2 \geqslant 0.$$

This constraint has been introduced in Fig. 17.1c and produces a new convex polyhedron.

By inspection, we see that the minimum for g in this new polyhedron corresponds to $[x^{(3)}] = [3/2\ 1]$; this does not provide an integer solution and the

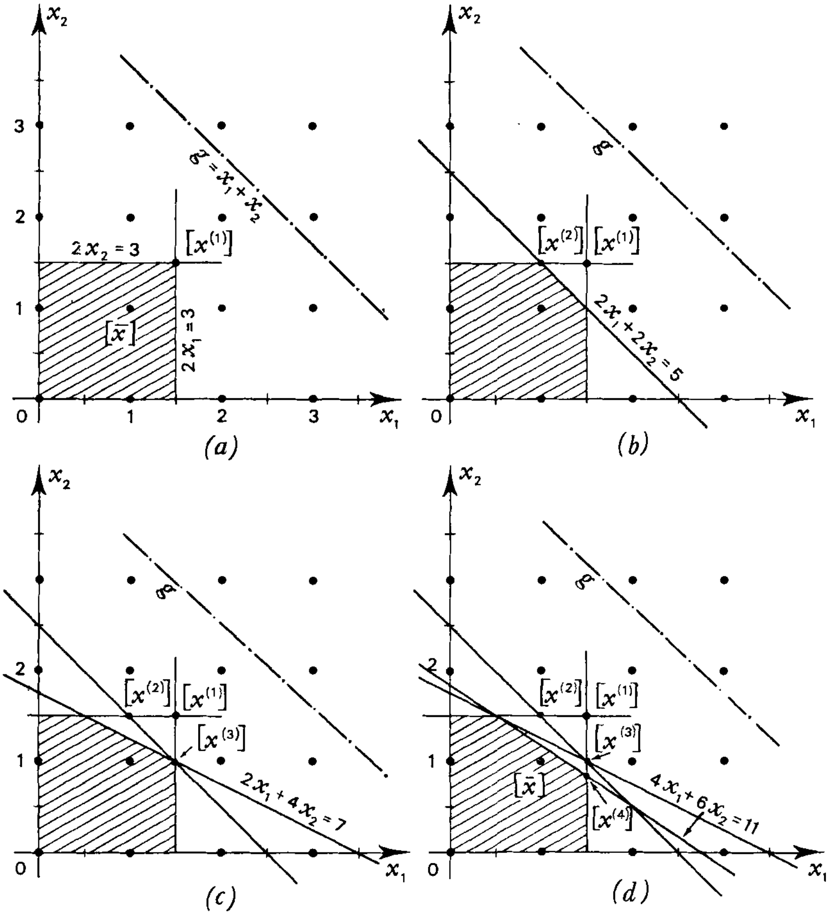


FIG. 17.1

following deviation variables are associated with it: $t_2 = s_1 = 0$. We now add a new Dantzig–Manne constraint,

$$(17.45)\qquad t_2 + s_1 - t_3 = 1, \qquad t_3 \geqslant 0.$$

By referring to the variables x_1 and x_2 we obtain

$$(17.46)\qquad 4x_1 + 6x_2 + t_3 = 11, \qquad t_3 \geqslant 0.$$

This fresh constraint has been introduced into Fig. 17.1d and gives a new convex polyhedron for which we obtain a minimum for g corresponding to $[x^{(4)}] = [3/2\ 5/6]$ that is still not an integer solution.

Indeed, we could continue indefinitely in this way without obtaining an integer solution. In fact, from simple inspection, we see that the maximum of function g of (17.38) is reached for the integer point $[x] = [1\ \ 1]$ shown in Fig. 17.1. By referring to (17.40) we have

(17.47)

$$s_1[1\ \ 1] = 1\,, \quad s_2[1\ \ 1] = 1\,, \quad s_3[1\ \ 1] = 1\,, \quad s_4[1\ \ 1] = 1\,.$$

Here $[x] \in \mathbf{R}^2$ and the $(n-1) = 2-1 = 1$ smallest values of the deviation variables are all equal to 1 (it is sufficient to consider the smallest, whichever it is). Hence none of them are null and, in accordance with Theorem 17.I, the process does not converge.

Second Example

Let us consider the integer linear program:

$$(17.48)\qquad \begin{aligned} &(1)\quad [\text{MIN}]\, f = -4x_1 - 3x_2 - 3x_3\,, \\ &(2)\quad 3x_1 + 4x_2 + 4x_3 \leqslant 6\,, \\ &(3)\quad 0 \leqslant x_1 \leqslant 1\,, \\ &(4)\quad 0 \leqslant x_2 \leqslant 1\,, \\ &(5)\quad 0 \leqslant x_3 \leqslant 1\,, \\ &(6)\quad x_1\,,\, x_2\,,\, x_3 \in \mathbf{N}\,. \end{aligned}$$

A complete enumeration of the eight vertices of the unit cube of $\mathbf{R}^3$ easily reveals that the minimum for this program is obtained for the point

$$(17.49)\qquad [\bar{x}] = [1\ \ \ 2\ \ \ 0]\,.$$

At this point the value of the economic function is $f = -4$. Since the variables x_1, x_2, and x_3 are bivalent in accordance with conditions (3)–(6) the necessary requirement for convergence with the Dantzig–Manne procedure given by Theorem 17.I is satisfied.

Nevertheless convergence cannot occur if the method is used to solve this program, since the necessary condition given by Theorem 17.II is not satisfied, as we shall now show.

Indeed, let us take the point $[x^*] = [1/2\ 1/2\ 1/2]$ of which all the coordinates are not integers. The value of the economic function f at this point is -5.

Let us rewrite constraints (2)–(5), adding to them the deviation variables s_i, $i = 1, 2, \ldots, 7$. It follows that

$$(17.50)\qquad \begin{aligned} &(1)\quad 3x_1 + 4x_2 + 4x_3 + s_1 = 6\,, \\ &(2)\quad x_1 + s_2 = 1\,, \\ &(3)\quad x_2 - s_3 = 1\,, \\ &(4)\quad x_3 - s_4 = 1\,, \end{aligned}$$

$$
\begin{aligned}
&(5)\quad x_1+s_5 = 0,\\
&(6)\quad x_2+s_6 = 0,\\
&(7)\quad x_3+s_7 = 0.
\end{aligned}
$$

For the point $[x^*]$, we have

$$
\begin{aligned}
(17.51)\qquad & s_1[x^*] = 1/2,\ s_2[x^*] = 1/2,\ s_3[x^*] = 1/2,\ s_4[x^*] = 1/2,\\
& s_5[x^*] = 1/2,\ s_6[x^*] = 1/2,\ s_7[x^*] = 1/2.
\end{aligned}
$$

The sum of the $n-1 = 3-1 = 2$ smallest deviation variables $s_i[x^*]$ that are here all equal to $1/2$ is 1, and by using the notation of (17.10) we have

$$(17.52)\qquad \sum_{k=1}^{n-1} s_{\lambda_k}^{(1)}[x^*] = 1.$$

of g will take place for

$$(17.53)\qquad [x^{(2)}] = [1 \quad 3/2],$$

Since $-5 < -4$ and by the use of line (1) of (17.48), we have

$$(17.54)\qquad [c]'.[x^*] < [c'].[\bar{x}].$$

Conditions (17.53) and (17.54) imply that the necessary requirements for convergence of Theorem 17.II are not satisfied and that the Dantzig–Manne method cannot produce it. We shall show this by carrying out a few iterations.

Let us suppose $g = -f$. Then, by adding the deviation variables $s_1, s_2, s_3, s_4 \geqslant 0$ and by considering the constraint $x_i \geqslant 0$, $i = 1, 2, 3$, instead of $x_i \in \mathbf{N}$, program (17.48) becomes (refer to what we performed in (16.88))

$$
(17.55)\qquad
\begin{aligned}
&(1)\quad [\mathrm{MAX}]\ g = 4x_1+3x_2+3x_3.\\
&(2)\quad 3x_1+4x_2+4x_3+s_1 = 6,\\
&(3)\quad x_1+s_2 = 1,\\
&(4)\quad x_2+s_3 = 1,\\
&(5)\quad x_3+s_4 = 1,\\
&(6)\quad x_1, x_2, x_3, s_1, s_2, s_3, s_4 \geqslant 0.
\end{aligned}
$$

A basic solution for program (17.55) is $x_1 = x_2 = x_3 = 0$ and $s_1 = 6$, $s_2 = 1$, $s_3 = 1$, $s_4 = 1$. We set out the calculations in successive tables such as that of (16.95) to which the reader may refer. In these tables we do not include the columns corresponding to the variables of the vector $[\varphi]_{m\times 1}$ of which the elements are null.

(17.56)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
			g	s_1	s_2	s_3	s_4	x_1	x_2	x_3
(0)	g	0	1	0	0	0	0	-4	-3	-3
(1)	s_1	6	0	1	0	0	0	③	4	4
(2)	s_2	1	0	0	1	0	0	1	0	0
(3)	s_3	1	0	0	0	1	0	0	1	0
(4)	s_4	1	0	0	0	0	1	0	0	1

↑

Column giving the point $s_1 = 6$, $s_2 = 1$, $s_3 = 1$, $s_4 = 1$, $g = 0$.

Table (17.56) does not contain an optimal solution since line (0) is not non-negative (see Theorem 16.1). We shall now use the primal-simplex method explained in Volume 1 and this will provide an optimal solution in two iterations; the reader is left to perform the calculations. The pivoting elements have been circled. The element in column (6) of line (0) is the most negative of this line, so that we have

$$\min(6/3,\ 1/1,\ 1/1,\ 1/1) = 1$$

and we take, for example, line (2) as the pivot line, although we could equally well have chosen line (3) or (4). By effecting a simplex operation we obtain table (17.57).

(17.57)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
			g	s_1	s_2	s_3	s_4	x_1	x_2	x_3
(0)	g	4	1	0	4	0	0	0	-3	-3
(1)	s_1	3	0	1	-3	0	0	0	④	4
(2)	x_1	1	0	0	1	0	0	1	0	0
(3)	s_3	1	0	0	0	1	0	0	1	0
(4)	s_4	1	0	0	0	0	1	0	0	1

↑

Column giving the point $s_1 = 3$, $x_1 = 1$, $s_3 = 1$, $s_4 = 1$, $g = 4$.

The solution provided by table (17.57) is not optimal, and a new table (17.58) is obtained by taking the circled element as the pivot. In line (5) we have circled the Dantzig–Manne constraint subsequently added.

(17.58)

		(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
			g	s_1	s_2	s_3	s_4	x_1	x_2	x_3	t_1
(0)	g	25/4	1	3/4	7/4	0	0	0	0	0	0
(1)	x_2	3/4	0	1/4	-3/4	0	0	0	1	1	0
(2)	x_1	1	0	0	1	0	0	1	0	0	0
(3)	s_3	1/4	0	-1/4	3/4	1	0	0	0	-1	0
(4)	s_4	1	0	0	0	0	1	0	0	1	0
(5)	t_1	-1	0	-1	-1	0	0	0	0	$\textcircled{-1}$	1

Dantzig-Manne constraint added ← since the solution is not integer.

Column for the deviation variable of the added Dantzig-Manne constraint.

Let us ignore line (5) and column (9) added here for convenience although they only intervene subsequently. Then table (17.58) represents an optimal solution of (17.55), since the elements of line (0) are nonnegative, as well as the elements in the next four lines of column (0). This soultion is

(17.59) $$x_2 = 3/4; x_1 = 1, s_3 = 1/4, s_4 = 1, x_3 = 0, s_1 = 0, s_2 = 0.$$

However this is not an integer solution. By reference to the constraints (17.50) we see that this point corresponds to $s_1 = s_2 = s_4 = 0$. We now add the following Dantzig–Manne constraint to the program:

(17.60) $$s_1+s_2+s_4-t_1 = 1, \qquad t_1 \geqslant 0.$$

That is, in accordance with (17.50),

(17.61) $$-s_1-s_2-x_3+t_1 = -1, \qquad t_1 \geqslant 0.$$

When this constraint is added to table (17.58) it still does not provide a solution since $t_1 < 0$, but line (0) is nonnegative. We shall now use the dual-simplex algorithm explained in Section 16.

For pivot we take the element -1 circled in table (17.58), the choice of which the reader can easily verify. We obtain table (17.62).

(17.62)

		(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
			g	s_1	s_2	s_3	s_4	x_1	x_2	x_3	t_1
(0)	g	25/4	1	3/4	7/4	0	0	0	0	0	0
(1)	x_2	-1/4	0	-3/4	-7/4	0	0	0	1	0	1
(2)	x_1	1	0	0	1	0	0	1	0	0	0
(3)	s_3	5/4	0	3/4	7/4	1	0	0	0	0	-1
(4)	s_4	0	0	-1	-1	0	1	0	0	0	1
(5)	x_3	1	0	1	1	0	0	0	0	1	-1

↑

Column corresponding to the point $x_2 = -1/4$, $x_1 = 1$,
$s_3 = 5/4$, $s_4{}' = 0$, $x_3 = 1$.

The point obtained is not a solution, since $x_2 < 0$, and we perform a new iteration with the dual-simplex algorithm, taking $-3/4$ as pivot, although $-7/4$ could equally well have been chosen. We give table (17.63) below.

(17.63)

		(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
			g	s_1	s_2	s_3	s_4	x_1	x_2	x_3	t_1
(0)	g	6	1	0	0	0	0	0	1	0	1
(1)	s_1	1/3	0	1	7/3	0	0	0	-4/3	0	-4/3
(2)	x_1	1	0	0	1	0	0	1	0	0	0
(3)	s_3	1	0	0	0	1	0	0	1	0	0
(4)	s_4	1/3	0	0	4/3	0	1	0	-4/3	0	-1/3
(5)	x_3	2/3	0	0	-4/3	0	0	0	4/3	1	1/3

↑

Column corresponding to the point $s_1 = 1/3$, $x_1 = 1$, $s_3 = 1$,
$s_4 = 1/3$, $x_3 = 2/3$, $g = 6$.

Table (17.63) provides a solution, but it is not integer. We must now add the new Dantzig–Manne constraint obtained in the same way as in (17.61),

$$-s_2-x_2-t_1+t_2 = -1, \qquad t_2 \geqslant 0. \tag{17.64}$$

This constraint is added to table (17.63) and we proceed by the use of the algorithm, the calculations being left to the reader. However we have sufficiently proved that the algorithm cannot converge for this example and that an integer solution of program (17.48) will never be obtained by this method.

Section 18. **Solving Linear Equations with Integers**[1]

1. Introduction and Definitions

One of the first problems to confront mathematicians was to solve equations that *appear opportunely*. It was not until 1621 that Bachet de Mezinac, the translator and commentator of Diophante[2] was able to solve the equation

$$ax+by = c, \tag{18.1}$$

when x and y must be nonnegative integers. Thus, the equation

$$3x+2y = 7, \tag{18.2}$$

has as its sole solution $x = 1$, $y = 2$. However, more complicated cases may occur and it is necessary to have appropriate methods available for dealing with them.

Before explaining a general method of solution, we may mention that the term *Diophantian equations* is often applied to equations or systems of equations the solution of which can only consist of natural numbers (nonnegative integers).

Let us now consider the case of linear equations of the type

$$\begin{aligned} A_{11}x_1 + A_{12}x_2 + \dots + A_{1n}x_n &= b_1, \\ A_{21}x_1 + A_{22}x_2 + \dots + A_{2n}x_n &= b_2, \\ \dots\dots\dots\dots \quad \dots\dots\dots\dots\dots\dots & \\ A_{m1}x_1 + A_{m2}x_2 + \dots + A_{mn}x_n &= b_m, \\ A_{i,j},\ b_i,\ x_j \in \mathbf{Z}, \qquad i = 1, 2, \dots, m;\ & j = 1, 2, \dots, n. \end{aligned} \tag{18.3}$$

Or in matrical form,

$$[A]_{m\times n}\cdot[x]_{n\times 1} = [b]_{m\times 1}. \tag{18.4}$$

[1] Part of this section is adapted from the article by J. C. Fiorot and M. Gondron [K32].

[2] He lived during the 4th century A.D.

We intend to show that when a solution $[x^{(0)}]_{n\times 1}$ of (18.4) exists all the solutions $[x]_{n\times 1}$ are expressed in the form

$$[x]_{n\times 1} = [x_0]_{n\times 1} + [w]_{n\times(n-m)}\cdot[t]_{(n-m)\times 1}, \tag{18.5}$$

where

$[x_0]_{n\times 1}$ is an integer solution of (18.4),

$[w]_{n\times(n-m)}$ is a matrix with related integer elements $(w_{ij}\in \mathbf{Z},\ i=1,2,\ldots,n,\ j=1,2,\ldots,(n-m))$,

$[t]_{(n-m)\times 1}$ is a vector with related integer elements $(t_i\in \mathbf{Z},\ i=1,2,\ldots,(n-m))$.

The results obtained in the present section will permit an easy transition when we study Gomory's methods of integer programming in Section 19.

Substitution Matrices[1]

Let us consider a matrix $[\Pi]_{n\times n}$ of which the elements Π_{ij}, $i,j=1,2,\ldots,n$, are such that

$$\Pi_{ij} = 0 \text{ or } 1, \qquad i,j = 1,2,\ldots,n, \tag{18.6}$$

$$\sum_{i=1}^{n} \Pi_{ij} = 1, \qquad j = 1,2,\ldots,n, \tag{18.7}$$

$$\sum_{j=1}^{n} \Pi_{ij} = 1, \qquad i = 1,2,\ldots,n, \tag{18.8}$$

then $[\Pi]$ is called a *substitution matrix.*

Differently stated, this means that we are concerned with a Boolean square in which each line and each column contains one and only one 1.

Thus the matrix $[\Pi]$ given in (8.9) is a substitution matrix of order 8.

$$[\Pi]_{8\times 8} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \tag{18.9}$$

Let any $[A]_{m\times n}$ be the premultiplication of $[A]$ by a substitution matrix

[1] Also called *commutative, permutative,* or *distributive* matrices.

$[\Pi]_{m\times n}$, that is to say $[\Pi]_{m\times m}\cdot[A]_{m\times n}$ will produce a permutation of the lines of $[A]$, whereas the postmultiplication of $[A]$ by a substitution matrix $[\Pi]_{n\times m}$ will produce a permutation of the columns of $[A]$. An example is given in (18.10) and (18.11).

(18.10)

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{array}{c} \left[\begin{matrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \end{matrix}\right] \end{array} \begin{array}{c} (1) \\ (2) \\ (3) \\ (4) \end{array} = \begin{bmatrix} a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \end{bmatrix} \begin{array}{c} (4) \\ (2) \\ (1) \\ (3) \end{array}$$

(18.11)

$$\begin{array}{ccccc} (1) & (2) & (3) & (4) & (5) \\ \hline a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{34} & a_{44} & a_{45} \end{array} \cdot \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} = \begin{array}{ccccc} (3) & (5) & (1) & (2) & (4) \\ \hline a_{13} & a_{15} & a_{11} & a_{12} & a_{14} \\ a_{23} & a_{25} & a_{21} & a_{22} & a_{24} \\ a_{33} & a_{35} & a_{31} & a_{32} & a_{34} \\ a_{43} & a_{45} & a_{41} & a_{42} & a_{44} \end{array}$$

Transposition Matrix

A substitution matrix in which two and only two 1's do not belong to the main diagonal is a transposition matrix.

Such a matrix premultiplying $[A]$ permutates between them two lines of $[A]$ and postmultiplying $[B]$ permutates between them two columns of $[B]$.

Thus the matrix $[P]$ given in (18.12) is a transposition matrix.

$$(18.12) \qquad [P]_{6\times 6} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{array}{l} \ (1) \\ \leftarrow (2) \\ \ (3) \\ \ (4) \\ \ (5) \\ \leftarrow (6) \end{array}$$

This matrix would permutate a line (2) with a line (6) or a column (2) with a column (6), depending on whether it is a question of pre- or postmultiplication.

To define the transposition that has been carried out it is convenient to express a transposition matrix as $[P_{ij}]$ if it permutates i with j (lines or columns as the case may be). Thus (18.12) can be expressed as $[P_{26}]_{6\times 6}$.

Unimodular Matrices. Definitions

The term *unimodular matrix* is given to a matrix $n \times n$ of which the determinant[1] has the value $(+1)$, (-1), or (0); that of *regular unimodular matrix* is given to a matrix $n \times n$ of rank n, that is to say, one where the determinant is $(+1)$ or (-1).

The product of two unimodular matrices gives a unimodular matrix.

A transposition matrix is a regular unimodular matrix and its determinant is (-1) since its parity[2] is always uneven.

A substitution matrix is always a regular unimodular matrix. If its parity is even its determinant is $(+1)$, and if it is odd its determinant has a value of (-1).

By an abuse of terms, but for convenience, we shall sometimes use the term *unimodular* for a matrix $m \times n$ $(m \neq n)$ if it is of rank $s = \text{MIN}\ (m, n)$ and if its determinants of order s are equal to $(+1)$, (-1), or (0). Clearly, since the matrix is of rank s, there must be at least one determinant of order s that is nonnull.

Subtraction Matrix

Let us first consider a unit matrix $n \times n$ in which a 0 has been replaced by any number $(-\alpha)$ where α is any real number. If $(-\alpha)$ is placed in the ith line and jth column, this matrix will have the notation $[U_{i,j,\alpha}]$ to recall these facts. An example of this is shown in (18.13).

$$(18.13)\qquad [U_{3,5,\alpha}]_{6\times 6} = \begin{array}{cl} \begin{array}{cccccc} & & & & \downarrow j & \end{array} & \\ \left[\begin{array}{cccccc} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -\alpha & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] & \begin{array}{l} \\ \\ \leftarrow i. \\ \\ \\ \\ \end{array} \end{array}$$

Matrices of this type are called *elementary subtraction matrices*. It is evident that their determinant is equal to 1 and that they are therefore regular uni-

[1] Some writers insist that the minors must also have these values. As far as we are concerned, these supplementary conditions define *totally unimodular matrices* which will not be used here.

[2] The substitutions can be divided into two classes, those having the same parity as the unit substitution (unit matrix) and obtained by an even number of tranpositions, and, secondly, those having a complementary parity obtained by an odd number of transpositions. See, for example, [K18], p. 118.

modular matrices. Such matrices possess the following properties. The premultiplication of a matrix $[A]$ by $[U_{i,j,\alpha}]$ has the effect of reducing the ith line of $[A]$ by α times the jth line of $[A]$. The postmultiplication of a matrix $[B]$ by $[U_{i,j,\alpha}]$ reduces the jth column of $[B]$ by α times the ith column of $[A]$.

The following example illustrates this property:

(18.14)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21}-\alpha a_{41} & a_{22}-\alpha a_{42} & a_{23}-\alpha a_{43} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix},$$

(18.15)

$$\begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14}-\alpha b_{12} \\ b_{21} & b_{22} & b_{23} & b_{24}-\alpha b_{22} \\ b_{31} & b_{32} & b_{33} & b_{34}-\alpha b_{32} \end{bmatrix}.$$

If we now take two elementary subtraction matrices $[U_{i,j,\alpha}]$ and $[U_{i,k,\beta}]$, $j \neq k$, their product will give a matrix formed by 1 in the main diagonal and elsewhere by 0, except in (i,j) where we find $(-\alpha)$ and in (i,k) where we find $(-\beta)$. Hence, by successive multiplications performed in any order we can construct matrices that we call *composite subtraction matrices*. The following example shows how such matrices are constructed or decomposed.

(18.16)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ -\alpha_1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -\alpha_2 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -\alpha_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\alpha_1 & 1 & -\alpha_2 & -\alpha_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

It is clear that every composite subtraction matrix has a determinant equal to 1 and is therefore a regular unimodular matrix. It superimposes the properties of subtraction, both in pre- and postmultiplication, of the elementary subtraction matrices that compose it by their products.

Theorem 18.1

The product of two subtraction matrices $n \times n$ gives a regular unimodular matrix.

This is obvious, since they are regular unimodular matrices.

The same conclusion applies to the product of a transposition matrix and a subtraction matrix and in general to every product, whatever the order of these matrices among themselves or with the others, since all of them are regular unimodular matrices.

Arithmetically Equivalent Matrices

Two matrices $[A]_{m\times n}$ and $[B]_{m\times n}$ are *arithmetically equivalent* if there are two regular unimodular matrices $[U]_{m\times m}$ and $[V]_{n\times n}$ such that

$$(18.17)\qquad [U]_{m\times m}\cdot[A]_{m\times n}\cdot[V]_{n\times n} = [B]_{m\times n}\,.$$

Let us consider an example:

(18.18)

$$\underset{[U]}{\begin{bmatrix} 1 & -2 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}} \underset{[A]}{\begin{bmatrix} 4 & 2 & 5 & -3 \\ 0 & 9 & -2 & 1 \\ 4 & 6 & 0 & -8 \end{bmatrix}} \underset{[V]}{\begin{bmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}} = \underset{[B]}{\begin{bmatrix} 14 & 2 & 9 & -23 \\ -9 & 9 & -2 & 28 \\ -2 & 6 & 0 & 10 \end{bmatrix}}.$$

The two matrices $[A]_{3\times 4}$ and $[B]_{3\times 4}$ are arithmetically equivalent.

2. Reducing a Linear System to Smith's Normal Form [K32]

Let $[A]$ be a matrix $m\times n$ of rank $r \leqslant \min\,(m, n)$, the elements of which are real numbers, and let $[D]$ be another matrix $m\times n$ of rank r of the following form known as *Smith's normal form*:

$$(18.19)\qquad [D] = \left.\underbrace{\left[\begin{array}{ccc:c} d_1 & 0 & \dots\ 0 & 0\ \dots\ 0 \\ 0 & d_2 & \dots\ 0 & 0\ \dots\ 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & d_r & 0\ \dots\ 0 \\ \hdashline 0 & 0 & \dots\ 0 & 0\ \dots\ 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots\ 0 & 0\ \dots\ 0 \end{array}\right]}_{n}\right\}m,$$

where $\qquad d_k \neq 0,\ k = 1, 2, \dots, r.$

Expressed differently, $[D]$ contains a square submatrix, the principal diagonal of which is formed of nonnull numbers, the other elements being 0's, and the other submatrices, formed from the square submatrix, that are also formed of 0.

Theorem 18.II

With every matrix $[A]_{m\times n}$ there can be associated a matrix $[D]_{m\times n}$ such as (18.19) and obtained by the transformation

$$[U]_{m\times m}\cdot[A]_{m\times n}\cdot[V]_{n\times n} = [D]_{m\times n}, \tag{18.20}$$

where $[U]$ and $[V]$ are regular unimodular matrices.

Proof

It must first be observed that the transformation expressed by (18.20) is not that known as *diagonalization* in classic matrical calculation[1] which enables us to obtain the values belonging to a square matrix. In the case considered here, $[A]$ is not necessarily a square matrix and $[V]$ usually differs from $[U]^{-1}$.

Let us begin by determining two matrices $[U_1]$ and $[V_1]$ such that

$$[U_1]\cdot[A]\cdot[V_1] = \left[\begin{array}{c|c} d_1 & [0]_{1\times(n-1)} \\ \hline [0]_{(m-1)\times 1} & [A']_{(m-1)\times(n-1)} \end{array}\right] = [D']. \tag{18.21}$$

To do so we shall proceed as follows:

a. By a suitable permutation of lines and columns, we bring the smallest term of $[A]$ in absolute but not null value into the position $(i, j) = (1, 1)$, namely, into the first line and the first column. Let $\lambda^{(1)}$ represent this term. To do this we shall employ a transposition matrix $[P_{i1}^{(1)}]$ that, by premultiplying $[A]$, brings $\lambda^{(1)}$ into the first line, and a transposition matrix $[P_{1j}^{(1)}]$ that, by postmultiplying $[A]$, brings $\lambda^{(1)}$ into the first column.

b. Then, by using suitable subtraction matrices $[U]$ and $[V]$ we shall replace each term of the first line and of the first column by the remaining p_{ij} defined as follows:

$$a_{1j} = \alpha_{1j}.\lambda^{(1)} + p_{1j}, \tag{18.22}$$

$$a_{i1} = \alpha_{i1}.\lambda^{(1)} + p_{i1}, \tag{18.23}$$

with α_{1j} and α_{i1} integer numbers.

This amounts to subtracting α_{1j} times the first column from each of the columns in which the first element is not null. This is obtained by postmultiplications with subtraction matrices $[V_{i,1}, \alpha_{1j}]$. It is also equivalent to subtracting α_{1i} times the first line from each line in which the element is not null. We obtain this by premultiplications with subtraction matrices $[U_{i,1}, \alpha_{1i}]$.

[1] The reader who may have forgotten the procedure should refer to M. Denis-Papine and A. Kaufmann, "Cours de Calcul Matriciel Appliqué," Albin Michel, Paris, 1969.

If the remaining P_{1_j} and P_{i_1} are all null we have obtained $[D']$. If not, we repeat with the new matrix what was done with $[A]$. We shall finally and definitely obtain the form $[D']$ in a finite number of iterations since, with each iteration, we replace all the nonnull terms except $\lambda^{(1)}$ in the first line and column by elements of strictly decreasing absolute value.

c.

(18.24) If $[A']_{(m-1)\times(m-1)} = [0]_{(m-1)\times(m-1)}$,

then $[U] = [U_1]$ and $[V] = [V_1]$,

satisfy (18.20). If not, we shall perform on $[A']$ the operations defined in (a) and (b) to obtain the form:

(18.25)

$$\begin{aligned} &[U_2].[D'].[V_2] = \\ &[U_2].[U_1].[A].[V_1].[V_2] = \left[\begin{array}{c|c|c} d_1 & 0 & [0]_{1\times(n-1)} \\ \hline 0 & d_2 & [0]_{1\times(n-1)} \\ \hline [0]_{(m-2)\times 1} & [0]_{(m-2)\times 1} & [A'']_{(m-2)\times(n-2)} \end{array}\right] \\ &= [D'']. \end{aligned}$$

In this way we obtain a succession of matrices $[A']$, $[A'']$, $[A''']$, ..., the dimensions of which decrease by a line and a column at each stage where (a) and (b) are performed. Accordingly, we have $r' \leqslant \min(m, n)$ such that $[A^{(r')}] = 0$ or void.

As for $[D] = [U].[A].[V]$, this is a matrix of rank r' by construction. In addition, $[D]$, obtained by multiplying $[A]$ by regular unimodular matrices, is of the same rank r as $[A]$, and $r' = r$.

3. Examples of Reduction

First Example

Let

$$(18.26) \qquad [A]_{3\times 5} = \begin{bmatrix} 1 & & 1 & -4 & 1 \\ 1 & 1 & 1 & 3 & -2 \\ 2 & 0 & 2 & -1 & -1 \end{bmatrix}.$$

One of the nonnull elements and the smallest in absolute value is 1 in position (1, 1). Hence we need not permutate the lines and columns to bring it into this position.

To carry out stage (b) we construct subtraction matrices $[U_1]$ and $[V_1]$. Let us first see how we construct the former. In the new matrix 1 must be in the position (1, 1) and 0 must be in $(i, 1)$, $i = 2, 3$. To do this we must multiply the first line of $[A]$ by (-1) and add it to the second, then multiply the first line of $[A]$ by (-2) and add it to the third, by which means we obtain the matrix $[U_1]$ shown in (18.27).

(18.27)

$$\underset{[U_1]}{\begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}} \underset{[A]}{\begin{bmatrix} 1 & 1 & 1 & -4 & 1 \\ 1 & 1 & 1 & 3 & -2 \\ 2 & 0 & 2 & -1 & -1 \end{bmatrix}} = \underset{[U_1] \cdot [A]}{\begin{bmatrix} 1 & 1 & 1 & -4 & 1 \\ 0 & 0 & 0 & 7 & -3 \\ 0 & -2 & 0 & 7 & -3 \end{bmatrix}}.$$

Now, in $[U_1] \cdot [A]$, the first column is such as we require, so we consider how $[V_1]$ is to be obtained. To do this we must multiply the first column of $[U_1] \cdot [A]$ by -1 and add it to the second column, multiply the first column by (-1) and add it to the third column, multiply the first column by 4 and add it to the fourth, multiply the first column by (-1) and add it to the fifth. This procedure means determining the elements of the subtraction matrix $[V_1]$. We notice that since line 1 of $[A]$ and of $[U_1] \cdot [A]$ does not vary in the premultiplication by $[U_1]$, the elements of $[V_1]$ can be calculated on $[A]$. It follows that

(18.28)

$$\underset{[U_1] \cdot [A]}{\begin{bmatrix} 1 & 1 & 1 & -4 & 1 \\ 0 & 0 & 0 & 7 & -3 \\ 0 & -2 & 0 & 7 & -3 \end{bmatrix}} \cdot \underset{[V_1]}{\begin{bmatrix} 1 & -1 & -1 & 4 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}} = \underset{[U_1] \cdot [A] \cdot [V_1]}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & -3 \\ 0 & -2 & 0 & 7 & -3 \end{bmatrix}}$$

We have thus obtained a matrix $[D'] = [U_1] \cdot [A] \cdot [V_1]$ that conforms to (18.21).

$$(18.29) \qquad [D'] = \begin{array}{c} \begin{array}{cc} [d_1] \quad & \quad [0] \end{array} \\ \left[\begin{array}{c|cccc} 1 & 0 & 0 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 7 & -3 \\ 0 & -2 & 0 & 7 & -3 \end{array}\right] \\ \begin{array}{cc} [0] \quad & \quad [A'] \end{array} \end{array}.$$

Let us now consider the smallest nonnull element in $[A']$; this is (-2) in position (3, 2). We must bring it to the position (2, 2) by a premultiplication of $[D']$ by a transposition matrix that will exchange the positions of lines 3 and 2.

(18.30)

$$\underset{[P_{23}]}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}} \underset{[D']}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7 & -3 \\ 0 & -2 & 0 & 7 & -3 \end{bmatrix}} = \underset{[P_{23}]\cdot[D']}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 7 & -3 \\ 0 & 0 & 0 & 7 & -3 \end{bmatrix}}.$$

We must now transform the right member of (18.30) so as to have a nonnull number in position (2, 2) and 0's in positions $(2, j)$, $j = 1, 3, 4, 5$, and $(i, 2)$, $i = 1, 3$. Since there is already a 0 in (3, 2) we need not premultiply but must postmultiply by a subtraction matrix $[V_2]$ to obtain 0's in $(2, j)$, $j = 3, 4, 5$. To obtain this matrix let us consider the second member of (18.30). The element (2, 3) is already 0; that of (2, 4) is 7, and the quotient of 7 by -2 produces -3. The element (2, 5) is -3 and the quotient of -3 by -2 produces 1. Hence we can now form matrix $[V_2]$ and, postmultiplying by it, we obtain

(18.31)

$$\underset{[P_{23}]\cdot[D']}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 7 & -3 \\ 0 & 0 & 0 & 7 & -3 \end{bmatrix}} \cdot \underset{[V_2]}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}} = \underset{[P_{23}]\cdot[D']\cdot[V_2]}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 1 & -1 \\ 0 & 0 & 0 & 7 & -3 \end{bmatrix}}.$$

The matrix of the right member of (18.31) is not yet such that all the elements $(2, j)$, $j = 3, 4, 5$, are null. Let us therefore return to stage (a) of the algorithm to seek the nonnull element with the least absolute value of lines 2 and 3; we have the choice of 1 in (2, 4) and -1 in (2, 5). If we arbitrarily select the 1 we must permutate column 2 with column 4 by a transposition matrix $[P_{24}]$ that gives us

(18.32)

$$\underset{[P_{23}]\cdot[D']\cdot[V_2]}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & -2 & 0 & 1 & -1 \\ 0 & 0 & 0 & 7 & -3 \end{bmatrix}} \underset{[P_{24}]}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}} = \underset{[P_{23}]\cdot[D']\cdot[V_2]\cdot[P_{24}]}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -2 & -1 \\ 0 & 7 & 0 & 0 & -3 \end{bmatrix}}.$$

Now let us replace the 7 in (3, 2) by a 0 by premultiplying the right member of (18.32) by a subtraction matrix $[U_2]$:

(18.33)

$$\underset{[U_2]}{\begin{bmatrix}1&0&0\\0&1&0\\0&-7&1\end{bmatrix}}\cdot\underset{[P_{23}]\cdot[D']\cdot[V_2]\cdot[P_{24}]}{\begin{bmatrix}1&0&0&0&0\\0&1&0&-2&-1\\0&7&0&0&-3\end{bmatrix}}=\underset{[U_2]\cdot[P_{23}]\cdot[D']\ [V_2]\cdot[P_{24}].}{\begin{bmatrix}1&0&0&0&0\\0&1&0&-2&-1\\0&0&0&14&4\end{bmatrix}}.$$

Next let us replace the -2 in (2, 4) and the -1 in (2, 5) by a postmultiplication by a subtraction matrix $[V_3]$:

(18.34)

$$\underset{[U_2]\cdot[P_{23}]\cdot[D']\cdot[V_2]\cdot[P_{24}]}{\begin{bmatrix}1&0&0&0&0\\0&1&0&-2&-1\\0&0&0&14&4\end{bmatrix}}\underset{[V_3]}{\begin{bmatrix}1&0&0&0&0\\0&1&0&2&1\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}}=\underset{\substack{[D'']=[U_2]\cdot[P_{23}]\cdot[D']\\ \times[V_2]\cdot[P_{24}]\cdot[V_3]}}{\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&0&14&4\end{bmatrix}}.$$

The matrix $[D'']$ thus obtained does not yet have the form of (18.19). The element with the least absolute value that is nonnull is in position (3, 5); it will be transferred to (3, 3) by a transposition matrix $[P_{53}]$:

(18.35)

$$\underset{[D'']}{\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&0&14&4\end{bmatrix}}\cdot\underset{[P_{53}]}{\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&0&0&1\\0&0&0&1&0\\0&0&1&0&0\end{bmatrix}}=\underset{[D''']=[D'']\cdot[P_{53}]}{\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&4&14&0\end{bmatrix}}.$$

$[D''']$ still does not have the form of (18.19) and we introduce a new subtraction matrix $[V_4]$:

(18.36)

$$\underset{[D''']}{\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&4&14&0\end{bmatrix}}\cdot\underset{[V_4]}{\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&-3&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}}=\underset{[D^{\mathrm{IV}}]=[D''']\cdot[V_4]}{\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&4&2&0\end{bmatrix}}$$

Since the element (3, 4) is equal to 2 and is less in absolute value than the element (3, 3) that is equal to 4, we permutate them by a transposition matrix $[P_{43}]$:

(18.37)

$$\underset{[D^{\mathrm{IV}}]}{\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&4&2&0\end{bmatrix}}\cdot\underset{[P_{43}]}{\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&0&1&0\\0&0&1&0&0\\0&0&0&0&1\end{bmatrix}}=\underset{[D^{\mathrm{V}}]=[D^{\mathrm{IV}}]\cdot[P_{43}]}{\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&2&4&0\end{bmatrix}}.$$

We are now almost at the end of the road. One last subtraction matrix, and we obtain

(18.38)

$$\underset{[D^{\mathrm{V}}]}{\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&2&4&0\end{bmatrix}}\cdot\underset{[V_5]}{\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&-2&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}}=\underset{[D^{\mathrm{VI}}]=[D^{\mathrm{V}}]\,[V_5]}{\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&2&0&0\end{bmatrix}}.$$

We have finally obtained the form (18.19) ,that is $[D] = [D^{\mathrm{VI}}]$.

Let us see through which regular and global unimodular matrices we have passed from $[A]$ to $[D]$. Regrouping all the calculations, we have

(18.39)

$$[D] = \underbrace{[U_2]\cdot[P_{23}]\cdot[U_1]}_{[U]}\cdot[A]\cdot\underbrace{[V_1]\cdot[V_2]\cdot[P_{24}]\cdot[V_3]\cdot[P_{23}]\cdot[V_4]\cdot[P_{43}]}_{[V]}\ .$$

We have

(18.40)

$[U_2]\cdot[P_{23}]\cdot[U_1]$

$$=\underset{[U_2]}{\begin{bmatrix}1&0&0\\0&1&0\\0&-7&1\end{bmatrix}}\cdot\underset{[P_{23}]}{\begin{bmatrix}1&0&0\\0&0&1\\0&1&0\end{bmatrix}}\cdot\underset{[U_1]}{\begin{bmatrix}1&0&0\\-1&1&0\\-2&0&1\end{bmatrix}}=\begin{bmatrix}1&0&0\\-2&0&1\\13&1&-7\end{bmatrix}.$$

(18.41)

$[V_1]\cdot[V_2]\cdot[P_{24}]\cdot[V_3]\cdot[P_{53}]\cdot[V_4]\cdot[P_{43}]\cdot[V_5]$

$$= \underset{[V_1]}{\begin{bmatrix} 1 & -1 & -1 & 4 & -1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}} \cdot \underset{[V_2]}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}} \cdot \underset{[P_{24}]}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}$$

$$\times \underset{[V_3]}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}} \cdot \underset{[P_{53}]}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}} \cdot \underset{[V_4]}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -3 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}$$

$$\times \underset{[P_{43}]}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}} \cdot \underset{[V_5]}{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}$$

$$= \begin{bmatrix} 1 & 1 & -2 & 5 & -1 \\ 0 & 3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 3 & 0 \\ 0 & 0 & -3 & 7 & 0 \end{bmatrix}.$$

And finally we verify that

(18.42)

$$\underset{[D]}{\left[\begin{array}{ccc|cc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \end{array}\right]} = \underset{[U]}{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 0 & 1 \\ 13 & 1 & -7 \end{bmatrix}} . \underset{[A]}{\begin{bmatrix} 1 & 1 & 1 & -4 & 1 \\ 1 & 1 & 1 & 3 & -2 \\ 2 & 0 & 2 & -1 & -1 \end{bmatrix}}$$

$$\times \underset{[V]}{\begin{bmatrix} 1 & 1 & -2 & 5 & -1 \\ 0 & 3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 3 & 0 \\ 0 & 0 & -3 & 7 & 0 \end{bmatrix}} .$$

$[U]$ and $[V]$ are regular unimodular matrices since they were obtained by the products of such matrices (see Theorem 18.I).

In our example, where $[A]$ is a matrix 3×5, its rank is $r = 3$, which means that $[D]$ has no submatrices $(m-r) \times m$ and $(m-r) \times n$.

First Observation

The matrical form (18.19) obtained by the above linear transformations is not unique and depends on the matrices $[U]$ and $[V]$ employed, and on the choice of the smallest term in absolute value in $[A]$ that becomes $[A']$. If this term is not the sole one, an arbitrary choice has to be made and, with a different selection, other matrices $[U]$, $[V]$, and $[D]$ would eventually be obtained.

Second Observation

Throughout our calculations we have assumed the elements of $[A]$ to be integers. The procedure can easily be applied to cases where these elements are fractional: all that is needed is to multiply them by the least common multiple p of the denominators to obtain a matrix

(18.43) $$[A'] = p[A],$$

in which all the elements are integers. By premultiplying by the same matrix $[U]$ and postmultiplying by the same matrix $[V]$, we then obtain a matrix $[D]$ such that

(18.44) $$[D] = p[U].[A].[V],$$

or again

$$[D'] = \frac{1}{p}[D] = [U].[A].[V].$$

But the elements of the main diagonal $[D']$ are composed of fractions, and we later assume that the elements of $[A]$ belong to $\mathbf{Z}$.

Let us now enunciate a theorem that will be of future assistance to us.

Theorem 18.III

A regular unimodular matrix formed of integers has as its inverse a similar matrix.

The proof is obvious if we recall how the inverse of a matrix is formed: we first take the transpose that is accordingly formed of integers, then the conjugate that has elements obtained in the first instance from the determinants of the transpose, so that the conjugate must be formed of integers. And since the given matrix is a regular unimodular matrix, its determinant is $+1$ or -1, and since, finally, the inverse is the quotient of the conjugate divided by the determinant, it must be formed of integers.

4. Using Reduction to Solve a Linear System with Integer Solutions

We shall now use Smith's normal form to solve linear equations with integers.

Let

$$(18.45) \qquad [D]_{m\times n} = [U]_{m\times m}\cdot[A]_{m\times n}\cdot[V]_{n\times n},$$

where $[A]_{m\times n}$ is a matrix with integer elements, $[U]_{m\times m}$ and $[V]_{n\times n}$ are regular unimodular matrices each formed of integers and obtained as shown above by the products of transposition and/or subtraction matrices, and $[D]_{m\times n}$ is a matrix formed of integers having Smith's normal form such as (18.19). In addition, let us assume that $[A]$, and hence $[D]$, are of rank r.

Theorem 18.IV

In order that the equation

$$(18.46) \qquad [A]_{m\times n}\cdot[x]_{n\times 1} = [b]_{m\times 1},$$

in which the elements of $[A]$ and $[b]$ are integers, shall have a solution $[x]$ formed of integers, it is necessary and sufficient that, having reduced $[A]$ to Smith's normal form $[D]$ by unimodular matrices $[U]$ and $[V]$, there is a matrix column $[y_r^r]$ such that

$$(18.47) \qquad [y_r^r]_{r\times 1} = [D_r^r]_{r\times r}^{-1}\cdot[([U]_{r\times r}\cdot[b]_{r\times 1})_r^r],$$

that must be formed of integers and also

$$[([U].[b])^{m-r}_{m-r}]_{(m-r)\times 1} = [0]_{(m-r)\times 1} \tag{18.48}$$

where $[D^r_r]$ is the submatrix formed by the first r lines and columns of $[D]$ in (18.19), $[([U].[b]^r_r)]$ is the submatrix of $([U].[b])$ formed by its first r lines, and $[([U].[b])^{m-r}_{m-r}]$ is the submatrix of $[U].[b]$ formed by its $m-r$ last lines. In (18.47), $[y^r_r]_{r\times 1}$ is a vector formed by the first r lines of a vector $[y]_{n\times 1}$.

Proof

We start from (18.46) and say

$$[A]_{m\times n}.[V]_{n\times n}.[V]^{-1}_{n\times n}.[x]_{n\times 1} = [b]_{m\times 1}, \tag{18.49}$$

$$[U]_{m\times m}.[A]_{m\times n}.[V]_{n\times n}.[V^{-1}]_{n\times n}.[x]_{n\times 1} = [U]_{m\times m}.[b]_{m\times 1}, \tag{18.50}$$

that is,

$$[D]_{m\times n}.[V^{-1}]_{n\times n}.[x]_{n\times 1} = [U]_{m\times m}.[b]_{m\times 1}. \tag{18.51}$$

Let us assume

$$[y]_{n\times 1} = ([V^{-1}].[x])_{n\times 1}, \tag{18.52}$$

and $[y^r_r]$ the vector formed by the first r lines of $[y]$. In accordance with Theorem 18.III, $[V^{-1}]$ is formed entirely of integers, hence (18.52) requires that, if $[x]_{n\times 1}$ is formed of integers, $[y]_{n\times 1}$ is also formed of integers. Let us expand (18.51), using the explicit form of $[D]$ given by (18.19). It follows that

(18.53)

$$[D^r_r]_{r\times r}.[([V^{-1}].[x])^r_r]_{r\times 1} + [0]_{r\times(n-r)}.[([V^{-1}].[x])^{n-r}_{n-r}]_{(n-r)\times 1} = [([U].[b])^r_r]_{r\times 1},$$

(18.54)

$$[0]_{(m-r)\times r}.[([V^{-1}].[x])^r_r]_{r\times 1} + [0]_{(m-r)\times(n-r)}.[([V^{-1}].[x])^{n-r}_{n-r}]_{(n-r)\times 1} = [([U].[b])^{m-r}_{m-r}]_{(m-r)\times 1}.$$

That is, again using (18.61),

$$[D^r_r]_{r\times r}.[y^r_r]_{r\times 1} = [([U].[b])^r_r]_{r\times 1}, \tag{18.55}$$

$$[0]_{(m-r)\times 1} = [([U].[b])^{m-r}_{m-r}]_{(m-r)\times 1}. \tag{18.56}$$

If $[x]$ is formed of integers, then $[y]$ obtained from (18.52) is also formed of integers. If $[x]$ satisfies (18.46), then $[y]$ satisfies (18.55) and (18.56). Premultiplying the two members of (18.55) by $[D^r_r]^{-1}$, we obtain (18.47). Equation (18.56) is identical with (18.48). Hence equations (18.47) and (18.48) are necessary if (18.46) is to have an integer solution.

Now, if $[y]$ satisfies (18.47) and (18.48), it satisfies (18.55) and (18.56). We now calculate $[x]$ by

$$(18.57)\qquad [x]_{n\times 1} = [V]_{n\times n}\cdot[y]_{n\times 1}\,.$$

From the manner in which we obtained (18.55) and (18.56) beginning with (18.46), $[x]$, which is given by (18.57), satisfies (18.46). Hence Theorem 18.IV provides sufficient conditions.

Example

Let us take the following linear system:

$$(18.58)\qquad \begin{aligned} x_1+x_2+x_3-4x_4+x_5 &= -6,\\ x_1+x_2+x_3+3x_4-2x_5 &= 2,\\ 2x_1+2x_3-x_4-x_5 &= 8. \end{aligned}$$

The matrix of the coefficients of the left member is the one given in (18.26).

Examining (18.42), we see that

$$(18.59)\qquad [D_r]_{3\times 3} = \begin{bmatrix} 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 2 \end{bmatrix},$$

that is

$$(18.60)[1]\qquad [D_r]^{-1}_{3\times 3} = \begin{bmatrix} 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1/2 \end{bmatrix},$$

$$(18.61)\qquad [U]_{3\times 3} = \begin{bmatrix} 1 & 0 & 0\\ -2 & 0 & 1\\ 13 & 1 & -7 \end{bmatrix},$$

$$(18.62)\qquad [b] = \begin{bmatrix} -6\\ 2\\ 8 \end{bmatrix}.$$

$$(18.63)[2]\qquad \underset{[y_r]}{\begin{bmatrix} y_1\\ y_2\\ y_3 \end{bmatrix}} = \underset{[D_r]^{-1}}{\begin{bmatrix} 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1/2 \end{bmatrix}} . \underset{([U].[b])_r}{\begin{bmatrix} 1 & 0 & 0\\ -2 & 0 & 1\\ 13 & 1 & -7 \end{bmatrix} . \begin{bmatrix} -6\\ 2\\ 8 \end{bmatrix}}$$

$$= \begin{bmatrix} 1 & 0 & 0\\ -2 & 0 & 1\\ 13/2 & 1/2 & -7/2 \end{bmatrix} . \begin{bmatrix} -6\\ 2\\ 8 \end{bmatrix} = \begin{bmatrix} -6\\ 20\\ -66 \end{bmatrix}.$$

Since y_1, y_2, and y_3 have integer values, in accordance with Theorem 18.IV the system possesses at least one integer solution.

From (18.57) we now have

$$[x]_{n\times 1} = [V]_{n\times n}\cdot[y]_{n\times 1}\,. \tag{18.64}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 1 & 1 & -2 & 5 & -1 \\ 0 & 3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 3 & 0 \\ 0 & 0 & -3 & 7 & 0 \end{bmatrix} . \begin{bmatrix} -6 \\ 20 \\ -66 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} 146+5y_4-y_5 \\ -6 \\ y_5 \\ 86+3y_4 \\ 198+7y_4 \end{bmatrix} . \tag{18.65}$$

Or again,

$$\begin{aligned} x_1 &= 146+5y_4-y_5\,, \\ x_2 &= -6\,, \\ x_3 &= y_5\,, \\ x_4 &= 86+3y_4\,, \\ x_5 &= 198+7y_4\,. \end{aligned} \tag{18.66}$$

There are an infinity of integer solutions resulting from integer values for y_4 and y_5. For instance, let us make

$$y_4 = 1\,, \qquad y_2 = 2\,, \tag{18.67}$$

whence we obtain

$$x_1 = 149\,, \quad x_2 = -6\,, \quad x_3 = 2\,, \quad x_4 = 89\,, \quad x_5 = 205\,. \tag{18.68}$$

By substituting (18.66) in (18.58) we can confirm that (18.58) is identically satisfied and, obviously, the particular case of it chosen as an example (18.68).

5. Smith's Reduced Form for an Integer Matrix

Matrix $[D]$ of rank (18.19), obtained by reduction to Smith's normal form by transforming matrix $[A]$ and such that

$$[U]_{m\times m}\cdot[A]_{m\times n}\cdot[V]_{n\times n} = [D]_{m\times n}\,, \tag{18.69}$$

where $[U]$ and $[V]$ are unimodular and regular, is formed by elements

$$d_k \neq 0\,, \qquad k = 1, 2, \ldots, r\,, \qquad r \leqslant \min(m,n)\,, \tag{18.70}$$

[1] In this example $r = m$. If $r < m$, we take the submatrix $[D_r^r]_{r\times r}$ in $[D]_{m\times m}$.

[2] In this example $r = m$, so that $([U].[b])_r^r = [U].[b]$. If $r < m$, we take the submatrix formed by the r first lines of the vector column $[U].[b]$.

such that[1]

(18.71) $$|d_1| \leqslant |d_2| \leqslant\ , \dots,\ \leqslant |d_r|.$$

This reduction to Smith's normal form applies to matrices $[A]$, the elements of which belong to $\mathbf{Z}$ (see 18.3). Let us now enunciate a theorem that will prove very useful later on for Gomory's cuts and for asymptomatic programming with integers.

Theorem 18.V

For every matrix $[A]_{m\times n}$, the elements of which are related integers and which is of rank r, there are two regular unimodular matrices $[U]_{m\times m}$ and $[V]_{n\times n}$ with integer coefficients that give a special case of Smith's normal form $[\Delta]_{m\times n}$ such that

(18.72) $$[U].[A].[V] = [\varDelta],$$

where

(18.73) $$[\varDelta]_{m\times n} = \begin{array}{cc} \left[\begin{array}{cccc|ccc} \delta_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \delta_2 & \dots & 0 & 0 & \dots & 0 \\ & & \ddots & \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \delta_r & 0 & & 0 \\ \hline 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \end{array}\right] & \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} r \\ \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} m-r \end{array} \\ \begin{array}{cc} \underbrace{\hphantom{\delta_1\quad 0\quad \dots\quad 0}}_{r} & \underbrace{\hphantom{0\quad \dots\quad 0}}_{n-r} \end{array} & \end{array}$$

with the property

(18.74) $$\delta_i \text{ divides } \delta_{i+1}, \qquad i = 1, 2, \dots, r-1.$$

Matrix $[\Delta]$ is called *Smith's reduced matrix* of $[A]$, and is unique.

The δ_i are called the *elementary divisors* of $[A]$.

Proof

Let us now see how we can obtain Smith's reduced form $[\Delta]_{m\times n}$ from Smith's normal form $[D]_{m\times n}$.

Let d_i, $i = 1, 2, \dots, r$, be the nonnull elements of $[D]$, and let us now consider a pair (d_k, d_l) among the $C_r^2 = r(r-1)/2$ pairs of d_i numbers. Let us assume that $k < l$, that is, $|d_k| \leqslant |d_l|$.

Let us define as

(18.75) G.C.D. (d_k, d_l) the greatest common divisor of the absolute values of d_k and d_l,

[1] If, at the end of the calculations, (18.71) has not been satisfied, it is sufficient to pre- and/or postmultiply $[D]$ of the appropriate permutation matrices.

(18.76) L.C.M. (d_k, d_l) the least common multiple of the absolute values of d_k and d_l.

Let us now make use of a very well-known arithmetical theorem known as *Bezout's theorem*: given two integers a and b, then there are always two integers λ and μ such that

(18.77) $$\lambda a + \mu b = \text{G.C.D.}\ (a, b).$$

Thus, if $a = 10$ and $b = 15$, giving a G.C.D. $(10, 15) = 5$, we have

(18.78) $$(-1).10+(1).15 = 5.$$

As another example, if $a = -14$, $b = 18$, the G.C.D. is $(-14, 18) = 2$, so that we have

(18.79) $$(-4)(-14)+(-3).18 = 2.$$

Applying this theorem we shall construct two matrices $[G]_{m\times m}$ and $[H]_{n\times n}$ such that

(18.80) $$[G]_{m\times m}\cdot[D]_{m\times n}\cdot[H]_{n\times n} = [\Delta]_{m\times n},$$

where Δ has the properties of (18.73) and (18.74), that is to say, possesses Smith's reduced form.

Let us consider a pair (d_k, d_l) chosen from the d_i of $[D]$ and let us construct a regular unimodular matrix $[G]_{m\times m}$ as follows:

(18.81)
$$g_{kk} = 1, \quad g_{kl} = 1, \quad g_{lk} = -\frac{\mu d_l}{\text{G.C.D.}(d_k, d_l)}, \quad g_{ll} = \frac{\lambda d_k}{\text{G.C.D.}(d_k, d_l)},$$

where λ and μ are Bezout integers such as (18.77). While

(18.82) $$g_{ij} = 0, \qquad i \neq j;\ i \neq k;\ j \neq l;\ i, j = 1, 2, \ldots, m$$

and

(18.83) $$g_{ij} = 1, \qquad i = j;\ i \neq k;\ j \neq l;\ i, j = 1, 2, \ldots, m.$$

Let us observe that we have

(18.84) $$\begin{vmatrix} g_{kk} & g_{kl} \\ g_{lk} & g_{ll} \end{vmatrix} = g_{kk}\cdot g_{ll} - g_{kl}\cdot g_{lk} = \frac{\lambda d_k + \mu d_l}{\text{G.C.D.}(d_k, d_l)} = 1,$$

in accordance with (18.77); which implies that $[G]$ is unimodular with a determinant $(+1)$. In a similar manner, let us construct a regular unimodular matrix $[H]_{n\times n}$ as follows:

(18.85)
$$h_{kk} = \lambda, \quad h_{kl} = \frac{-d_l}{\text{G.C.D.}(d_k, d_l)}, \quad h_{lk} = \mu, \quad h_{ll} = \frac{d_k}{\text{G.C. D.}(d_k, d_l)}.$$

While

$$(18.86) \qquad h_{ij} = 0, \qquad i \neq j\,;\; i \neq k, j \neq k\,;\; i, j = 1, 2, \ldots, n,$$

and

$$(18.87) \qquad h_{ij} = 1, \qquad i = j\,;\; i \neq k\,;\; j \neq k\,;\; i, j = 1, 2, \ldots, n.$$

Let us observe that we have

$$(18.88) \qquad \begin{vmatrix} h_{kk} & h_{kl} \\ h_{lk} & h_{ll} \end{vmatrix} = h_{kk}.h_{ll} - h_{kl}.h_{lk} = \frac{\lambda d_k + \mu d_l}{\text{G.C.D.}\,(d_k,\, d_l)} = 1,$$

still because of (18.77), which implies that $[H]$ is equally regular unimodular with a determinant of $(+1)$.

Now, if we consider the submatrices contained in the lines and columns in which the elements d_k and d_l appear, we can say

$$(18.89) \qquad \begin{bmatrix} g_{kk} & g_{kl} \\ g_{lk} & g_{ll} \end{bmatrix} . \begin{bmatrix} d_k & 0 \\ 0 & d_l \end{bmatrix} . \begin{bmatrix} h_{kk} & h_{kl} \\ h_{lk} & h_{ll} \end{bmatrix}$$

$$= \begin{bmatrix} g_{kk}\cdot h_{kk}\cdot d_k + g_{kl}\cdot h_{kl}\cdot d_l & g_{kk}\cdot h_{kl}\cdot d_k + g_{kl}\cdot h_{ll}\cdot d_l \\ g_{lk}\cdot h_{kk}\cdot d_k + g_{ll}\cdot h_{lk}\cdot d_l & g_{lk}\cdot h_{kl}\cdot d_k + g_{ll}\cdot h_{ll}\cdot d_l \end{bmatrix}.$$

Substituting (18.81) and (18.85) in (18.89) we obtain

$$(18.90) \qquad \begin{bmatrix} \lambda d_k + \mu d_l & 0 \\ 0 & \dfrac{\lambda d_k^2\, d_l + \mu d_k d_l^2}{[\text{G.C.D.}\,(d_k,\, d_l)]^2} \end{bmatrix}.$$

The element (1, 1) of (18.90) is the G.C.D., in accordance with (18.77). Now let us consider the element (2, 2). An elementary theorem gives

$$(18.91) \qquad \text{L.C.M.}\,(a,\, b).\text{G.C.D.}\,(a,\, b) = a \times b.$$

Hence we can say

$$(18.92) \qquad \frac{\lambda d_k^2\, d_l + \mu d_k d_l^2}{[\text{G.C.D.}\,(d_k,\, d_l)]^2} = \frac{(\lambda d_k + \mu d_l)}{\text{G.C.D.}\,(d_k,\, d_l)} . \frac{d_k \cdot d_l}{\text{G.C.D.}\,(d_k,\, d_l)}$$

$$= \text{L.C.M.}\;(d_k,\, d_l).$$

Hence matrix (18.90) is expressed

$$(18.93) \qquad \begin{bmatrix} \text{G.C.D.}\,(d_k,\, d_l) & 0 \\ 0 & \text{L.C.M.}\,(d_k,\, d_l) \end{bmatrix}.$$

And finally we can say

$$(18.94)\qquad \begin{bmatrix} 1 & 1 \\ -\dfrac{\mu d_l}{\text{G.C.D.}(d_k, d_l)} & \dfrac{\lambda d_k}{\text{G.C.D.}(d_k, d_l)} \end{bmatrix} \cdot \begin{bmatrix} d_k & \\ 0 & d_l \end{bmatrix}$$

$$\times \begin{bmatrix} \lambda & \dfrac{-d_l}{\text{G.C.D.}(d_k, d_l)} \\ \mu & \dfrac{d_k}{\text{G.C.D.}(d_k, d_l)} \end{bmatrix}$$

$$= \begin{bmatrix} \text{G.C.D.}(d_k, d_l) & 0 \\ 0 & \text{L.C.M.}(d_k, d_l) \end{bmatrix}.$$

Thus, the transformation produced by a premultiplication by a matrix $[G]$, defined by (18.81)–(18.83), and by a postmultiplication by a matrix $[H]$, defined by (18.85)–(18.87), replaces d_k by $\delta_k = \text{G.C.D.}\ (d_k, d_l)$ and d_l by $\delta_l = \text{L.C.M.}\ (d_k, d_l)$.

This procedure ensures that the sequences $d_1, d_2, \ldots, d_r$ can be replaced, step by step, by sequences $\delta_1, \delta_2, \ldots, \delta_r$. But, because of the property revealed in (18.94), this sequence finally becomes such that

(18.95) $\qquad l > k \Rightarrow \delta_l$ is a multiple of δ_k in absolute value.[1]

Example

Let us take a matrix with Smith's normal form $[D]_{4\times 3}$ obtained from a matrix $[A]_{4\times 3}$ and such that

$$(18.96)\qquad [D] = \begin{bmatrix} 2 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{bmatrix}.$$

We have $d_1 = 2, \quad d_2 = -3, \quad d_3 = 4.$

Let us first consider the pair (d_1, d_2). The G.C.D. is $(2, -3) = 1$, so that we can say

$$\lambda(2) + \mu(-3) = 1 \quad \text{is satisfied by } \lambda = 2 \text{ and } \mu = 1,$$

[1] Need we recall that the G.C.D. is always a submultiple of the L.C.M.? Our readers will remember this from their school days.

that is,

$$(2)\,(2)+(1)\,(-3)=1.$$

Let us construct the matrices $[G_{12}]$ and $[H_{12}]$ in conformity with (18.81)–(18.83), on the one hand, and with (18.85)–(18.87), on the other, that is to say, for the submatrices formed by tne lines and columns 1 and 2, and by taking account of (18.94),

(18.97)

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 3 & 4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 & 3 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -6 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{bmatrix}.$$

In the right member of (18.97), let us consider the pair (d_2, d_3). We have

$$\text{G.C.D.}\ (-6, 4) = 2.$$

And we can say

$$\lambda(-6)+\mu(4)=2 \quad \text{is satisfied by } \lambda=-1 \text{ and } \mu=-1.$$

Let us construct the matrices $[G_{23}]$ and $[H_{23}]$ in accordance with (18.94):

(18.98)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & -6 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & -2 \\ 0 & -1 & -3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -12 \\ 0 & 0 & 0 \end{bmatrix}.$$

The result gives a Smith's reduced form; δ_3 is divisible by δ_2 and δ_2 by δ_1. Let us now calculate matrices $[G]$ and $[H]$, which allows the right number of (18.98) to pass from (18.96):

(18.99)

$$[G] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 2 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 0 & 0 \\ 3 & 4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 3 & 4 & 1 & 0 \\ 6 & 8 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$(18.100)\qquad [H] = \begin{bmatrix} 2 & 3 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & -2 \\ 0 & -1 & -3 \end{bmatrix} = \begin{bmatrix} 2 & -3 & -6 \\ 1 & -2 & -4 \\ 0 & -1 & -3 \end{bmatrix} .$$

And we can finally confirm that

$$(18.101)\qquad \underset{[G]}{\begin{bmatrix} 1 & 1 & 0 & 0 \\ 3 & 4 & 1 & 0 \\ 6 & 8 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}} \cdot \underset{[D]}{\begin{bmatrix} 2 & 0 & 0 \\ 0 & -3 & 0 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{bmatrix}} \cdot \underset{[H]}{\begin{bmatrix} 2 & -3 & -6 \\ 1 & -2 & -4 \\ 0 & -1 & -3 \end{bmatrix}}$$

$$= \underset{[\Delta]}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & -12 \\ 0 & 0 & 0 \end{bmatrix}} .$$

First Observation

In Smith's reduced form every pair (δ_k, δ_l) is such that $\delta_k = \text{G.C.D.}\ (\delta_k, \delta_l)$ and $\delta_l = \text{L.C.M.}\ (\delta_k, \delta_l)$. Let us note that, by construction, all the δ_i, $i = 1, 2, \ldots, r-1$, are positive, but δ_r, the last, can be positive or negative.

Second Observation

Since both determinants of $[G]$ and $[H]$ are equal to $(+1)$, we have, in accordance with (18.80),

$$(18.102)\qquad \det [\Delta] = \det [D] = \delta_1 . \delta_2 \ \ldots \ \delta_r .$$

Third Observation

By considering (18.94) we see that if the elements d_i, $i = 1, 2, \ldots, r$, of $[D]$ are first, taken two by two, the method employed here to obtain Smith's reduced form gives

$$(18.103)\qquad \delta_1 = \delta_2 = \ldots = \delta_{r-1} = 1 \quad \text{and} \quad \delta_r = \det [D] .$$

For instance, the elements of (18.96) are not all first, taken two by two (2 and 4 are not first between them), so that the method employed does not satisfy (18.103). On the other hand, the reader can confirm that (18.103) is satisfied by the following example:

$$(18.104)\qquad [D]=\begin{bmatrix}1&0&0&0\\0&3&0&0\\0&0&7&0\\0&0&0&8\\0&0&0&0\end{bmatrix}$$

gives

$$(18.105)\qquad [\Delta]=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&168\\0&0&0&0\end{bmatrix}.$$

Section 19. Gomory's Method for Solving Integer Programs

1. Introduction to the Method

In Section 17 we saw that the Dantzig–Manne method of integer programming does not always lead to convergence. We have already introduced the important concept of the cut into it, that is to say, supplementary constraints that are not satisfied by a solution with real numbers. In 1959 R. E. Gomory [K42] produced the basis of a method and theory that he has ever since continued to improve (see [K40], [K41], [K44]). Nevertheless, it appears that his method has not yet been able to achieve the results for computer calculations obtained by methods of direct search for which widespread commercial programs now exist. But, by using the concept of Gomory's cut as an exclusion criterion in direct methods, some very interesting variants are obtained [K59]. In this section, therefore, we shall provide a suitably instructional presentation of Gomory's basic theory.

2. Description of the Method

Let us take for solution the following program:

$$(19.1)\qquad \begin{array}{ll}(1) & [\mathrm{MAX}]\ g=[c]'_{1\times n}\cdot[x]_{n\times 1}\,,\\ (2) & [a]_{m\times n}\cdot[x]_{n\times 1}\leqslant[b]_{m\times 1}\,,\\ (3) & [x]_{n\times 1}\in \mathbf{Z}^n\,,\\ (4) & [x]_{n\times 1}\geqslant[0]_{n\times 1}\,.\end{array}$$

We assume, in addition, that the elements of $[c]$, $[a]$, and $[b]$ are related integers.

We shall now discover that the principle of Gomory's method is similar to that of Dantzig–Manne given in Section 17. Let us first, however, solve the program while ignoring constraint (3). Two disjunctive cases can appear:

a. The solution $[x^*]$ is formed of integers, in which case the minimum sought in the program has been found.

b. The solution $[x^*]$ does not contain only integers. We shall then add new constraints or *cuts* that are not satisfied by the noninteger solutions.

The matrical relation or simplex table given in (16.8), which relates to $[x^*]$ as the basic solution, has its first line composed of submatrices that are all formed of nonnegative elements, since $[x^*]$ is an optimal solution of the linear program (19.1) in which constraint (3) has been ignored. The new constraint added to the last line of (16.8) is not satisfied by $[x^*]$. This means that the value of the basic variable in the last line is negative. We shall then proceed by dual iterations (see Section 16.2). We shall give a fully instructional explanation of the method, but before doing so we shall explain how Gomory's cuts are generated.

Let us use the notation of Section 16 and assume that at an iteration k the table of the optimal linear program is

(19.5)[1] $$[x_B]_{m\times 1} = [B]^{-1}_{m\times m}\cdot[b]_{m\times 1} - [B]^{-1}_{m\times m}\cdot[N]_{m\times n}\cdot[x_N]_{n\times 1} \,,$$

where $[B]_{m\times m}$ and $[N]_{m\times n}$ are, as in Section 16, matrices chosen from the $m+n$ columns of $[[a]_{m\times n}[I]_{m\times m}]$. Let us say

(19.6) $$[\mathscr{L}]_{m\times n} = [B]^{-1}_{m\times m}\cdot[N]_{m\times n} \,,$$

and let us identify as $\bar{a}_{ij}$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$, the element of line i and of column j of $[\mathscr{L}]$; let us identify as x_{B_i}, $i = 1, 2, \ldots, m$, the element of line i of $[x_B]$ and as x_{N_j}, $j = 1, 2, \ldots, n$, the element of line j of $[x_N]$. Finally let us give the notation $\bar{b}_i$, $i = 1, 2, \ldots, m$, to the element of line i of matrix $[B]^{-1}.[b]$. Equation (19.5) can then be expressed as

(19.7)[2] $$x_{B_i} = \bar{b}_i - \sum_{j=1}^{n} \bar{a}_{ij} x_{N_j} \quad , \qquad i = 1, 2, \ldots, m \,.$$

Having stated this, let us now recall that we use the term *equivalence* or *congruence* modulo p between two numbers $u \in \mathbf{R}$ and $v \in \mathbf{R}$, shown as $\simeq$ or $=$ modulo p, for the following property: $u = v$ modulo p if u and v have the same remainder by p, $p \in \mathbf{N}_0$, from their division by p, $\mathbf{N}_0$ representing the set of positive integers.

[1] Equation numbers (19.2)–(19.4) omitted in the French edition.

[2] Let us recall the notation in Section 16. That of x_{B_i} means that we are considering the ith component of the vector of the basic variables $[x_B]$; x_{N_j} means that we are considering the jth component of the vector of variables that do not belong to the basis $[x_N]$.

In order that all the x_{B_i} of (19.7) are to be integers, we must have

$$\bar{b}_i - \sum_{j=1}^{n} \bar{a}_{ij} . x_{N_j} = 0, \quad \text{modulo } 1, \qquad i = 1, 2, \ldots, m . \tag{19.8}$$

Let us use the notation

$$\langle \bar{a}_{ij} \rangle = \text{largest integer less than } \bar{a}_{ij}, \qquad \begin{matrix} i = 1, 2, \ldots, m , \\ j = 1, 2, \ldots, n . \end{matrix} \tag{19.9}$$

Again let us assume

$$\{ \bar{a}_{ij} \} = \bar{a}_{ij} - \langle \bar{a}_{ij} \rangle, \qquad \begin{matrix} i = 0, 1, 2, \ldots, m , \\ j = 1, 2, \ldots, n , \end{matrix} \tag{19.10}$$

where, obviously, $0 \leqslant \{ \bar{a}_{ij} \} < 1$.

Hence, to express these notations numerically,

If $\bar{a}_{ij} = 2.35$: $\langle \bar{a}_{ij} \rangle = 2$ and $\{ \bar{a}_{ij} \} = 0.35$.

If $\bar{a}_{ij} = -0.45$: $\langle \bar{a}_{ij} \rangle = -1$ and $\{ \bar{a}_{ij} \} = 0.55$.

By considering (19.10) we can then express (19.8) in the following manner:

$$\{ \bar{b}_i \} + \langle \bar{b}_i \rangle - \sum_{j=1}^{n} \{ \bar{a}_{ij} \} \cdot x_{N_j} - \sum_{j=1}^{n} \langle \bar{a}_{ij} \rangle \cdot x_{N_j} = 0, \quad \text{modulo } 1, \qquad i = 1, 2, \ldots, m . \tag{19.11}$$

But the $\langle b_i \rangle$, the $\langle a_{ij} \rangle$ and the x_{N_j} are integers that can be eliminated from (19.11), and we are left with

$$\{ \bar{b}_i \} - \sum_{j=1}^{n} \{ \bar{a}_{ij} \} . x_{N_j} = 0, \quad \text{modulo } 1, \qquad i = 1, 2, \ldots, m . \tag{19.12}$$

This constitutes a necessary and sufficient condition for all the x_{B_i}, $i = 1, 2, \ldots, m$, to be integers, but is difficult to satisfy. Let us see how we can obtain one that is more easily satisfied but that will not, unfortunately, be sufficient.

By definition

$$0 \leqslant \{ \bar{a}_{ij} \} < 1 , \qquad i = 1, 2, \ \ldots, m, \qquad j = 1, 2, \ldots, n$$

and

$$x_{N_j} \geqslant 0 , \qquad j = 1, 2, \ldots, n .$$

From this we deduce

$$\sum_{j=1}^{n} \{ \bar{a}_{ij} \} . x_{N_j} \geqslant 0 . \tag{19.13}$$

Let us suppose that the component x_{B_i} of $[x_B]$ is not integer, in other words,

$$1 > \{ \bar{b}_i \} > 0 . \tag{19.14}$$

In that case the left member of (19.12) can only be a positive integer in accordance with (19.13). We thus have the following necessary condition that is not satisfied for $x_{N_j} = 0, j = 1, 2, \ldots, n$, in accordance with (19.14):

$$\{\bar{b}_i\} - \sum_{j=1}^{n} \{\bar{a}_{ij}\}.x_{N_j} \leqslant 0, \qquad i = 1, \ , \ldots, m. \tag{19.15}$$

If this necessary condition still appears too difficult to obtain, we can take a necessary subcondition, namely, a single line only of (19.15) or, in other words, choose one i from this $i = 1, 2, \ldots, m$.

An inequation such as (19.15) for a value of i, $i = 1, 2, \ldots, m$, constitutes what we call a *Gomory cut.*

It remains for us to show how, by a sequential addition of such Gomory constraints or cuts to the given linear program, we converge toward an integer solution of (19.1).

Before doing so, however, we shall give a numerical example that converges toward an integer solution when it is solved by Gomory's method.

3. Examples

Given the linear program in integers:

$$\begin{array}{ll} (1) & [\text{MAX}]\ g = 3x_1 - x_2, \\ (2) & 3x_1 - 2x_2 \leqslant 3, \\ (3) & -5x_1 - 4x_2 \leqslant -10, \\ (4) & 2x_1 + x_2 \leqslant 5, \\ (5) & x_1, x_2 \in \mathbf{Z}, \\ (6) & x_1, x_2 \geqslant 0. \end{array} \tag{19.16}$$

Let us first solve this linear program without its constraint of integrity (5). Let us assume that an optimum has been obtained by the dual-simplex method described in Section 16.

If $u_1, u_2, u_3 \geqslant 0$ are the deviation variables of constraints (2), (3), and (4) of (19.16), the initial table will be as follows:

(19.17)

	x_1	x_2	u_1	u_2	u_3	Second member	Basic variables
(0)	-3	1	0	0	0	$g = 0$	
(1)	3	-2	1	0	0	3	u_1
(2)	-5	-4	0	1	0	-10	u_2
(3)	2	1	0	0	1	5	u_3

Let us observe that the determinant of the basis matrix is equal to 1.

In Fig. 19.1 we have shown the convex domain related to the above-mentioned constraints.

The solution of the program without constraint (5) carried out by the dual-simplex method provides the following table:

(19.18)

	x_1	x_2	u_1	u_2	u_3	Second member	Basic variable
(0)	0	0	5/7	0	3/7	$g = 4\frac{2}{7}$	
(1)	1	0	1/7	0	2/7	$1\frac{6}{7}$	x_1
(2)	0	0	-3/7	1	3 1/7	$4\frac{3}{7}$	u_2
(3)	0	1	-2/7	0	3/7	$1\frac{2}{7}$	x_2

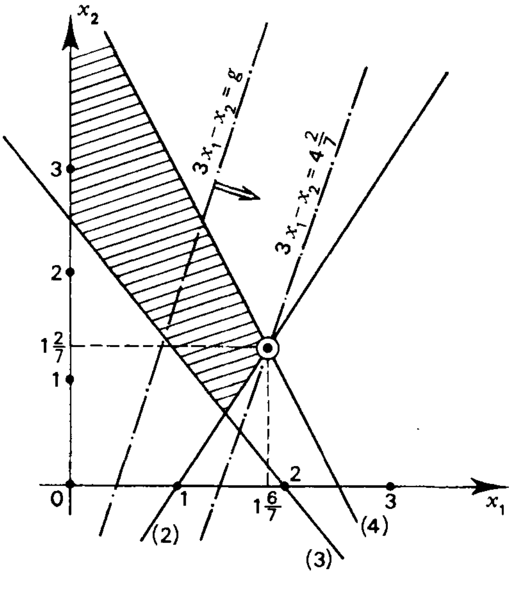


FIG. 19.1

The optimal solution of (19.16) ignoring constraint (5) gives

(19.19) $$x_1 = 1\ 6/7, \quad x_2 = 1\ 2/7,$$
$$u_1 = 0, \quad u_2 = 4\ 3/7, \quad u_3 = 0, \quad \max g = 4\ 2/7.$$

The point $[x_1\ x_2] = [1\ 6/7\ 1\ 2/7]$ is shown in Fig. 19.1; it does not represent an integer solution.

We shall now generate a Gomory cut, choosing for it in table (19.18) the expression of x_1 as a function of the variables that are not in the basis, that is, u_1 and u_3. We have

$$x_1 = 1\,6/7 - 1/7\,u_1 - 2/7\,u_3 . \tag{19.20}$$

With the notation used in (19.7)–(19.15) this gives

$$\langle 1\,6/7\rangle = 1, \qquad \langle 1/7\rangle = 0, \qquad \langle 2/7\rangle = 0, \tag{19.21}$$

and

$$\begin{aligned} \{1\,6/7\} &= 1\,6/7 - \langle 1\,6/7\rangle = 1\,6/7 - 1 = 6/7, \\ \{1/7\} &= 1/7 - \langle 1/7\rangle = 1/7 - 0 = 1/7, \\ \{2/7\} &= 2/7 - \langle 2/7\rangle = 2/7 - 0 = 2/7. \end{aligned} \tag{19.22}$$

Hence we have as constraint,

$$\{1\,6/7\} - \{1/7\}\,u_1 - \{2/7\}\,u_3 \leqslant 0, \tag{19.23}$$

that is,

$$6/7 - 1/7\,u_1 - 2/7\,u_3 \leqslant 0,$$

or again,

$$6 - u_1 - 2u_3 \leqslant 0; \tag{19.24}$$

or yet again,

$$u_1 + 2u_3 \geqslant 6. \tag{19.25}$$

Let us transform this constraint into another in which the variables x_1 and x_2 will appear, a procedure that is always possible, since any variable that is not in a basis can always be expressed as a function of the variables in that basis, this being implicit in the principle of the simplex method. From (19.17) we obtain

$$u_1 = 3 - 3x_1 + 2x_2, \tag{19.26}$$

with

$$u_3 = 5 - 2x_1 - x_2. \tag{19.27}$$

If we now substitute (19.26) and (19.27) in (19.25) we obtain

$$u_1 + 2u_3 = 3 - 3x_1 + 2x_2 + 10 - 4x_1 - 2x_2 = 13 - 7x_1 \geqslant 6, \tag{19.28}$$

and, finally,

$$x_1 \leqslant 1. \tag{19.29}$$

This constraint is a Gomory cut that we shall add to the program. It is equivalent to the constraint given below in which a new deviation variable $u_4 \geqslant 0$

has been introduced. We shall use (19.29) to represent it in Fig. 19.2. But a different expression of the same constraint given in (19.30) and also in (19.23) will be used for the tables of the dual-simplex method.

(19.30) $$-1/7u_1 - 2/7u_3 + u_4 = -6/7.$$

Constraint (19.29) has been shown in Fig. 19.2 and reduces the domain of possible solutions.

Let us, however, proceed with the use of the dual-simplex method that we are combining here with Gomory's cuts, Figs. 19.1 and 19.2 simply being given to illustrate what is taking place.

Let us, accordingly, complete table (19.18) by introducing the new constraint (19.30) as well as its associated deviation variable u_4. It follows

(19.31)

	x_1	x_2	u_1	u_2	u_3	u_4	Second member	Basic variables	
(0)	0	0	5/7	0	3/7	0	$g = 4\frac{2}{7}$		
(1)	1	0	1/7	0	2/7	0	$1\frac{6}{7}$	x_1	
(2)	0	0	-3/7	1	3 1/7	0	$4\frac{3}{7}$	u_2	
(3)	0	1	-2/7	0	3/7	0	$1\frac{2}{7}$	x_2	
(4)	0	0	-1/7	0	-2/7	1	-6/7	u_4	←
					↑				

This table does not provide a possible solution, but its dual gives one. We shall use the algorithms for the dual-simplex method explained in Section 16 and will pivot on element $(-2/7)$ of the fifth line, since

$$\left(\frac{3/7}{-2/7}\right) < \left(\frac{5/7}{-1/7}\right).$$

Making the new basis clear, we now obtain

(19.32)

	x_1	x_2	u_1	u_2	u_3	u_4	Second member	Basic variables	
(0)	0	0	1/2	0	0	3/2	$g = 3$		
(1)	1	0	0	0	0	1	1	x_1	
(2)	0	0	$\textcircled{-2}$	1	0	11	-5	u_2	←
(3)	0	1	-1/2	0	0	1/2	0	x_2	
(4)	0	0	1/2	0	1	-7/2	3	u_3	
			↑						

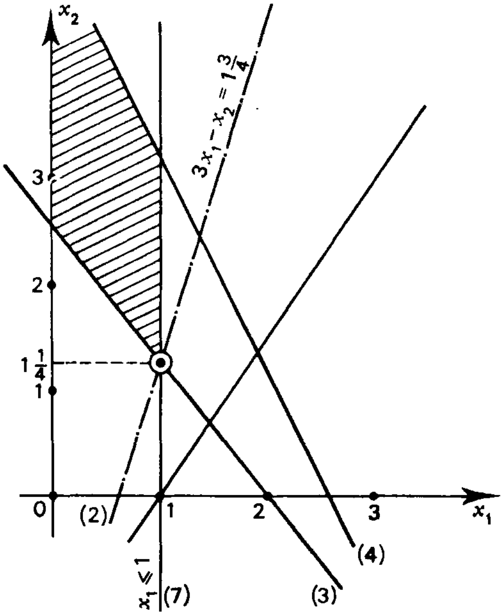


FIG. 19.2

Table (19.32) does not give a solution since $u_2 = -5 < 0$. The choice of pivot for a dual-simplex iteration is -2 at the intersection of the line and column indicated by arrows. We now obtain table (19.33).

(19.33)

	x_1	x_2	u_1	u_2	u_3	u_4	Second member	Basic variables
(0)	0	0	0	1/4	0	$4\frac{1}{4}$	$g = 1\frac{3}{4}$	
(1)	1	0	0	0	0	1	1	x_1
(2)	0	0	1	-2/4	0	$-5\frac{2}{4}$	$2\frac{2}{4}$	u_1
(3)	0	1	0	-1/4	0	$-1\frac{1}{4}$	$1\frac{1}{4}$	x_2
(4)	0	0	0	1/4	1	-3/4	$1\frac{3}{4}$	u_3

We now have a possible optimal solution:

(19.34) $x_1 = 1, \quad x_2 = 1\ 1/4,$

$$u_1 = 2\ 2/4, \quad u_2 = 0, \quad u_3 = 1\ 3/4, \quad u_4 = 0; \qquad \max g = 7/4.$$

This solution is represented in Fig. 19.2. The point $[x_1\ x_2] = [1\ 1\ 1/4]$ is not an integer solution.

We shall now generate a new Gomory cut choosing, in table (19.33), the expression of u_3 as a function of the variables that are not in the basis, namely,

u_2 and u_4.

(19.32) $$u_3 = 1\ 3/4 - 1/4\ u_2 + 3/4\ u_4,$$

which gives

(19.33) $$\langle 1\ 3/4 \rangle = 1, \quad \langle 1/4 \rangle = 0, \quad \langle -3/4 \rangle = -1,$$

and

(19.34) $$\{1\ 3/4\} = 1\ 3/4 - \langle 1\ 3/4 \rangle = 1\ 3/4 - 1 = 3/4,$$

(19.35) $$\{1/4\} = 1/4 - \langle 1/4 \rangle = 1/4 - 0 = 1/4,$$

(19.36) $$\{-3/4\} = -3/4 - \langle -3/4 \rangle = -3/4 - (-1) = 1/4.$$

We have as a constraint

(19.37) $$\{1\ 3/4\} - \{1/4\}\ u_2 - \{1/4\}\ u_4 \leqslant 0 \quad \text{or} \quad 3/4 - 1/4u_2 - 1\ 4u_4 \leqslant 0,$$

or again

(19.38) $$3 - u_2 - u_4 \leqslant 0,$$

or even

(19.39) $$u_2 + u_4 \geqslant 3.$$

To discover to which constraint in x_1 and x_2 this corresponds in Fig. 19.3 we have to express u_2 and u_4 as a function of these two variables. From (19.17) we obtain first,

(19.40) $$u_2 = -10 + 5x_1 + 4x_2.$$

From (19.32) we obtain

(19.41) $$u_4 = -7/3 + 1/3u_2 + 4/3u_3;$$

But from (19.17) we also have

(19.42) $$u_3 = 5 - 2x_1 - x_2.$$

Let us substitute (19.42) in (19.41) and the result in (19.39); then (19.40) in (19.39), and we obtain

(19.43) $$\begin{aligned} u_2 + u_4 &= -10 + 5x_1 + 4x_2 - 7/3 + 1/3u_2 + 4/3u_3 \\ &= -10 + 5x_1 + 4x_2 - 7/3 + 1/3\ (-10 + 5x_1 + 4x_2) \\ &\quad + 4/3\ (5 - 2x_1 - x_2) \\ &= -9 + 4x_1 + 4x_2 \end{aligned}$$

Substituting this result in (19.39), it follows that

(19.44) $$(8) \quad x_1 + x_2 \geqslant 3.$$

This constraint is a Gomory cut and will be added to the program, being indicated by (8) in Fig. 19.3.

Let us now consider constraint (19.37) and let us introduce the deviation variable $u_5 \geqslant 0$. We have

(19.45) $$-1/4u_2 - 1/4u_4 \leqslant -3/4,$$

that is,

(19.46) $$-1/4u_2 - 1/4u_4 + u_5 = -3/4.$$

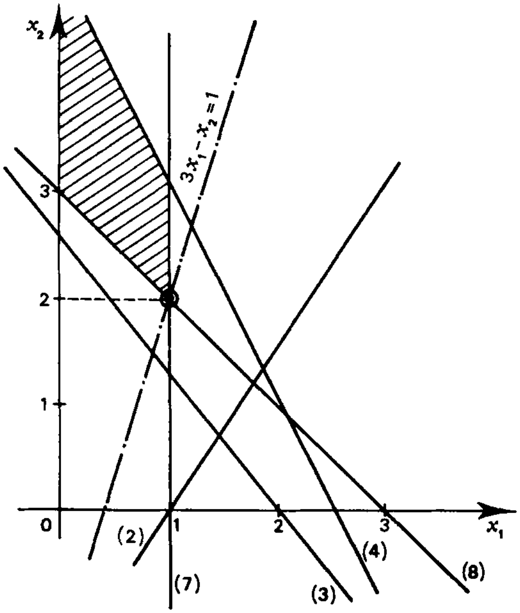


FIG. 19.3

By introducing this new constraint and this new variable in table (19.33), we obtain

(19.47)

	x_1	x_2	u_1	u_2	u_3	u_4	u_5	Second member	Basic variables	
(0)	0	0	0	1/4	0	$4\frac{1}{4}$	0	$g = 1\frac{3}{4}$		
(1)	1	0	0	0	0	1	0	1	x_1	
(2)	0	0	1	-2/4	0	$-5\frac{2}{4}$	0	$2\frac{2}{4}$	u_1	
(3)	0	1	0	-1/4	0	$-1\frac{1}{4}$	0	$1\frac{1}{4}$	x_2	
(4)	0	0	0	1/4	1	-3/4	0	$1\frac{3}{4}$	u_3	
(5)	0	0	0	-1/4	0	-1/4	1	-3/4	u_5	←
				↑						

The corresponding point in table (19.47) is not a solution, since $-3/4$ is a negative value in the column representing the second member. Let us now pass to the following dual table by pivoting on $-1/4$ in line (5). We obtain

(19.48)

	x_1	x_2	u_1	u_2	u_3	u_4	u_5	Second member	Basic variables
(0)	0	0	0	0	0	1	4	$g = 1$	
(1)	1	0	0	0	0	0	1	1	x_1
(2)	0	0	1	0	0	2	-5	4	u_1
(3)	0	1	0	0	0	1	-1	2	x_2
(4)	0	0	0	0	1	-1	-1	1	u_3
(5)	0	0	0	1	0	-4	1	3	u_2

As the optimal solution of this new program, we obtain

$$(19.48)\qquad \begin{aligned} &x_1 = 1, \quad x_2 = 2, \\ &u_1 = 4, \; u_2 = 3, \; u_3 = 1, \; u_4 = u_5 = 0; \\ &\max g = 1. \end{aligned}$$

This time the optimal program corresponds to a solution in which all the x_1 and x_2 variables are integers. Our calculations are completed.

In Fig. 19.4 we have indicated by heavy dots all the possible integer solutions of the program. All these solutions lie in the interior of or on the periphery of a convex polyhedron (in this particular case it is a polygon and there are none in the interior). This convex polyhedron is sometimes called the *convex shell* of the integer solutions and is obviously a subset of the convex domain of the linear program given without the condition of integrity.

We shall now study the properties of cuts, and to do this a recapitulation of various algebraic concepts is required, at least for some of our readers.

4. Concept of Algebraic Modulus and Recapitulation of the Properties of Groups

We shall now recall some of the properties of the most important structures in the theory of sets. The examples given will apply to finite sets, but the properties are equally true of infinite sets.

Let us consider a set $\mathbf{E}$ and an operation $*$ that, to every pair $(x, y) \in \mathbf{E} \times \mathbf{E}$, makes correspond an element $z \in \mathbf{E}$; this operation $*$ is then said to be an internal operation.

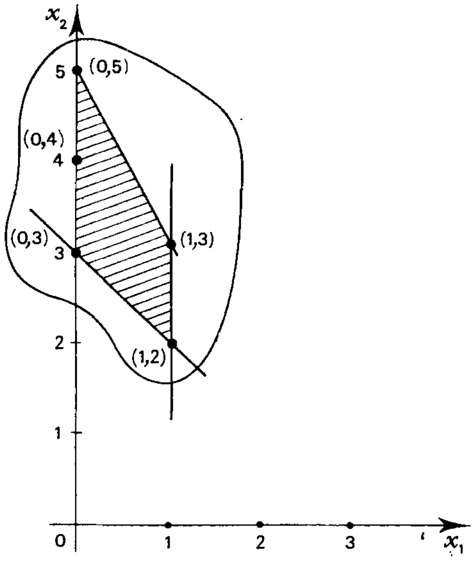


FIG. 19.4

Closure

The internal law $*$ is said to be closed if to every pair $(x, y) \in \mathbf{E} \times \mathbf{E}$ there corresponds one and only one $z \in \mathbf{E}$.

Unit to the Left

A set $\mathbf{E}$, for which an internal law $*$ has been defined, possesses a *unit to the left* if a particular element $e_G \in \mathbf{E}$ exists, such that

$$\text{(19.49)} \qquad \forall a \in \mathbf{E}: \qquad e_G * a = a\,.$$

Unit to the Right

A set $\mathbf{E}$, for which an internal law $*$ has been defined, possesses a *unit to the right* if a particular element $e_D \in \mathbf{E}$ exists such that

$$\text{(19.50)} \qquad \forall a \in \mathbf{E}: \qquad a * e_D = a\,.$$

Unit Element

A set $\mathbf{E}$, for which an internal law $*$ has been defined, possesses a unit if a particular element $e \in \mathbf{E}$ exists that is both a unit to the left and a unit to the right; that is to say, if we have

$$\text{(19.51)} \qquad \forall a \in \mathbf{E}: \qquad e * a = a * e = a\,.$$

It is easy to prove that if a unit exists it is always unique.

Associativity

A law $*$ defined for a set $\mathbf{E}$ is associative if

$$\text{(19.52)} \qquad \forall a, b, c \in \mathbf{E}: \qquad (a * b) * e = a * (b * c)\,.$$

Inverse

If a law $*$ defined for a set **E** has a unit element e and if, for every $a \in \mathbf{E}$, there is one and only one $b \in \mathbf{E}$ such that

$$(19.53) \qquad a * b = b * a = e,$$

we say that b is the *inverse* or *symmetrical*[1] of a often referring to it as $\bar{a}$ or a^{-1}. We say that the law has an inverse for each of its elements if this property is satisfied.

Commutativity or Abelian Property

A law $*$ defined for **E** is said to be *commutative* or *abelian* if

$$(19.54) \qquad \forall (a,b) \in \mathbf{E} \times \mathbf{E} : \qquad a * b = b * a .$$

The element b will not be called the symmetrical of a unless (19.53) is applicable.

Let us now give a brief recapitulation of some important structures.

Groupoid

A set **E** defined throughout by a law $*$ is called a *groupoid.* We can also say that in this case the law is closed.

Modulus

A modulus of which the law is associative is called a *monoid* or *semigroup.*

Group

A monoid in which every element has an inverse is termed a *group.*

Abelian Group

A group that possesses the property of symmetry (19.54) is called *abelian.*

Figure 19.6, which follows, summarizes these properties.

Figures 19.7–19.11 provide examples of these structures in the form of exceptions that will provide the reader with material for reflection.

Among the moduluses and groups we shall be particularly interested in those concerned with sets of real numbers with operations $*$ that constitute modulus n additions. Let us, therefore, first describe such structures and begin by considering the set of related integers:

$$(19.55) \qquad \mathbf{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}.$$

Let there then be

$$(19.56) \qquad n \in \mathbf{N}_0, \qquad r \in \mathbf{N}, \qquad a, b, q, q' \in \mathbf{Z}.$$

Two numbers a and b are called *modulo n equivalents* or *modulo n congruents* if their difference is divisible by n or, which amounts to the same thing, if their division by n produces the same nonnegative remainder r. Let

$$(19.57) \qquad a = n.q+r$$

[1] The term symmetrical is one of those most frequently used in mathematics with the most diverse meanings that are sometimes ambiguous and even contradictory in relation to each other.

and

$$b = n.q' + r, \tag{19.58}$$

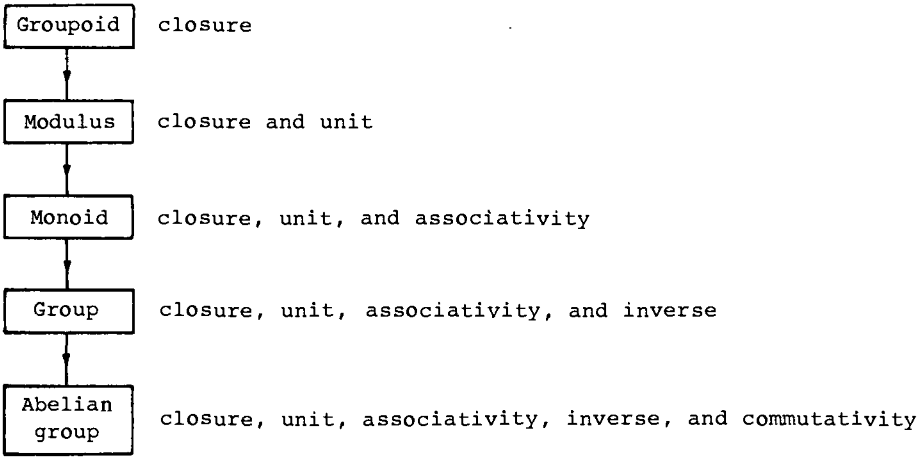

Groupoid
closure
Modulus
closure and unit
Monoid
closure, unit, and associativity
Group
closure, unit, associativity, and inverse
Abelian group
closure, unit, associativity, inverse, and commutativity

FIG. 19.6

*	*a*	*b*	*c*	*d*	*f*
a	*c*	*c*	*b*	*d*	*f*
b	*d*	*a*	*b*	*b*	*a*
c	*b*	*d*	*d*	*c*	*b*
d	*d*	*f*	*b*	*a*	*c*
f	*f*	*a*	*b*	*c*	*b*

FIG. 19.7. Groupoid that is not a modulus.

*	*e*	*a*	*b*	*c*	*d*
e	*e*	*a*	*b*	*c*	*d*
a	*a*	*c*	*c*	*a*	*b*
b	*b*	*b*	*b*	*e*	*b*
c	*c*	*c*	*d*	*b*	*e*
d	*d*	*d*	*b*	*d*	*a*

FIG. 19.8. Modulus that is not a monoid.

*	*e*	*a*	*b*	*c*	*d*
e	*e*	*a*	*b*	*c*	*d*
a	*a*	*a*	*c*	*c*	*d*
b	*b*	*c*	*b*	*c*	*d*
c	*c*	*c*	*c*	*c*	*d*
d	*d*	*d*	*d*	*d*	*d*

FIG. 19.9. Monoid that is not a group.

*	e	a	b	c	d	f
e	e	a	b	c	d	f
a	a	b	e	d	f	c
b	b	e	a	f	c	d
c	c	f	d	e	b	a
d	d	c	f	a	e	b
f	f	d	c	b	a	e

FIG. 19.10. Group that is not commutative. (It should be noted that there cannot be a noncommutative finite group with less than five elements.)

*	e	a	b	c	d
e	e	a	b	c	d
a	a	c	e	d	b
b	b	e	d	a	c
c	c	d	a	b	e
d	d	b	c	e	a

FIG. 19.11. Commutative group.

then

$$a-b = n(q-q'). \tag{19.59}$$

We can say

$$a \simeq b \pmod{n}. \tag{19.60}$$

+	{0}
{0}	{0}

mod 1

+	{0}	{1}
{0}	{0}	{1}
{1}	{1}	{0}

mod 2

+	{0}	{1}	{2}
{0}	{0}	{1}	{2}
{1}	{1}	{2}	{0}
{2}	{2}	{0}	{1}

mod 3

+	{0}	{1}	{2}	{3}
{0}	{0}	{1}	{2}	{3}
{1}	{1}	{2}	{3}	{0}
{2}	{2}	{3}	{0}	{1}
{3}	{3}	{0}	{1}	{2}

mod 4

+	{0}	{1}	{2}	{3}	{4}
{0}	{0}	{1}	{2}	{3}	{4}
{1}	{1}	{2}	{3}	{4}	{0}
{2}	{2}	{3}	{4}	{0}	{1}
{3}	{3}	{4}	{0}	{1}	{2}
{4}	{4}	{0}	{1}	{2}	{3}

mod 5

. . . .

FIG. 19.12

We place in the same class called the *modulo n residual class* all the numbers of the form $a+kn$, where a, $K \in \mathbf{Z}$ and $n \in \mathbf{N}_0$. Hence there are n classes and the set quotient $\mathbf{Z}/\mathcal{R}$, where $\mathcal{R}$ is the modulo n relation of equivalence considered, is indicated by $\mathbf{Z}/n$. These are

$$\begin{array}{llll} & \text{class } 0: & a_0 = 0 & (\text{mod } n), \\ & & a_1 = 1 & (\text{mod } n), \\ (19.61) & & a_2 = 2 & (\text{mod } n), \\ & & \cdots\cdots\cdots & \\ & & a_{n-1} = n-1 & (\text{mod } n). \end{array}$$

The set quotient $\mathbf{Z}/n$ includes n elements, each of which represents the set of the elements of the same equivalence class. Thus, taking $n = 7$, for example,

(19.62)

	0	1	2	3	4	5	6
......	...	...	...	...	...	...	...
$k = -2$	-14	-13	-12	-11	-10	-9	-8
$k = -1$	-7	-6	-5	-4	-3	-2	-1
$k = 0$	0	1	2	3	4	5	6
$k = 1$	7	8	9	10	11	12	13
$k = 2$	14	15	16	17	18	19	20
......	...	...	...	...	...	...	...

The designation of a class by the choice of the representative corresponding to the representative $k = 0$ is arbitrary; we can take any value of k, but it must obviously be the same for all the classes.

Each of the equivalence classes for the modulo n addition forms a commutative group and examples of these groups are given here for $n = 1, 2, 3, 4$, and 5. The representative of a class will be designated by a number enclosed by braces $\{\ \}$.

Cyclic Group[1]

The term *cyclic group of order n* is used for a group $(\mathbf{E}, *)$ of which all the elements $x \in \mathbf{E}$ can be obtained as

$$\begin{array}{lll} (19.63) & x_1 & = 0, \\ & x_2 & = a * x_1 = a, \end{array}$$

[1] Not to be confused with the concept of a cycle in a substitution class nor with that in the theory of graphs.

$$\begin{aligned}
x_3 &= a * x_2 = a * a, \\
&\dots \\
x_{n-1} &= a * x_{n-2} = \underbrace{a * a * \dots * a}_{n-1}, \\
x_n &= a * x_{n-1} = \underbrace{a * a * \dots * a}_{n} = 0, \\
x_{n+1} &= a * x_n = a. \\
&\dots
\end{aligned}$$

The number a is called the generator and the unit element is 0.

Let us consider an example in which $*$ is the operation $+$, that is say, *modulo 1 addition* (Fig. 19.13), and another example, that of the nonnegative integers less than 10 and divisible by 3, namely,

$$\{0, 3, 6, 9\}. \tag{19.63a}$$

$\overset{1}{+}$	0	1/6	2/6	3/6	4/6	5/6
0	0	1/6	2/6	3/6	4/6	5/6
1/6	1/6	2/6	3/6	4/6	5/6	0
2/6	2/6	3/6	4/6	5/6	0	1/6
3/6	3/6	4/6	5/6	0	1/6	2/6
4/6	4/6	5/6	0	1/6	2/6	3/6
5/6	5/6	0	1/6	2/6	3/6	4/6

FIG. 19.13. Modulo 1 addition.

This is a cyclic group in relation to modulo 12 addition and is also a subgroup of $(\mathbf{E}, *)$, where $\mathbf{E} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$, $*$ = add. mod 12.

$\overset{12}{+}$	0	3	6	9
0	0	3	6	9
3	3	6	9	0
6	6	9	0	3
9	9	0	3	6

FIG. 19.14. Modulo 12 addition.

Figure 19.14 shows the cyclic group formed by (19.63) for the modulo 12 addition.

All cyclic groups are commutative, that is to say abelian, since they are generated as shown in (19.62) and since we always have

$$\begin{aligned} x_i * x_j &= x_{i+j}, \bmod n, \qquad i, j = 0, 1, 2, \dots, n, \\ x_j * x_i &= x_{j+1}, \bmod n, \end{aligned} \tag{19.64}$$

whence

$$x_i * x_j = x_j * x_i. \tag{19.65}$$

Extension of Modulo n Equivalence Classes to Set **R**

What we have just shown for modulo n ($n \in \mathbf{N}_0$) equivalence classes in set **Z** is easily extended to set **R**. We say that two numbers $a, b \in \mathbf{R}$ are modulo n equivalents, that is to say,

$$a \simeq b \text{ modulo } n \qquad \text{(also expressed } a = b \text{ modulo } n) \tag{19.66}$$

if a and b have the same remainder r when divided by n. Thus,

$$\dots \simeq -2.63 \simeq -0.63 \simeq 1.37 \simeq 3.37 \simeq 5.37 \simeq \dots \text{ modulo } 2$$

$$\dots -1.518 \simeq -0.518 \simeq 0.482 \simeq 1.482 \simeq 2.482 \simeq \dots \text{ modulo } 1.$$

All the properties enunciated for **Z** are to be found in **R** and, in particular, the presence of the groups and especially the cyclic groups. We have used **Z** above in order to provide a clearer illustration.

5. Gomory's Cuts

In the second subdivision of this section we explained how to obtain a particular Gomory cut. We shall now give a general definition and proceed to show that this definition of the set of cuts that have been obtained, starting with a noninteger solution and employing (19.8), invests this set with the structure of an abelian group. We shall also show that the number of separate cuts in this group is ($\det[B] - 1$) where $\det[B]$ is the determinant of the base matrix of the simplex table corresponding to a noninteger solution. Thus, there are $7-1=6$ cuts that can be obtained starting from table (19.18). It is useful to know that the set of cuts forms a group so as to be able to choose "good cuts" that will quickly produce an integer solution. The explanation that we are giving differs from Gomory's but is, we believe, more instructional.

We shall use the following notation, already introduced in (19.10) above:

$$\{k\} = k - \langle k\rangle, \tag{19.67}$$

and we shall say that two real numbers A and B are *modulo 1 equivalents* if

$$\{A\} = \{B\}, \tag{19.68}$$

which means that the remainders from their division by 1 are the same.

With this notation retained, let us recall (19.5), which specifies that, for the optimal table, we must have

$$(19.69) \qquad [x_B]_{m\times 1} = [B]^{-1}_{m\times m}\cdot[b]_{m\times 1} - [B]^{-1}_{m\times m}\cdot[N]_{m\times n}\cdot[x_N]_{n\times 1}\,.$$

The element in line i of column $[x_B]_{m\times 1}$ will have the notation x_{B_i}, that of line i of matrix $[B]_{m\times m}$ will be $[B]_i^{-1}$, so that a line i of (19.69) becomes

$$(19.70) \qquad x_{B_i} = ([B]_i^{-1})_{1\times m}\cdot[b]_{m\times 1} - ([B]_i^{-1})_{1\times m}\cdot[N]_{m\times n}\cdot[x_N]_{n\times 1}\,.$$

If the optimal solution is to be integer we must have

$$(19.71) \qquad \forall i \in \{1, 2, \ldots, m\} : \qquad x_{B_i} \text{ integer}.$$

In particular, any linear combination formed by related integers as coefficients of equations such as (19.70), must give an integer. Thus,

$$(19.72) \qquad [\alpha]_{1\times m}, \qquad \alpha_i \in \mathbf{Z}; \quad i = 1, 2, \ldots, m,$$

is a matrix line of coefficients. We must have

$$(19.73) \qquad [\alpha]_{1\times m}\cdot[x_B]_{m\times 1} = \text{a related integer}.$$

If we consider, for example, table (19.18) we must have

$$(19.74) \qquad x_1 = 1\ 6/7 - 1/7u_1 - 2/7u_3 = \text{an integer number},$$

$$(19.75) \qquad u_2 = 4\ 3/7 + 3/7u_1 - 3\ 1/7u_3 = \text{an integer number}.$$

For instance, if we take the linear combination[1] of coefficients 4 and 10 we should have

(19.76)

$$\begin{aligned}
4x_1 + 10u_2 &= (4).(1\ 6/7 - 1/7u_1 - 2/7u_3) + (10)\ (4\ 3/7 + 3/7u_1 - 3\ 1/7u_3)\\
&= (4).(1\ 6/7) + (10).(4\ 3/7)\\
&\quad + [(4).(-1/7) + (10).(3/7)]\ u_1 +\\
&\quad + [(4).(-2/7) + (10).(-3\ 1/7)]\ u_3\\
&= 51\ 5/7 + 3\ 5/7u_1 - 32\ 4/7u_3\\
&= \text{an integer}.
\end{aligned}$$

If we generate a Gomory cut beginning with this, we must have

(19.77)

$$\begin{aligned}
&\{(4).(1\ 6/7) + (10).(4\ 3/7)\}\\
&\leqslant \{-((4).(-1/7) + 10).(3/7))\}\ u_1 + \{(4).(-2/7) + (10).(-3\,1/7)\}u_3,
\end{aligned}$$

[1] In this example 4 and 10 are positive integers, but we might equally have chosen any other elements of **Z** such as (-2) and (18).

or again,

$$\{51\ 5/7\} \leqslant \{-3\ 5/7\}\ u_1 + \{32\ 4/7\}\ u_3 . \tag{19.78}$$

We have

$$\langle 51\ 5\ 7\rangle = 51, \qquad \langle -3\ 5/7\rangle = -4, \qquad \langle 32\ 4/7\rangle = 32 \tag{19.79}$$

and

$$\begin{aligned} \{51\ 5/7\} &= 51\ 5/7 - 51 = 5/7, \\ \{-3\ 5/7\} &= -3\ 5/7 - (-4) = 2/7, \\ \{32\ 4/7\} &= 32\ 4/7 - 32 = 4/7. \end{aligned} \tag{19.80}$$

Thus, (19.77) gives

$$5/7 \leqslant 2/7\ u_1 + 4/7 u_3 . \tag{19.81}$$

We must be able to say for any $[\alpha]$ that conforms to (19.72),

$$\begin{aligned} [\alpha]_{1\times m}\cdot[x_B]_{m\times 1} &= [\alpha]_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[b]_{m\times 1} \\ &\quad - [\alpha]_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[N]_{m\times n}\cdot[x_N]_{n\times 1} = \text{an integer}. \end{aligned} \tag{19.82}$$

Hence

(19.83)

$$[\alpha]_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[b]_{m\times 1} - [\alpha]_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[N]_{m\times n}\cdot[x_N]_{n\times 1} = \text{an integer}.$$

And again,

(19.84)

$$\begin{aligned} &\langle[\alpha].[B]^{-1}.[b]\rangle + \{[\alpha].[B]^{-1}.[b]\} \\ &- \langle[\alpha].[B]^{-1}.[N]\rangle.[x_N] - \{[\alpha].[B]^{-1}.[N]\}.[x_N] = \text{an integer}. \end{aligned}$$

Given that, by hypothesis, $[x_N]$, $\langle[\alpha].[B]^{-1}.[b]\rangle$ and $\langle[\alpha].[B]^{-1}.[N]\rangle$ the noninteger part of (19.84) must be such that

$$\{[\alpha].[B]^{-1}.[b]\} - \{[\alpha].[B]^{-1}.[N]\}.[x_N] = \text{an integer}. \tag{19.85}$$

From the definition of the noninteger part of a real number, we have

$$0 \leqslant \{[\alpha].[B]^{-1}.[b]\} < 1, \tag{19.86}$$

and also

$$\{[\alpha].[B]^{-1}.[N]\}.[x_N] \geqslant 0. \tag{19.87}$$

Given that (19.85) is an integer and that $[x_N] \geqslant 0$, this can only be a nonpositive integer because of (19.85) and (19.86) and of the minus sign in front of $\{[\alpha].[B]^{-1}.[N]\}.[x_N]$ in (19.85). Hence the general equation for a Gomory cut is given by

(19.88) $$\{[\alpha].[B]^{-1}.[b]\} - \{[\alpha].[B]^{-1}.[N]\}.[x_N] \leqslant 0. \;.$$

This formula generalizes (19.15).

Hence, to obtain (19.81) beginning with table (19.18), we can state

(19.89) $$[\alpha] = [4 \quad 10 \quad 0].$$

Two different Gomory cuts generally correspond to two different vectors $[\alpha^{(1)}]$ and $[\alpha^{(2)}]$.

As we noted above, for the example introduced in (19.76) we arbitrarily selected nonnegative components of $[\alpha]$, but any integer, whether positive, negative, or null, would have been suitable, as the reader can easily verify. The same Gomory cut beginning with (19.81) would have been obtained if, for example, we had chosen $[\alpha] = [4 \; -4 \; 0]$. Let us take, for instance, the cut obtained in table (19.18) for $[\alpha^{(1)}] = [1\;0\;0]$, namely (19.20),

(19.90) $$x_1 = 1\;6/7 - 1/7\;u_1 - 2/7\;u_3 = \text{an integer},$$

from which we obtained (19.29), namely,

(19.91) $$x_1 \leqslant 1.$$

Let us now calculate the cut corresponding to $[\alpha^{(2)}] = [4\;0\;0]$, that is,

(19.92) $$4x_1 = (4).(1\;6/7) - (4).(1/7)\;u_1 - (4).(2/7)\;u_3 = \text{an integer}.$$

Whence

(19.93) $$\{(4).(1\;6/7)\} \leqslant \{(4).(1/7)\}\;u_1 + \{(4).(2/7)\}\;u_3,$$

and again,

(19.94) $$3/7 \leqslant 4/7u_1 + 1/7u_3;$$

finally,

(19.95) $$4u_1 + u_3 \geqslant 3.$$

By making use of (19.26) and (19.27), which give u_1 and u_3 explicitly as functions of x_1 and x_2, we obtain

(19.96) $$\begin{aligned} 4u_1 + u_3 &= 12 - 12x_1 + 8x_2 + 5 - 2x_1 - x_2 \\ &= 17 - 14x_1 + 7x_2 \geqslant 3, \end{aligned}$$

and thus,

(19.97) $$2x_1 - x_2 \leqslant 2.$$

We can verify in Fig. 19.15 that the optimal solution $x_1 = 1$, $x_2 = 2$ satisfies constraint (19.97), whereas the solution in table (19.18), namely, $x_1 = 1\ 6/7$, $x_2 = 1\ 2/7$ does not satisfy it.

Let us observe that $[\alpha^{(1)}] = [1\ 0\ 0]$ gives the cut $x_1 \leqslant 1$ and $[\alpha^{(2)}] = [4\ 0\ 0]$ gives the cut $2x_1 - x_2 \leqslant 2$.

Still from Fig. 19.15, let us observe that the point $x_1 = 1$, $x_2 = -1$ satisfies $x_1 \leqslant 1$, whereas it does not satisfy $2x_1 - x_2 \leqslant 2$ since this Goomory cut slices off a different area of the original polyhedron of the constraints (the shaded portion in the figure). It may seem surprising to the reader that with the same constraints but with different $[\alpha]$ we can obtain different cuts. This will be explained later in this section when we shall show that these Gomory cuts form a group that is often cyclic. In such cases all the cuts of the same cyclic group can be generated by one and the same constraint.

Observation

To prove several properties connected with the abelian group of Gomory cuts it is necessary to employ various properties of modulo 1 operations. So as not to interrupt the successive stages of the very important reasoning in this section, we have given these properties in a supplement to which the reader who is less conversant with the use of modulo 1 congruences is referred.

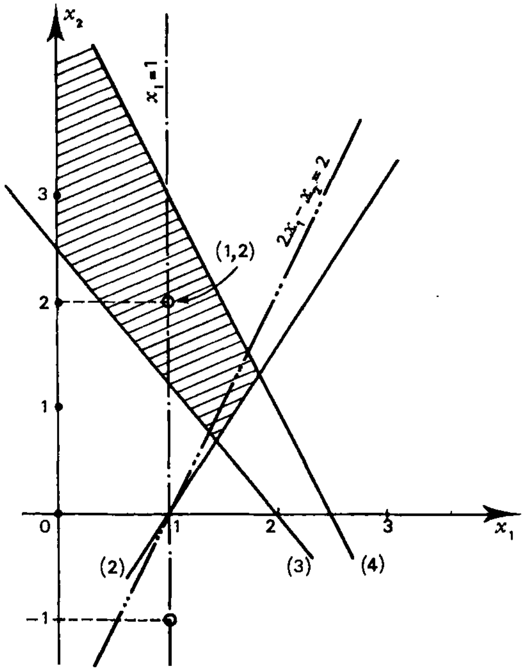


Fig. 19.15

6. Abelian Group of Gomory Cuts

Let us show how the cuts, the definition of which is given in (19.88), form an abelian group for an operation we shall now define.

Let us indicate as I a Gomory cut or inequality.

(19.98)

$$I: \quad \{[\alpha]_{1\times m}\cdot[B]_{m\times m}^{-1}\cdot[b]_{m\times 1}\} \leqslant \{[\alpha]_{1\times m}\cdot[B]_{m\times m}^{-1}\cdot[N]_{m\times n}\}\cdot[x_N]_{n\times 1}$$

and let us define an operation $*$ in the set **I** of Gomory cuts in the following manner: Given

$$(19.99) \qquad I_1: \quad \{[\alpha^{(1)}]\cdot[B]^{-1}\cdot[b]\} \leqslant \{[\alpha^{(1)}]\cdot[B]^{-1}\cdot[N]\}\cdot[x_N],$$

$$(19.100) \qquad I_2: \quad \{[\alpha^{(2)}]\cdot[B]^{-1}\cdot[b]\} \leqslant \{[\alpha^{(2)}]\cdot[B]^{-1}\cdot[N]\}\cdot[x_N].$$

we shall now define $*$ as follows:

$$(19.101) \qquad I_1 * I_2: \quad \{([\alpha^{(1)}]+[\alpha^{(2)}])\cdot[B]^{-1}\cdot[b]\}$$
$$\leqslant \{([\alpha^{(1)}]+[\alpha^{(2)}])\cdot[B]^{-1}\cdot[N]\}\cdot[x_N].$$

Hence, to carry out the operation $*$ is to construct a new cut or inequality by introducing the common sum of $[\alpha^{(1)}]$ and $[\alpha^{(2)}]$.

We shall now show that the set of cuts **I** forms an abelian group for the operation $*$ in which, that amounts to the same thing, the set of vectors $[\alpha]$ indicated by **A** forms a group for the common addition of two vectors $[\alpha^{(1)}]$ and $[\alpha^{(2)}]$. Let us recall how the vectors $[\alpha]$ were defined by (19.72):

$$(19.102) \qquad [\alpha] = [\alpha_1, \alpha_2, \dots, \alpha_m],$$

$$(19.103) \qquad \alpha_i \in \mathbf{Z}; \quad i = 1, 2, \dots, m,$$

and let us show that the set **A** of such vectors forms a group for common addition defined by

$$(19.104) \qquad [\alpha^{(1)}]+[\alpha^{(2)}] = [\alpha_1^{(1)}, \alpha_2^{(1)}, \dots, \alpha_m^{(1)}] + [\alpha_1^{(2)}, \alpha_2^{(2)}, \dots, \alpha_m^{(2)}]$$
$$= [\alpha_1^{(1)}+\alpha_1^{(2)}, \alpha_2^{(1)}+\alpha_2^{(2)}, \dots, \alpha_m^{(1)}+\alpha_m^{(2)}].$$

We shall now successively prove the following properties: closure, existence of a unit, associativity, existence of the inverse, and commutativity. This will enable us to say that (**A**, +) is a group[1] and thence that (**I**, *) is a group since it is homomorphic to (**A**, +).[2]

[1] A reader who is sufficiently advanced in the new mathematics will not require the following proofs; it will suffice to know that **Z** is a group for common addition and hence that the vectors, the elements of which belong to **Z**, also form a group for the addition of the vectors.

[2] **I** is homomorphic to **A** since, for a vector of **A** there is only one corresponding cut, but for a cut there is an infinity of correponding vectors of **A**. See [K11], Volume 1, p. 60.

Closure

A is closed for common addition since the addition of two vectors $[\alpha^{(1)}]$ and $[\alpha^{(2)}]$, defined in accordance with (19.102)–(19.104), gives an element of **A**. In effect,

$$(\alpha_i^{(1)} \in \mathbf{Z} \text{ and } \alpha_i^{(2)} \in \mathbf{Z}) \Rightarrow ((\alpha_i^{(1)}+\alpha_i^{(2)}) \in \mathbf{Z}), \qquad i = 1, 2, \ldots, m. \tag{19.105}$$

As a result **I** is closed.

Existence of a Unit

The unit of **A** for common addition is

$$[0]_{1\times m} = [0, 0, \ldots, 0]. \tag{19.106}$$

The corresponding unit of **I** will be the cut

$$0_{1\times 1} \leqslant [0]_{1\times n} \cdot [x_N]_{n\times 1}. \tag{19.107}$$

Associativity

If

$$\alpha_i^{(1)}, \alpha_i^{(2)}, \alpha_i^{(3)} \in \mathbf{Z}, \qquad i = 1, 2, \ldots, m,$$

we have

$$(\alpha_i^{(1)}+\alpha_i^{(2)}) + \alpha_i^{(3)} = \alpha_i^{(1)} + (\alpha_i^{(2)}+\alpha_i^{(3)}), \tag{19.108}$$

and thence,

$$([\alpha]^{(1)}] + [\alpha^{(2)}]) + [\alpha^{(3)}] = [\alpha^{(1)}] + ([\alpha^{(2)}] + [\alpha^{(3)}]), \tag{19.109}$$

that is also to say,

$$(I_1 * I_2) * I_3 = I_1 * (I_2 * I_3). \tag{19.110}$$

Existence of an Inverse

We have

(19.111)[1] $\forall\, \alpha_i^{(1)} \in \mathbf{Z}: \qquad \exists!\, \beta_i^{(1)} \in \mathbf{Z}$ such that

$$\alpha_i^{(1)}+\beta_i^{(1)} = \beta_i^{(1)}+\alpha_i^{(1)} = 0, \qquad i = 1, 2, \ldots, m.$$

Let

$$\beta_i^{(1)} = -\alpha_i^{(1)}. \tag{19.112}$$

Thus to every vector $[\alpha^{(1)}] \in \mathbf{A}$, there corresponds a vector $[-\alpha^{(1)}] \in \mathbf{A}$ that is its sole inverse.

[1] Let us recall the meaning of the symbols $\forall, \exists, \exists!$; $\forall$: for every; $\exists$: there is one; $\exists!$: there is one and only one.

The inverse of

$$(19.113) \qquad [\alpha^{(1)}] = [\alpha_1^{(1)}, \alpha_2^{(1)}, \ldots, \alpha_m^{(1)}]$$

is

$$(19.114) \qquad [-\alpha^{(1)}] = [-\alpha_1^{(1)}, -\alpha_2^{(1)}, \ldots, -\alpha_m^{(1)}].$$

We shall make I_1^{-1} correspond to $[-\alpha^{(1)}]$ to indicate the inverse cut of cut I_1. That is to say

$$(19.115) \qquad I_1 \ : \quad \{[\alpha^{(1)}].[B]^{-1}.[b]\} \leqslant \{[\alpha^{(1)}].[B]^{-1}.[N]\}.[x_N]$$

$$(19.116) \qquad I_1^{-1} : \quad \{[-\alpha^{(1)}].[B]^{-1}.[b]\}$$
$$\leqslant \{[-\alpha^{(1)}].[B]^{-1}.[N]\}.[x_N].$$

Commutativity

We have

$$(19.117) \qquad \forall \alpha_i^{(1)}, \alpha_i^{(2)} \in \mathbf{Z} : \quad \alpha_i^{(1)}+\alpha_i^{(2)} = \alpha_i^{(2)}+\alpha_i^{(1)}; \quad i = 1, 2, \ldots, m,$$

hence

$$(19.118) \qquad \forall [\alpha^{(1)}], [\alpha^{(2)}] \in \mathbf{A} : \quad [\alpha^{(1)}] + [\alpha^{(2)}] = [\alpha^{(2)}] + [\alpha^{(1)}],$$

and thence,

$$(19.119) \qquad I_1 * I_2 = I_2 * I_1.$$

Thus, $(\mathbf{A}, +)$ and $(\mathbf{I}, *)$ are homomorphic abelian groups.

Number of Elements Existing in the Group of the Cuts

Let us now consider how to find the number of elements of group $\mathbf{I}$ of Gomory's cuts, mainly because this will give us an idea of the degree of non-integrity of the associated linear program at each stage of the process and will also provide useful information about the difficulty of the problem to be solved in integers. In this part we shall make use of the results given in Section 18 with regard to Smith's reduced form.

Let us see the number of different cuts of type (19.88) that can be engendered from all the vectors

$$(19.120) \qquad [\alpha] = [\alpha_1, \alpha_2, \ldots, \alpha_m] \in \mathbf{Z}^m.$$

We know that there are two regular unimodular matrices $[U]_{m\times m}$ and $[V]_{n\times n}$ and a matrix $[\Delta]_{m\times n}$ of type (18.83) such that

$$(19.121) \qquad [U]_{m\times m}\cdot[B]_{m\times n}\cdot[V]_{n\times n} = [\Delta]_{m\times n}.$$

Now, if we consider a matrix $[B]_{m\times m}$, the relation (19.121) can be expressed as

$$(19.122) \qquad [U]_{m\times m}\cdot[B]_{m\times m}\cdot[V]_{m\times m} = [\Delta]_{m\times m}.$$

Let us at once observe that if $[B]_{m\times m}$ is regular, that is to say if it permits of an inverse, then $[\Delta]_{m\times m}$ is also regular since $[U]_{m\times m}$ and $[V]_{n\times n}$ being regular unimodular possess inverses, and the product of three regular matrices gives a regular matrix. Hence, in this case, Smith's reduced form will give a matrix $[\Delta]$ such that

$$(19.123)\qquad [\varDelta]_{m\times m} = \begin{bmatrix} \delta_1 & 0 & \dots & 0 \\ 0 & \delta_2 & \dots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \dots & \delta_m \end{bmatrix},$$

with the property

$$(19.124)\qquad \delta_i \text{ divides } \delta_{i+1}, \qquad i = 1, 2, \dots, m-1.$$

Premultiplying (19.122) by $[U]^{-1}$ and then postmultiplying the result by $[V]^{-1}$, we obtain

$$(19.125)\qquad [B] = [U]^{-1}.[\varDelta].[V]^{-1}.$$

Inverting the two members of (19.125), it follows that

$$(19.126)\qquad [B]^{-1} = [V].[\varDelta]^{-1}.[U],$$

with

$$(19.127)\qquad [\varDelta]^{-1}_{m\times m} = \begin{bmatrix} 1/\delta_1 & 0 & \dots & 0 \\ 0 & 1/\delta_2 & \dots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \dots & 1/\delta_m \end{bmatrix}.$$

In accordance with (19.88) a Gomory cut is written

$$(19.128)\qquad \{[\alpha]_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[b]_{m\times 1}\}$$
$$\leqslant \{[\alpha]_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[N]_{m\times n}\}\cdot[x_N]_{n\times 1}.$$

If we substitute (19.126) in (19.128) we obtain as the expression of a Gomory cut

$$(19.129)\qquad \{[\alpha]_{1\times m}\cdot[V]_{m\times m}\cdot[\varDelta]^{-1}_{m\times m}\cdot[U]_{m\times m}\cdot[b]_{m\times 1}\}$$
$$\leqslant \{[\alpha]_{1\times m}\cdot[V]_{m\times m}\cdot[\varDelta]^{-1}_{m\times m}\cdot[U]_{m\times m}\cdot[N]_{m\times n}\}\cdot[x_N]_{n\times 1}.$$

Now let us observe that, since $[V]$ is unimodular and formed of integers, $[V]^{-1}$ is also formed of integers and

$$(19.130)\qquad [\beta]_{1\times m} = [\alpha]_{1\times m}\cdot[V]_{m\times m}$$

is a matrix line formed of related integers. Substituting (19.130) in (19.129) we obtain

$$(19.131)\qquad \{[\beta]_{1\times m}\cdot[\Delta]^{-1}_{m\times m}\cdot[U]_{m\times m}\cdot[b]_{m\times 1}\}$$
$$\leqslant \{[\beta]_{1\times m}\cdot[\Delta]^{-1}_{m\times m}\cdot[U]_{m\times m}\cdot[N]_{m\times n}\}\cdot[x_N]_{n\times 1}\,.$$

Again, let us assume

$$(19.132)\qquad [\mathcal{U}]_{m\times 1} = [U]_{m\times m}\cdot[b]_{m\times 1}\,,$$

$$(19.133)\qquad [\mathcal{T}]_{m\times n} = [U]_{m\times m}\cdot[N]_{m\times n}\,.$$

and let us substitute (19.132) and (19.133) in (19.131)

$$(19.134)\qquad \{[\beta]_{1\times m}\cdot[\Delta]^{-1}_{m\times m}\cdot[\mathcal{U}]_{m\times 1}\}$$
$$\leqslant \{[\beta]_{1\times m}\cdot[\Delta]^{-1}_{m\times m}\cdot[\mathcal{T}]_{m\times n}\}\cdot[x_N]_{n\times 1}\,.$$

Let us further assume

$$(19.135)\qquad [\mu]_{1\times m} = [\beta]_{1\times m}\cdot[\Delta]^{-1}_{m\times m} = [\beta_1/\delta_1\,,\ \beta_2/\delta_2\,,\ \ldots\ \beta_m/\delta_m]\,.$$

Then (19.134) can be expressed

$$(19.136)\qquad \{[\mu]_{1\times m}\cdot[\mathcal{U}]_{m\times 1}\} \leqslant \{[\mu]_{1\times m}\cdot[\mathcal{T}]_{m\times n}\}\cdot[x_N]_{n\times 1}\,.$$

Let us observe that $[\mathcal{U}]_{m\times 1}$ and $[\mathscr{T}]_{m\times n}$ are matrices composed of related integers; in consequence, by using properties 6 and 8 given in the supplement[1] under references (A1.30) and (A1.34), the inequation (19.36) can be written

$$(19.137)\qquad \{\{[\mu]_{1\times m}\}\cdot[\mathcal{U}]_{m\times 1}\} \leqslant \{\{[\mu]_{1\times m}\}\cdot[\mathcal{T}]_{m\times 1}\}\cdot[x_N]_{n\times 1}\,.$$

The maximal number of cuts that can be obtained corresponds to the number of different cuts (19.137) that we can obtain for each vector

$$(19.138)\qquad \{[\mu]_{1\times m}\} = \{[\beta]_{1\times m}\cdot[\Delta]^{-1}_{m\times m}\}\,.$$

Now, there are δ_1 values of $\{\mu_1\} = \{\beta_1/\delta_1\}$, δ_2 values of $\{\mu_2\} = \{\beta_2/\delta_2\}$, ..., δ_m values of $\{\mu_m\} = \{\beta_m/\delta_m\}$. Hence we have $\delta_1.\delta_2.\ldots.\delta_m$ possible values of the vector $\{[B]_{1\times m}\cdot[\Delta]_{m\times m}\}$ and, if we exclude the vector $\{[0]_{1\times m}\}$, there are

$$|\delta_1.\delta_2.\ldots.\delta_m|-1$$

separate nonnull vectors $\{[\beta]_{1\times m}\cdot[\Delta]^{-1}_{m\times m}\}$.

Let us now return to (19.122). The matrix $[\Delta]_{m\times m}$ is regular and diagonal and its determinant

$$(19.139)\qquad \det[\Delta] = \delta_1.\delta_2.\ldots.\delta_m$$

is equal to that of $[B]_{m\times m}$ since $[U]$ and $[V]$ are regular unimodular except for the sign (since $\det[U] = 1$ or -1 and $\det[V] = 1$ or -1).

[1] So as not to overburden the text we have given various properties of modulo 1 operations as a supplement.

Thus the number of possible Gomory cuts is

(19.140)[1] $\mathcal{N} = |\det [B]| - 1,$

Let us take an example and consider the solution obtained in table (19.31). For this table we have

(19.141) $$[B]^{-1} = \begin{bmatrix} 1/7 & 0 & 2/7 \\ -3/7 & 1 & 22/7 \\ -2/7 & 0 & 3/7 \end{bmatrix}.$$

Thence,

(19.142) $$[B] = \begin{bmatrix} 3 & 0 & -2 \\ -5 & 1 & -4 \\ 2 & 0 & 1 \end{bmatrix} \qquad \text{with } \det [B] = 7.$$

By using the method giving Smith's normal form explained in Section 18, and by noticing that $[B]$ is a matrix composed of integers in consequence of which the normal form becomes a reduced one, we find

(19.143)

$$\underset{[U]}{\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 2 \end{bmatrix}} \cdot \underset{[B]}{\begin{bmatrix} 3 & 0 & -2 \\ -5 & 1 & -4 \\ 2 & 0 & 1 \end{bmatrix}} \cdot \underset{[V]}{\begin{bmatrix} 0 & 0 & 1 \\ 1 & 4 & 3 \\ 0 & 1 & -2 \end{bmatrix}} = \underset{[\Delta]}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 7 \end{bmatrix}}.$$

If we continue to refer to (19.31) we see that

(19.144) $$[b] = \begin{bmatrix} 3 \\ -10 \\ 5 \end{bmatrix}$$

and

(19.145) $$[N] = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Matrix $[b]$ is taken from table (19.17) (second column member). Matrix $[B]$, that forms the basis in table (19.31), is formed of the columns of the variables

[1] If $[M]$ is a square matrix we are free to use the symbols $|M|$ or $\det[M]$ to represent its determinant. But since the symbol $|a|$ sometimes also represents the absolute value of the number a, confusion may arise, and we have therefore given the necessary distinctions.

x_1, u_2, and x_2 in table (19.17), while $[N]$ is composed of the columns of u_1 and u_3 in that table.

It follows

$$(19.146) \qquad [\mathcal{U}] = [U].[b] = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 2 \end{bmatrix} . \begin{bmatrix} 3 \\ -10 \\ 5 \end{bmatrix} = \begin{bmatrix} -10 \\ 5 \\ 13 \end{bmatrix},$$

$$(19.147) \qquad [\mathcal{C}] = [U].[N] = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 2 \end{bmatrix} . \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 2 \end{bmatrix}.$$

By using the preceding notation and by considering Smith's reduced matrix $[\Delta]$ calculated in (19.143), we have

$$(19.148) \qquad \delta_1 = 1, \qquad \delta_2 = 1, \qquad \delta_3 = 7.$$

In this particular case, from (19.133) vector $[\mu]$ is therefore expressed as

$$(19.149) \qquad [\mu] = [\beta_1/1, \beta_2/1, \beta_3/7].$$

And since $\beta_1, \beta_2, \beta_3$ are integers,

$$(19.150) \qquad [\mu] = [0 \quad 0 \quad \{\beta_3/7\}].$$

Hence the six nonnull possible vectors $\{[\mu]\}$ are

$$(19.151) \qquad [0\ 0\ 1/7], \quad [0\ 0\ 2/7], \quad [0\ 0\ 3/7],$$
$$[0\ 0\ 4/7], \quad [0\ 0\ 5/7], \quad \text{and } [0\ 0\ 6/7].$$

By replacing (19.146), (19.147), and (19.151) in the general equation (19.138) of the Gomory cuts and by observing that $x_{N_1} = u_1$ and $x_{N_2} = u_3$ in table (16.31), we obtain the group of six Gomory cuts relative to that table:

$$(19.152) \qquad \{\tfrac{1}{7}.13\} \leqslant \{\tfrac{1}{7}.1\}.x_{N_1} + \{\tfrac{1}{7}.2\}.x_{N_2}, \text{ that is, } \tfrac{6}{7} \leqslant \tfrac{1}{7}u_1 + \tfrac{2}{7}u_3,$$

$$(19.153) \qquad \{\tfrac{2}{7}.13\} \leqslant \{\tfrac{2}{7}.1\}\ x_{N_1} + \{\tfrac{2}{7}.2\}.x_{N_2}, \text{ that is, } \tfrac{5}{7} \leqslant \tfrac{2}{7}u_1 + \tfrac{4}{7}u_3,$$

$$(\ 9.154) \qquad \{\tfrac{3}{7}.13\} \leqslant \{\tfrac{3}{7}\ 1\}.x_{N_1} + \{\tfrac{3}{7}.2\}.x_{N_2}, \text{ that is, } \tfrac{4}{7} \leqslant \tfrac{3}{7}u_1 + \tfrac{6}{7}u_3,$$

$$(19.155) \qquad \{\tfrac{4}{7}.13\} \leqslant \{\tfrac{4}{7}.1\}.x_{N_1} + \{\tfrac{4}{7}.2\}.x_{N_2}, \text{ that is, } \tfrac{3}{7} \leqslant \tfrac{4}{7}u_1 + \tfrac{1}{7}u_3,$$

$$(19.156) \qquad \{\tfrac{5}{7}.13\} \leqslant \{\tfrac{5}{7}.1\}.x_{N_1} + \{\tfrac{5}{7}.2\}.x_{N_2}, \text{ that is, } \tfrac{2}{7} \leqslant \tfrac{5}{7}u_1 + \tfrac{3}{7}u_3,$$

$$(19.157) \qquad \{\tfrac{6}{7}.13\} \leqslant \{\tfrac{6}{7}.1\}.x_{N_1} + \{\tfrac{6}{7}.2\}.x_{N_2}, \text{ that is, } \tfrac{1}{7} \leqslant \tfrac{6}{7}u_1 + \tfrac{5}{7}u_3.$$

We observe that cut (19.152) is the one we obtained by direct methods in (19.30).

Cyclic Group of Gomory Cuts

If, in matrix (19.123), we have: $\delta_1 = \delta_2 = \cdots = \delta_{m-1} = 1$, the $|\det[\Delta]| - 1$ vectors $\{[\mu]_{m\times 1}\}$ can be written as

$$(19.158) \qquad \{[\mu]\} = [0 \quad 0.\ldots.0 \quad \{\beta_m/\delta_m\}].$$

This case was shown in (19.150). The general form (19.138) of Gomory cuts is therefore expressed as

(19.159)

$$I: \quad \{\{\beta_m/\delta_m\}.\mathcal{U}_m\} \leqslant \{\{\beta_m/\delta_m\}.\mathcal{T}_{m1}\}.x_{N_1} + \{\{\beta_m/\delta_m\}.\mathcal{T}_{m_2}\}.x_{N_2} + \cdots$$
$$+ \{\{\beta_m/\delta_m\}.\mathcal{T}_{mn}\}.x_{N_n}.$$

The cuts are all obtained by the operation $*$ defined in (19.101) beginning with the following cut:

$$(19.160) \qquad \{\mathcal{U}_m/\delta_m\} \leqslant \{\mathcal{T}_{m1}/\delta_m\}.x_{N_1} + \{\mathcal{T}_{m2}/\delta_m\}.x_{N_2} + \cdots$$
$$+ \{\mathcal{T}_{mn}/\delta_m\}.x_{N_n}.$$

Hence the group of cuts defined in (19.159) forms a cyclic group (see the definition in (19.62)) of which cut (19.160) is a generator. In our example cut (19.152) was a generator of the cyclic group of six cuts obtained from table (16.31) as starting point.

If we recall the definition in Smith's reduced form of a matrix $[B]_{m\times m}$ and if we remember that δ_{i+1} is always divisible by δ_i, the case $\delta_1 = \delta_2 = \cdots = \delta_{m-1} = 1$ corresponds to the one where the δ_i are first among them. It should be observed that this particular case frequently occurs.

We have not given the formal proof for the convergence of Gomory's fractional method, so called because in the course of the iterations the elements in the simplex tables have fractional values. Although it does not require any concepts besides those given here this proof is a somewhat cumbersome one, and the reader is referred to the original article [K42].

7. Programming Called *All-Integer*

This method of Gomory's[1] enables us to obtain integers only in the different simplex tables for the iterations. It has the advantage of eliminating the difficulties of rounding off during the calculation on the computer.

Let us assume that a table such as (16.8) has a nonnegative first line and that all its elements are integers. Let us further assume that the column for this table given by (16.76) is not nonnegative. This means that the vector represented by this column is not a solution of the integer program (19.1). As we saw in (16.79), there is at least one element b_r of column $[\bar{b}]_{n\times 1}$ that is negative.

[1] The reader is referred to [K40] for the proof of the convergence.

Hence we can express the ith line as we did in (19.7),

$$x_{B_r} = \bar{b}_r - \sum_{j=1}^{N} \bar{a}_{rj}.x_{N_j}. \tag{19.161}$$

We shall express constraint (19.161) in another manner. To do so let us assume, with the notation of (19.9),

$$f_{rj} = \bar{a}_{rj} - \lambda\langle \bar{a}_{rj}/\lambda\rangle, \qquad j = 1, 2, \ldots, n, \qquad \lambda \in N_0. \tag{19.168^1}$$

Since $\bar{a}_{rj}$ is a related integer, f_{rj} is an integer such that

$$0 \leqslant f_{rj} < \lambda, \qquad j = 1, 2, \ldots, n. \tag{19.169}$$

Let us also assume

$$g_r = \bar{b}_r - \lambda.\langle \bar{b}_r/\lambda\rangle, \tag{19.170}$$

which results in $0 \leqslant g_r < \lambda$.

Then constraint (19.161) can be expressed

$$x_{B_r} + \sum_{j=1}^{n} f_{rj}.x_{N_j} = g_r + \lambda.\left(\langle \bar{b}_r/\lambda\rangle - \sum_{j=1}^{n} \langle \bar{a}_{rj}/\lambda\rangle.x_{N_j}\right). \tag{19.171}$$

If $[x_B]$ and $[x_N]$ are, by hypothesis, nonnegative integers, the left member of (19.171) is a nonnegative integer, since f_{rj}, $j = 1, 2, \ldots, n$, is also a nonnegative integer. Hence the right member is also a nonnegative integer. Accordingly we must have

$$\langle \bar{b}_r/\lambda\rangle - \sum_{j=1}^{n} \langle \bar{a}_{rj}/\lambda\rangle.x_{N_j} = \text{a related integer}, \tag{19.172}$$

since g_r is a related integer.

If (19.172) were not nonnegative it would be less than or equal to -1. Hence the right member of (19.171) would be less than or equal to $g_r - \lambda$ since $\lambda \geqslant 0$. Like $g_r < \lambda$ the right member would be negative. Hence we must have

$$\langle \bar{b}_r/\lambda\rangle - \sum_{j=1}^{n} \langle \bar{a}_{rj}/\lambda\rangle.x_{N_j} \geqslant 0. \tag{19.173}$$

This constraint called an *all-integer constraint*[2] is not satisfied for $x_{N_j} = 0$, $j = 1, 2, \ldots, n$.

As in Gomory's procedure explained above, we shall add this constraint to the simplex table after choosing a suitable value for λ that will ensure an all-

[1] Equation numbers (19.162)–(19.167) omitted in the French edition.

[2] Let us observe that if $\lambda = 1$ in (19.173) constraint (19.171) has the form of constraint (19.98), a fractional Gomory cut.

integer simplex table at the next iteration. Let us now explain how to make a suitable choice for λ.

On the one hand, in order that the first line of the simplex table will be non-negative at the following iteration (see (16.8)), we must, when taking line r as the pivoting line and column s as the pivoting column (see (16.81)), have

$$(19.174)\qquad \min_j \left(\frac{\bar{c}_j}{-\langle \bar{a}_{rj}/\lambda \rangle} \right) = \frac{\bar{c}_s}{-\langle \bar{a}_{rs}/\lambda \rangle}$$

where $\min_j$ is chosen from the j's such that $\langle \bar{a}_{rj}/\lambda \rangle < 0$.

On the other hand, in order that the simplex table remains integer at the next iteration, a λ must be chosen such that the pivoting element

$$(19.175)\qquad \langle \bar{a}_{rs}/\lambda \rangle = -1 .$$

We shall now refer to the transformation formulas (16.82)–(16.86) and (19.174).

Hence, by substituting (19.175) in (19.174) we still have

$$(19.176)\qquad \bar{c}_s \leqslant \frac{\bar{c}_j}{-\langle \bar{a}_{rj}/\lambda \rangle} \leqslant \bar{c}_j$$

where j is such that $\langle \bar{a}_{rj}/\lambda \rangle < 0$.

Finally, the variation of the economic function at the next iteration is

$$(19.177)\qquad \bar{c}_s . \langle \bar{b}_r/\lambda \rangle .$$

In a problem of maximization solved by the dual-simplex method the value of the economic function decreases at each iteration. Accordingly we seek, at each iteration, to maximize the absolute value of this diminution expressed by (19.177). We shall therefore take the smallest possible value for λ compatible with (19.175) and (19.176).

$$(19.178)\qquad \lambda = \max \left(-\bar{a}_{rs} , \left\langle \frac{-\bar{a}_{rj}}{\langle \bar{c}_j/\bar{c}_s \rangle} \right\rangle \right) ,$$

$$s \text{ and } j \text{ such that } \bar{a}_{rs} < 0 \text{ and } \bar{a}_{rj} < 0 , \quad j = 1, 2, \ldots , n.$$

Hence in the all-integer method we first choose column s of the pivot (see (19.176)) taking s such that

$$\bar{c}_s = \min_j \bar{c}_j \text{ and } \bar{a}_{rs} < 0 , \qquad \bar{a}_{rj} < 0 , \qquad j = 1, 2, \ldots , n ,$$

and finally choosing the value of λ to calculate the all-integer constraint by means of (19.178). Constraint (19.172), thus obtained, will be added to the simplex table and we shall pivot on the element of column s relative to this new line, an element that is always equal to (-1).

Example

Given the program

$$\begin{aligned} &1)\quad [\text{MIN}]\ z = 3x_1+8x_2\,,\\ &2)\quad 4x_1+5x_2 \geqslant 2\,,\\ &3)\quad 3x_1+7x_2 \geqslant 2\,,\\ &4)\quad x_1,\, x_2 \in \mathbf{N}\,. \end{aligned} \tag{19.179}$$

Let us change the direction of inequalities (2) and (3) and introduce deviation variables $u_1 \geqslant 0$, $u_2 \geqslant 0$, and assume, as we did in (16.88),

$$g = -z\,, \tag{19.180}$$

it then follows that

$$\begin{aligned} &1)\quad [\text{MAX}]\ g = -3x_1-8x_2\,,\\ &2)\quad -4x_1-5x_2+u_1 = -2\,,\\ &3)\quad -3x_1-7x_2+u_2 = -2\,,\\ &4)\quad x_1\,,\, x_2\,,\, u_1\,,\, u_2 \in \mathbf{N}\,. \end{aligned} \tag{19.181}$$

The first simplex table, constructed in the same way as (16.92) but omitting the columns of the artificial variables φ_1 and φ_2 (that are here of no use), will be

(19.182)

			(1)	(2)	(3)	(4)	(5)	(6)
			g	u_1	u_2	x_1	x_2	u_3
(0)	g	0	1	0	0	3	8	0
(1)	u_1	-2	0	1	0	-4	-5	0
(2)	u_2	-2	0	0	1	-3	-7	0
(3)	u_3	-1	0	0	0	$\enclose{circle}{-1}$	-2	1
		↑						

.

where line (3) and column (6) will not be introduced until the next iteration, (19.182) being the initial table.

Line (0) of table (19.182) composed of related integers is nonnegative, but the arrowed column is not nonnegative, in consequence of which the table does not provide a solution of (19.181). Let us choose line (1), the element of which in the arrowed column is negative, namely $-2 < 0$, as the line to generate an all-integer constraint (with the notation of formula (16.79) this will be $\bar{b}_1 = -2$). In consequence of what was done in (19.76), columns (4) and (5) of (19.182) are candidates for the pivoting column, that is to say, for entering the new basis, since -4 and -5 in line (1) are negative. As, in line (0), the elements corresponding to these columns are 3 and 8 and as 3 is less than 8, we shall select column (4) corresponding to x_1 as pivot in accordance with what was laid down in (19.176). We then choose λ by means of (19.178) and have

$$\begin{aligned} \lambda &= \max\left(-(-4),\ \left\langle\frac{-(-5)}{\langle 8/3\rangle}\right\rangle\right) \\ &= \max\,(4\,,\ \langle 5/2\rangle) \qquad (19.183) \\ &= \max\,(4,\ 2) = 4\,. \end{aligned}$$

Let us now express the all integer constraint of type (19.173), obtained from line (1) of table (19.182), as follows:

$$\langle -2/\lambda\rangle - \langle -4/\lambda\rangle . x_1 - \langle -5/\lambda\rangle . x_2 \geqslant 0\,. \qquad (19.184)$$

If we now introduce the value of λ obtained in (19.183), that is, $\lambda = 4$, in (19.184), we find

$$\langle -2/4\rangle - \langle -4/4\rangle . x_1 - \langle -5/4\rangle . x_2 \geqslant 0\,, \qquad (19.185)$$

that is

$$-1 + x_1 + 2x_2 \geqslant 0\,. \qquad (19.186)$$

To transform the latter into an equation we introduce a deviation variable u_3 and bring the variables to the right-hand side:

$$-1 = -x_1 - x_2 + u_3\,. \qquad (19.187)$$

This is the *all-integer* constraint introduced beforehand in table (19.182).

As we have explained, we shall therefore pivot on the element at the intersection of line (3) and column (4) that is always, as we mentioned earlier, equal to -1.

(19.188)

			(1)	(2)	(3)	(4)	(5)	(6)
			g	u_1	u_2	x_1	x_2	u_3
(0)	g	-3	1	0	0	0	2	8
(1)	u_1	2	0	1	0	0	3	-4
(2)	u_2	1	0	0	1	0	-1	-3
(3)	x_1	1	0	0	0	1	2	-1

.

Table (19.188) is optimal and represents the integer solution

(19.189)
$$\begin{aligned} u_1 &= 2, & u_2 &= 1, \\ x_1 &= 1, & x_2 &= 0, \\ u_3 &= 0, & g &= -3, \end{aligned}$$

for which line (0) of the table is nonnegative.

The corresponding solution for program (19.179) is

(19.190)
$$z = 3, \qquad x_1 = 1, \qquad x_2 = 0.$$

SUPPLEMENT. MIXED PROGRAMMING AND RECENT METHODS OF INTEGER PROGRAMMING

Section 20. **Asymptotic Programming in Integers**

1. Nature of the Asymptotic Problem

Let us take the linear program in integers:

$$(20.1) \qquad \begin{aligned} &(1)\quad [\text{MAX}]\; g = [c]'_{1\times n}\cdot[x]_{n\times 1},\\ &(2)\quad [a]'_{m\times n}\cdot[x]_{n\times 1} \leqslant [b]_{m\times 1},\\ &(3)\quad [x]_{n\times 1} \in \mathbf{N}^n. \end{aligned}$$

In the second part of Volume 1 and in Section 14 of this volume we showed that the solution of such a problem in linear programming was situated at a vertex of the polyhedron of the constraints. By contrast, in the case of linear programming in integer numbers, no theoretical a priori information is available about a subset of solutions that might include the optimal solution or solutions of (20.1). We shall show[1] that its solution has a *periodic* character for different matrices $[b]_{m\times 1}$, and we shall give a precise meaning to this term.

Let us consider the linear program obtained if we replace constraint (3) by

$$(20.2) \qquad [x]_{n\times 1} \geqslant [0]_{n\times 1}.$$

As we did in Section 16 immediately after formula (16.12), let us use $[\hat{x}_B]_{m\times 1}$ for the vector of the variables belonging to the optimal basis of this

[1] R. E. Gomory is the originator of this theory [K41] that has been further developed by Glover [K38], White [K73], and Gondran [K77]. For its practical application the reader should consult Shapiro [K65] and Thiriez [K70].

linear program, and $[\hat{x}_N]_{n\times 1}$ for the vector corresponding to the variables that do not belong to this basis. Let us take $\hat{g}$ for the maximal value of the economic function g of the linear program composed by (1) and (2) of (20.1) to which (20.2) has been added, giving the program the same form as (16.12). By keeping $[\varphi]_{m\times 1} = [0]_{m\times 1}$ in (16.13) and (16.14), we have

$$\begin{aligned}\hat{g} = \max g &= [c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[b]_{m\times 1}\\ &\quad - (-[c_N]'_{1\times n} + [c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[N]_{m\times n})\cdot[x_N]_{n\times 1}.\end{aligned} \tag{20.3}$$

$$[x_B]_{m\times 1} = [B]^{-1}_{m\times m}\cdot[b]_{m\times 1} - [B]^{-1}_{m\times m}\cdot[N]_{m\times n}\cdot[x_N]_{n\times 1}. \tag{20.4}$$

For $[x_N]_{n\times 1} \geqslant [0]_{n\times 1}$, (20.4) is the equation of a CPC (see (14.49)) the vertex of which is $[B]^{-1}.[b]$.

If $[x_B]_{m\times 1}$ has integer elements it is the optimal solution of (20.1). If not, the following method known as *asymptotic programming in integers* can be used. However, this method does not guarantee an optimal solution in all cases, and later in this section we shall give the sufficient but restrictive conditions required if the asymptotic optimal solution is to be the solution of (20.1). If it is a solution it will be optimal.

Let us therefore define as follows what is called an *asymptotic program* associated with an integer program such as (20.1):

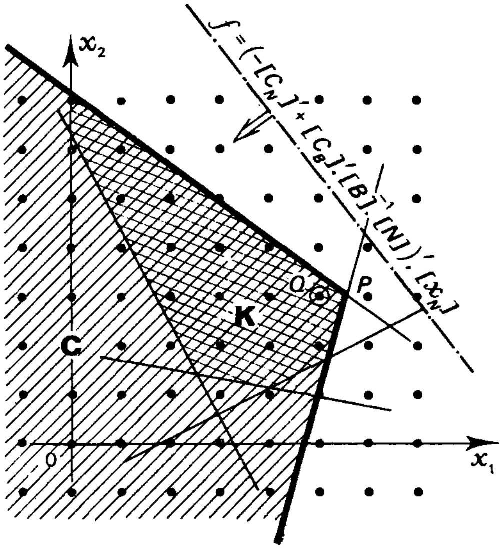
$$\begin{array}{ll} (1) & [\text{MIN}]\, f = (-[c_N]'_{1\times n} + [c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[N]_{m\times n})\\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\cdot[x_N]_{n\times 1},\\ (2) & [x_B]_{m\times 1} + [B]^{-1}_{m\times m}\cdot[N]_{m\times n}\cdot[x_N]_{n\times 1} = [B]^{-1}_{m\times m}\cdot[b]_{m\times 1},\\ (3) & [x_N]_{n\times 1} \in \mathbf{N}^n,\\ (4) & [x_B]_{m\times 1} \in \mathbf{Z}^m. \end{array} \tag{20.5}$$

Program (20.5) differs from (20.1) by the fact that $[x_B]_{m\times 1}$ is not constrained to be nonnegative. We shall now illustrate the relation existing between the solution of program (20.1) and that of its associated asymptotic program (20.5).

In Fig. 20.1 we have shown an integer program of the type of (20.1). Point P is the solution of the linear program obtained by replacing constraint (3) of (20.1) by constraint (20.2). To solve the asymptotic program associated with (20.1), namely, to solve (20.5), is to seek a point with integer values no longer strictly belonging to the convex polyhedron **K** but to the cone **C** that contains it. In Fig. 20.1 point Q, the optimal solution of the asymptotic program, belongs to **K** and is therefore an optimal solution of (20.1). By contrast, as shown in Fig. 20.2, point Q, which belongs to **C** but not to **K**, is not a solution.

Let us now express (20.5) in a different form by saying, to simplify the notation,

$$[\bar{c}_N]'_{1\times n} = -[c_N]'_{1\times n} + [c_B]'_{1\times m}\cdot[B]^{-1}_{m\times m}\cdot[N]_{m\times n}. \tag{20.6}$$



$\mathbf{K} = \{(x_1, x_2) \mid [a]'.[x] \leqslant [b], [x] \geqslant [0]\}$
$\mathbf{C} = \{(x_1, x_2) \mid [x_B] + [B]^{-1}.[N].[x_N] = [B]^{-1}.[b], [x_N] \geqslant [0]\}.$
Point P such that $[x_N] = [0]$.

FIG. 20.1

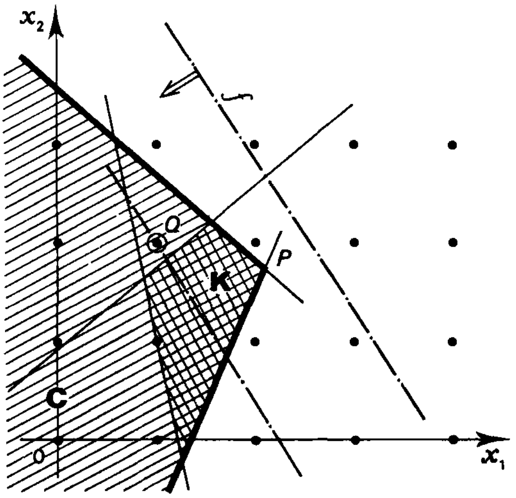


FIG. 20.2

With this notation, (20.5) becomes

$$
(20.7)\qquad
\begin{aligned}
&(1)\quad [\text{MIN}]\, f = [\bar{c}_N]'_{1\times n}\cdot[x_N]_{n\times 1},\\
&(2)\quad [B]^{-1}_{m\times m}\cdot[b]_{m\times 1} - [B]^{-1}_{m\times m}\cdot[N]_{m\times n}\cdot[x_N]_{n\times 1} = [0]_{m\times 1},\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{modulo } 1,\\
&(3)\quad [x_N]_{n\times 1}\in \mathbf{N}^n.
\end{aligned}
$$

Let $[x_N|b]$ be the optimal solution of (20.7); it will remain the same for all the vectors $[b]$ and $[b']$ such that

$$
(20.8)\qquad \{[B]^{-1}\cdot[b]\} = \{[B]^{-1}\cdot[b']\}.
$$

Accordingly we qualify it as *periodic.*[1]

We can then, after observing that

$$
(20.9)\qquad [x^*]_{(m+n)\times 1} = \begin{bmatrix} [x^*_B]_{m\times 1} \\ [x^*_N]_{n\times 1} \end{bmatrix}_{(m+n)\times 1},
$$

express the optimal solution of the asymptotic program (20.5) as

$$
(20.10)\qquad [x^*]_{(m+n)\times 1} = \begin{bmatrix} [x^*_B]_{m\times 1} \\ [x^*_N]_{n\times 1} \end{bmatrix} = \begin{bmatrix} [B]^{-1}_{m\times m}\cdot[b]_{m\times 1} \\ [0]_{n\times 1} \end{bmatrix}
\underbrace{+ \begin{bmatrix} -[B]^{-1}_{m\times m}\cdot[N]_{m\times n} \\ [1]_{n\times n} \end{bmatrix}\cdot[x_N|b]_{n\times 1}}_{}.
$$

In Eq. (20.10) we observe that the solution $[x^*]$ of the asymptotic program is the sum of the solution of the linear program obtained by replacing constraint (3) of (20.1) by constraint (20.2) and of a periodic term indicated by a horizontal bracket.

2. Solution of the Asymptotic Problem by a Method of Dynamic Programming[2]

We shall confine our explanations to the frequent case where the elements of Smith's reduced form $[\Delta]_{m\times m}$ of $[B]_{m\times m}$ are constituted by numbers that are first when taken two by two. We saw in (19.158) and beyond in Section 19 that in this case the group of Gomory cuts is cyclic. Hence, in accordance with (19.126), we can say

$$
(20.11)\qquad [B]^{-1}_{m\times m} = [V]_{m\times m}\cdot[\Delta]^{-1}_{m\times m}\cdot[U]_{m\times m}.
$$

[1] The reader can verify that $[b']_{m\quad 1} = [b]_{m\times 1}+([B]^i)_{m\times 1}$ is a solution of (20.8) where $[B]^i$ is any column of $[B]_{m\times m}$.

[2] Dynamic programming was sufficiently explained in Volume 2 to make a recapitulation unnecessary here.

Constraint (2) of (20.7) can be expressed

$$(20.12)\qquad \{[B]^{-1}_{m\times m}\cdot[b]_{m\times 1}\} = \{\{[B]^{-1}_{m\times m}\cdot[N]_{m\times n}\}\cdot[x_N]_{n\times 1}\}.$$

(Observe what was done for (19.137) where we used Properties 6 and 8 of modulo 1 operations, given in the Appendix as formulas (A1.27) and (A1.34).)

By observing that $[V]$ is formed of related integers and by using Property 8 and, finally, by substituting (20.11) in (20.12), we obtain

(20.13)

$$\{[\varDelta]^{-1}_{m\times m}\cdot[U]_{m\times m}\cdot[b]_{m\times 1}\} = \{\{[\varDelta]^{-1}_{m\times m}\cdot[U]_{m\times m}\cdot[N]_{m\times n}\}\cdot[x_N]_{n\times 1}\}.$$

If we employ the notation defined in (19.132), (19.133), and (19.127) and proceed as for (19.159), assuming

$$(20.14)\qquad \mathcal{U}_i:\ \text{element of line } i \text{ of } [\mathcal{U}]_{m\times 1},$$

$$(20.15)\qquad [\mathcal{V}]_i:\ i\text{th line of } [\mathcal{V}]_{m\times n},$$

it follows that

$$(20.16)\qquad \{\mathcal{U}_i\} = \{\{([\mathcal{V}]_i)_{1\times n}\}\cdot[x_N]_{n\times 1}\},\qquad i = 1, 2, \dots, m-1.$$

$$(20.17)\qquad \left\{\frac{\mathcal{U}_m}{\delta_m}\right\} = \left\{\left\{\frac{1}{\delta_m}\cdot([\mathcal{V}]_m)_{1\times n}\right\}\cdot[x_N]_{n\times 1}\right\}.$$

Given that matrices $[\mathcal{U}]$ and $[\mathcal{V}]$, as well as $[x_N]$, are composed of related integers in accordance with formula (3) of (20.7), if we use Property 1 given in the Appendix, we observe that constraints (20.16) are satisfied for every $[x_N]$ that satisfies (3) of (20.7).

The asymptotic program can therefore be simplified and, where the group of Gomory cuts is cyclic, can be expressed as

$$(20.18)\qquad \begin{aligned} &(1)\quad [\text{MIN}]\,f = [\bar{c}_N]'\cdot[x_N],\\ &(2)\quad \left\{\frac{\mathcal{U}_m}{\delta_m}\right\} = \left\{\left\{\frac{1}{\delta_m}\cdot[\mathcal{V}]_m\right\}\cdot[x_N]\right\},\\ &(3)\quad [x_N]\in\mathbf{N}^n. \end{aligned}$$

Let us note that (20.18) has thus assumed a form resembling that of the *problem of the knapsack* given in (2.20) of this volume and also treated from a different aspect on page 86 of Volume 2. The method given here for solving program (20.18) is described in the Appendix.

3. Examples

We shall first show that the solution of the asymptotic problem associated with problem (19.16), already treated by the method of Gomory cuts, is not a solution of (19.16).

In accordance with (19.146)–(19.148) constraint (2) of (20.18) for this problem can be expressed as

$$(20.19)\qquad \begin{aligned} \{6/7\} &= \left\{ \{1/7.[1 \quad 2]\}.\begin{bmatrix} u_1 \\ u_3 \end{bmatrix} \right\} \\ &= \left\{ \{[1/7 \quad 2/7]\}.\begin{bmatrix} u_1 \\ u_3 \end{bmatrix} \right\} \\ &= \left\{ [1/7 \quad 2/7].\begin{bmatrix} u_1 \\ u_3 \end{bmatrix} \right\} \end{aligned}$$

or again

$$(20.20)\qquad \{1/7u_1 + 2/7u_3\} = \{6/7\}.$$

Taking the value of the elements of line (0) of (19.18) in the columns of u_1 and u_3, the asymptotic program for this example is expressed as

$$(20.21)\qquad \begin{aligned} &(1)\quad [\text{MIN}]\, f = 5/7u_1 + 3/7u_3, \\ &(2)\quad \{1/7u_1 + 2/7u_3\} = 6/7, \\ &(3)\quad u_1, u_3 \in \mathbf{N}. \end{aligned}$$

Line (2) in Eq. (20.21) is of a particular type that is called *modulo 1 equation of type*

$$\left\{ \frac{a_1}{\delta} x_1 + \frac{a_2}{\delta} x_2 + \dots + \frac{a_n}{\delta} x_n \right\} = \left\{ \frac{b}{\delta} \right\},$$

for which one method of solution is given in the Appendix. Among the infinitude of solutions that an equation of this type may possess we shall select the one (or more) than minimizes (1) in (20.21). From the algorithm given in the Appendix (A1.44), we obtain

$$(20.22)\qquad u_1 = 0 \text{ and } u_3 = 3, \qquad \text{with } f = 9/7.$$

By substituting (20.22) in (19.16), it follows that

$$(20.23)\qquad x_1 = 1, \quad x_2 = 0, \quad u_1 = 0, \quad u_2 = -5, \quad u_3 = 3,$$

which is not a solution of (19.16) although it is integer for x_1 and x_2 since, the deviation variable u_2 being negative, constraint (3) of (19.16) is not satisfied.

Let us now consider another example where, by contrast, the solution of the associated asymptotic program solves the linear program in integers.

Given the linear program in integer numbers

$$(20.24)\qquad \begin{aligned} &(1)\quad [\mathrm{MAX}]\ g = 2x_1+x_2+x_3+3x_4+x_5\,,\\ &(2)\quad 2x_2+x_3+4x_4+2x_5 \leqslant 47\,,\\ &(3)\quad 3x_1-4x_2+4x_3+x_4-x_5 \leqslant 41\,,\\ &(4)\quad x_1\,,\, x_2\,,\, x_3\,,\, x_4\,,\, x_5 \in \mathbf{N}\,. \end{aligned}$$

By replacing the constraint of integrity (4) by

$$(20.25)\qquad x_1\,,\, x_2\,,\, x_3\,,\, x_4\,,\, x_5 \geqslant 0\,,$$

and by introducing the deviation variables $u_1 \geqslant 0$, $u_2 \geqslant 0$, we obtain as the optimal solution of this linear program (as the reader should be able to verify)

$$(20.26)\qquad \begin{aligned} &x_1 = 45\,,\quad x_2 = 23\ 1/2\,,\quad x_3 = x_4 = x_5 = 0\,,\\ &u_1 = 0\,,\quad u_2 = 0\,. \end{aligned}$$

This solution corresponds to the following optimal simplex table:

(20.27)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
			g	x_2	x_1	x_3	x_4	x_5	u_1	u_2
(1)	g	$113\frac{3}{6}$	1	0	0	$3\frac{3}{6}$	5	2	$1\frac{5}{6}$	4/6
(2)	x_2	$23\frac{3}{6}$	0	1	0	3/6	2	1	3/6	0
(3)	x_1	45	0	0	1	2	3	1	4/6	2/6

The optimal basis matrix $[B]_{2\times 2}$ is formed from the columns of x_1 and x_2 of (20.24)

$$(20.28)\qquad [B] = \begin{array}{c} \\ x_1 \\ x_2 \end{array}\!\!\begin{array}{c} \begin{array}{cc} x_1 & \;\;x_2 \end{array} \\ \begin{bmatrix} 0 & 2 \\ 3 & -4 \end{bmatrix} \end{array}.$$

By proceeding as explained in Section 18 (Smith's reduced form), the reader can obtain the regular unimodular matrices $[U]$ and $[V]$ that enable us to transform matrix $[B]$ into Smith's reduced form $[\Delta]$. We find

$$(20.29)\qquad \underset{[U]}{\begin{bmatrix} 0 & -1 \\ 1 & 2 \end{bmatrix}} \cdot \underset{[B]}{\begin{bmatrix} 0 & 2 \\ 3 & -4 \end{bmatrix}} \cdot \underset{[V]}{\begin{bmatrix} 1 & 4 \\ 1 & 3 \end{bmatrix}} = \underset{[\Delta]}{\begin{bmatrix} 1 & 0 \\ 0 & 6 \end{bmatrix}}\,.$$

Let us now calculate $[\mathscr{U}]_{2\times 1}$ and $[\mathscr{T}]_{2\times 5}$. We have

$$[\mathscr{U}]_{2\times 1} = [U]_{2\times 2}\cdot[b]_{2\times 1} = \begin{bmatrix}0 & -1\\ 1 & 2\end{bmatrix}\cdot\begin{bmatrix}47\\ 41\end{bmatrix} = \begin{bmatrix}-41\\ 129\end{bmatrix}, \tag{20.30}$$

$$\begin{aligned}[\mathscr{T}]_{2\times 5} &= [U]_{2\times 2}\cdot[N]_{2\times 5}\\ &= \begin{bmatrix}0 & -1\\ 1 & 2\end{bmatrix}\cdot\begin{array}{c}\begin{array}{ccccc} x_3 & x_4 & x_5 & u_1 & u_2\end{array}\\ \begin{bmatrix}1 & 4 & 2 & 1 & 0\\ 4 & 1 & -1 & 0 & -1\end{bmatrix}\end{array}\\ &= \begin{array}{c}\begin{array}{ccccc} x_3 & x_4 & x_5 & u_1 & u_2\end{array}\\ \begin{bmatrix}-4 & -1 & 1 & 0 & -1\\ 9 & 6 & 0 & 1 & 2\end{bmatrix}\end{array},\end{aligned} \tag{20.31}$$

where, as we learned in Section 19, $[N]$ represents the matrix of the columns that do not belong to the basis.

Let us consider the elements[1] in columns (4)–(8) of line (0) in (20.27) and substitute $\delta_m = \delta_2 = 6$, then $\mathscr{U}_m = \mathscr{U}_2 = 129$, then $[\mathscr{T}]_m = [\mathscr{T}]_2 = [9\ 6\ 0\ 1\ 2]$ in formula (2) of (20.18). The asymptotic program of type (20.18) corresponding to (20.24) will then be

$$\begin{aligned}&(1)\quad [\text{MIN}]\, f = (3\ 3/6)\, x_3 + 5x_4 + 2x_5 + (1\ 5/6)\, u_1 + 4/6 u_2,\\ &(2)\quad \{129/6\} = \left\{\{1/6\ [9\ \ 6\ \ 0\ \ 1\ \ 2]\}\cdot\begin{bmatrix}x_3\\ x_4\\ x_5\\ u_1\\ u_2\end{bmatrix}\right\},\\ &(3)\quad x_3, x_4, x_5, u_1, u_2 \in \mathbf{N}.\end{aligned} \tag{20.32}$$

Let us take as another example

$$\begin{aligned}&(1)\quad [\text{MIN}]\, f = (3\ 3/6)\, x_3 + 5x_4 + 2x_5 + (1\ 5/6)\, u_1 + 4/6 u_2,\\ &(2)\quad 3/6 = \{3/6 x_3 + 1/6 u_1 + 2/6 u_2\},\\ &(3)\quad x_3, x_4, x_5, u_1, u_2 \in \mathbf{N}.\end{aligned} \tag{20.33}$$

We shall again leave it to the reader to solve this problem, using the algorithm given in the Appendix. We notice that the problem can be simplified

[1] For simplification we shall assume $\delta_m \geqslant 0$. If this were not the case δ_m would be replaced in all the calculations by its absolute value $|\delta_m|$.

by observing that $x_4 = 0$, $x_5 = 0$ for the optimum since x_4 and x_5 have nonnegative coefficients in (1) and do not appear in (2). We obtain the following table:

(20.34)

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(0)	ξ	$\Lambda_1(\xi)$	$x_3^*(\xi)$	$\Lambda_2(\xi)$	$u_1^*(\xi)$	$\Lambda_3(\xi)$	$u_2^*(\xi)$
(1)	0	0	0	0	0	0	0
(2)	1/6	∞		$1\frac{5}{6}$	1	$1\frac{5}{6}$	0
(3)	2/6	∞		$3\frac{4}{6}$	2	4/6	1
(4)	3/6	$3\frac{1}{2}$	1	$3\frac{1}{2}$	0	$2\frac{1}{2}$	1
(5)	4/6	∞		$5\frac{2}{6}$	1	8/6	2
(6)	5/6	∞		$7\frac{1}{6}$	2	$3\frac{1}{6}$	2

.

Column (6) gives the minimal values of the economic function for 6 different second members of the form $\{\lambda/6\}$ of (20.33). In particular for $\{\xi\} = 3/6$ we find $\Lambda_3(3/6) = 2\,1/2$ and, by calculating the optimal solution as we have done in the Appendix, $u_2^*(3/6) = 1$, $u_1^*(3/6) = 1$, $x_3^*(3/6) = 0$. The optimal solution of (20.34) can therefore be given as

$$\begin{aligned} x_3^* &= 0, \quad x_4^* = 0, \quad x_5^* = 0, \\ u_1^* &= 1, \quad u_2^* = 1, \quad f = 2\,1/2. \end{aligned} \tag{20.35}$$

We could also calculate the optimal solutions for the five other possible second members. As an exercise, the reader can check that the following results are obtained:

(20.36)

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(0)	$\lambda/6$	f	x_3	x_4	x_5	u_1	u_2
(1)	0	0	0	0	0	0	0
(2)	1/6	$1\frac{5}{6}$	0	0	0	1	0
(3)	2/6	4/6	0	0	0	0	1
(4)	3/6	$2\frac{1}{6}$	0	0	0	1	1
(5)	4/6	8/6	0	0	0	0	2
(6)	5/6	$3\frac{1}{6}$	0	0	0	1	2

.

We substitute the values given by (20.35) in equations (1), (2), and (3) of (20.27) that now become

$$(20.37)\qquad \begin{aligned} &(1)\ \ x_2 = 23\ 3/6 - 3/6x_3 - 2x_4 - x_5 - 3/6u_1\,,\\ &(2)\ \ x_1 = 45 - 2x_3 - 3x_4 - x_5 - 4/6u_1 - 2/6u_2\,,\\ &(3)\ \ g\ = 113\ 3/6 - (3\ 3/6)\,x_3 - 5x_4 - 2x_5 - (1\ 5/6)u_1\\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad - 4/6u_2\,, \end{aligned}$$

that is, with the values of (20.35),

$$(20.38)\qquad x_2^* = 23\,,\qquad x_1^* = 44\,,\qquad g = 111\,.$$

Since x_2 and x_1 are nonnegative in (20.38), then (20.35) and (20.38) constitute an optimal solution of (20.24).

4. Necessary Condition for Solving the Linear Problem in Integers by the Solution of the Asymptotic Problem

Theorem 20.1

The optimal solution of (20.18) is such that

$$(20.39)^1\qquad \sum_{i=1}^{n} x_{N_i}^{*} \leqslant \delta_m - 1\,,\qquad \text{with}\quad \delta_m = \det\,[B]\,.$$

Proof

We shall use a general theorem from the theory of graphs for this proof. For this purpose we shall choose a problem of the shortest path, the solution of which will allow us to obtain that of the asymptotic problem (20.18).

To begin with, we assume $\mathscr{T}_p/\delta_m$ to be equal to the pth element of the vector $\{1/\delta_m([\mathscr{T}]_m)_{n\times 1}\}$ that appears in constraint (2) of (20.18). Let us construct a graph with δ_m vertices (0), (1), ..., $(\delta_m - 1)$. Each of these vertices will be made to correspond with one of the values $0/\delta_m$, $1/\delta_m$, ..., δ_{m-1}/δ_m of a group that is cyclic for modulo 1 addition. We shall construct an oriented arc between vertex (i) and another vertex (j) if there is a p, $i \leqslant p \leqslant n$ such that

$$(20.40)\qquad \left\{\frac{i}{\delta_m} + \frac{\tau_p}{\delta_m}\right\} = \left\{\frac{j}{\delta_m}\right\} = \frac{j}{\delta_m}\,.$$

Let us apportion a length $\bar{c}_p$ (see line (1) of Eq. (20.18)) to this arc. As an exercise the reader can construct the graph corresponding to the problem (A1.56) the solution of which is given in the Appendix. This graph is shown in Fig. 20.3.

The problem of asymptotic programming leads to the same optimal value for the economic function as that of the shortest path between vertex (0) and

[1] See Volume 2, pages 256–264. See also [K18] pp. 347–377.

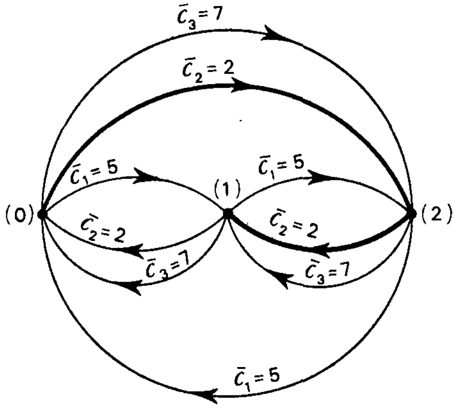


FIG. 20.3

the vertex corresponding to the value $\{\mathscr{U}_m/\delta_m\}$ of the first member of line (2) of Eq. (20.18). For example in (A1.56) it is a question of finding the shortest path between vertex (0) and vertex (1) which is the path starting from (0) and passing through (2) and (1) shown by a heavy line in Fig. 20.3. Its length is

$$(20.41) \qquad f = \bar{c}_2 + \bar{c}_2 = 4 .$$

It corresponds to the solution $x_2^* = 2$, $x_1^* = 0$, $x_3^* = 0$ of program (A1.56).

In a graph where the arcs have nonnegative values, the shortest path between two vertices does not include a loop, that is to say there are never more than $\delta_m - 1$ arcs in a graph containing δ_m vertices. By taking $x_i = 1$ each time that we employ an arc of length $\bar{c}_i$ in the shortest path the theorem is proved.

Corollary 20.II

Let us use $|[x_N^*]|$ for the *length* or *distance in relation to the origin* of the vector constituting the optimal solution of program (20.18). We have

$$(20.42) \qquad |[x_N^*]_{n\times 1}| \leqslant |\det[B]| - 1 .$$

Proof

The length of $[x_N^*]$ is explicitly expressed as

$$(20.43) \qquad |[x_N^*]_{n\times 1}| = [(x_{N_1}^*)^2 + (x_{N_2}^*)^2 + \dots + (x_{N_n}^*)^2]^{1/2} .$$

We have

$$(20.44) \qquad [(x_{N_1}^*)^2 + (x_{N_2}^*)^2 + \dots + (x_{N_n}^*)^2]^{1/2} \leqslant x_{N_1}^* + x_{N_2}^* + \dots + \overset{*}{x}_{N_n} ,$$

since $x_{N_i} \geqslant 0, \quad i = 1, 2, \dots, n.$

Hence this corollary is proved in accordance with (20.43) and (20.44) and Theorem 20.1.

Theorem 20.III[1]

A sufficient condition for the optimal solution of $[x_N^*]$ of the asymptotic problem (20.18) to provide a solution for the problem in integers (20.1) if we calculate $[x_B]$ by line (2) of Eq. (20.5) as a function of $[x_N^*]$ is

$$(20.45)\qquad [B]_{m\times m}^{-1}\cdot[b]_{m\times 1} \geqslant l_{\max}\cdot(|\det[B]|-1)\cdot[1]_{m\times 1},$$

where $l_{\max}$ is the length of the longest of the m vectors forming the lines of the matrix $[B]_{m\times m}^{-1}\cdot[N]_{m\times n}$ and $[1]_{m\times 1}$ is a vector all the elements of which are equal to 1.

Proof

We expand the following expression:

$$(20.46)\qquad ([B]^{-1})_{m\times m}\cdot[N]_{m\times n}\cdot[x_N^*]_{n\times 1} = \begin{bmatrix}([B]^{-1}\cdot[N])_1\cdot[x_N^*]\\ ([B]^{-1}\cdot[N])_2\cdot[x_N^*]\\ \vdots\\ ([B]^{-1}\cdot[N])_m\cdot[x_N^*]\end{bmatrix}.$$

For two vectors $[p]_{1\times n}$ and $[q]_{n\times 1}$ Schwartz's inequality can always be applied:

$$[p]_{1\times n}\cdot[q]_{n\times 1} \leqslant |[p]|\cdot|[q]|,$$

and (20.46) becomes

$$(20.47)\qquad ([B]^{-1})_{m\times m}\cdot[N]_{m\times n}\cdot[x_N^*]_{n\times 1} \leqslant \begin{bmatrix}|([B]^{-1}\cdot[N])_1|\cdot|[x_N^*]|\\ |([B]^{-1}\cdot[N])_2|\cdot|[x_N^*]|\\ \cdots\\ |([B]^{-1}\cdot[N])_m|\cdot|[x_N^*]|\end{bmatrix}_{m\times 1}.$$

By using the definition of $l_{\max}$ and Corollary 20.II, (20.47) becomes

$$(20.48)\qquad ([B^{-1}])_{m\times m}\cdot[N]_{m\times n}\cdot[x_N^*]_{n\times 1} \leqslant \begin{bmatrix}l_{\max}\cdot(|\det[B]|-1)\\ l_{\max}\cdot(|\det[B]|-1)\\ \cdots\\ l_{\max}\cdot(|\det[B]|-1)\end{bmatrix}_{m\times 1}.$$

$$= l_{\max}\cdot(|\det[B]|-1)\cdot[1]_{m\times 1}$$

If we now assume condition (20.45) of the theorem is satisfied, then

$$(20.49)\qquad ([B^{-1}])_{m\times m}\cdot[b]_{m\times 1} \geqslant l_{\max}\cdot(|\det[B]|-1)\cdot[1]_{m\times 1}$$
$$\geqslant ([B]^{-1})_{m\times m}\cdot[N]_{m\times n}\cdot[x_N^*]_{n\times 1},$$

[1] We are indebted to Gomory [K41] for these theorems but, for the purpose of instruction, their proofs are different here.

in accordance with (20.48). We then have

$$[x_B^*]_{m\times 1} = [B]^{-1}.[b] - [B]^{-1}.[N].[x_N^*] \geqslant [0]_{m\times 1},$$

which proves that $|[x_B^*]\ [x_N^*]|$ is a solution of (20.1).

Geometric Interpretation

$[B^{-1}]_{m\times m}.[y]_{m\times 1} \geqslant l_{\max}(|\det[B]|-1).[1]_{m\times 1}$ is the equation of a cone contained in the cone

$$([B^{-1}])_{m\times m}.[y]_{m\times 1} \geqslant [0]_{m\times 1}.$$

Theorem 20.III shows that a sufficient condition for the solution of the asymptotic problem to give a solution of the problem in integers is for $[B]_{m\times 1}$, the second member, to be a vector contained in this cone (see Fig. 20.4). The rays of the cone are the vector columns of the inverse of matrix $[B]$ namely $([B]^{-1})^i$, $([B^1]^{-1})^j$ in Fig. 20.4.

The cone is the set of points situated at a geometric distance[1] of more than $l_{\max}(|\det[B]|-1)$ of the hyperplanes that delimit the cone $[B]^{-1}.[y] \geqslant [0]$.

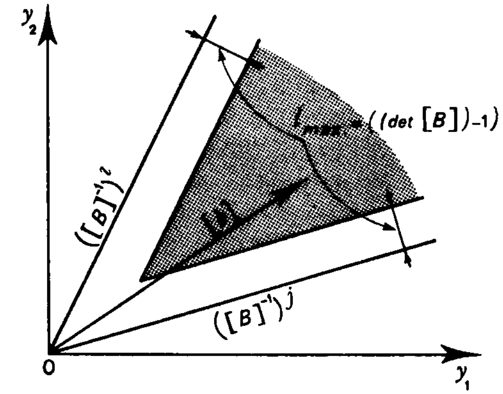


FIG. 20.4

Example

Let us take problem (19.16) and consider table (19.18). Let us use Theorem 20.III to discover whether the asymptotic problem guarantees a solution. We

[1] The geometric distance to a plane $([B]_i^{-1})_{1\times m}.[y]_{m\times 1} = 0$ of a point $[b]$ is $[B]_i^{-1}.[b]$; the metric distance is $[B]_i^{-1}.[b]/|[B]_i^{-1}|$.

have

(20.50)

$$[B]^{-1}_{3\times 3}\cdot[b]_{3\times 1} = \begin{bmatrix} 1\ 6/7 \\ 4\ 3/7 \\ 1\ 2/7 \end{bmatrix}, \qquad [B]^{-1}_{3\times 3}\cdot[N]_{3\times 2} = \begin{bmatrix} 1/7 & 2/7 \\ -3/7 & 3\ 1/7 \\ -2/7 & 3/7 \end{bmatrix},$$

$\det[B] = 7,$

$$l_{\max} = \max(\sqrt{(1/7)^2+(2/7)^2},\quad \sqrt{(-3/7)^2+(3\ 1/7)^2},\quad \sqrt{(-2/7)^2+(3/7)^2})$$

$$= \sqrt{493/7},$$

that is $l_{\max} \simeq 19.3$.

We do not have $43/7 \geqslant (7-1)\ (19.3)$ and we cannot guarantee that the optimum for the asymptotic problem will give a solution of the problem in integer numbers (19.16).

Section 21. **Partition of Linear Programs into Mixed Numbers**

1. Solution of Linear Programs by Partition

We shall now explain a method that enables us to reduce the solution of a problem of very large dimensions to the alternate solution of two smaller associated problems that we refer to as *master* and *slave*. It will also enable us to explain how other methods such as that of Benders [K29], which is given later, can transform the solution of such programs into the alternate solution of an integer and a linear program.

Let us consider the following linear program[1]:

$$(21.1)\quad \begin{aligned} &(1)\quad Z^* = \max_{[x],[w]}(Z = [c]'_{1\times n}\cdot[x]_{n\times 1} + [e]'_{1\times p}\cdot[w]_{p\times 1}),\\ &(2)\quad [a]_{m\times n}\cdot[x]_{n\times 1} + [d]_{m\times p}\cdot[w]_{p\times 1} \leqslant [b]_{m\times 1},\\ &(3)\quad [x]\in \mathbf{R}^n, \qquad [w]\in\mathbf{R}^p,\\ &(4)\quad [x]\geqslant[0], \qquad [w]\geqslant[0]. \end{aligned}$$

Let us take $[w]$ as a parameter with $[\bar{w}]$ a value of it such that $[\bar{w}] \geqslant 0$. Let us assume successively

$$(21.2)\qquad g = [c]'_{1\times n}\cdot[x]_{n\times 1},$$

[1] The notation [MAX] signifies a search for the maximum of the function; that of $\max_{[x],[w]} Z$ indicates the value of that maximum in relation to the n elements of $[x]$ and to the p elements of $[w]$; max indicates the maximum in relation to the n elements of $[x]$ only.

$$(21.3)\qquad Z_1([\bar{w}]) = \max_{[x]} (g = [c]'_{1\times n}\cdot[x]_{n\times 1} \big| [a]_{m\times n}\cdot[x]_{n\times 1} \leqslant [b]_{m\times 1} - [d]_{m\times p}\cdot[\bar{w}]_{p\times 1},\ [x]_{n\times 1} \geqslant [0]),$$

$$(21.4)\qquad \hat{Z}([\bar{w}]) = [e]'_{1\times p}\cdot[\bar{w}]_{p\times 1} + Z_1([\bar{w}])\,.$$

We can then say

$$(21.5)\qquad Z^* = \max_{[\bar{w}]} (\hat{Z}([\bar{w}]) \big| [\bar{w}]_{p\times 1} \geqslant [0]_{p\times 1})\,.$$

This provides an expression of recurring functions such that, by substituting (21.3) in (21.4) and the result in (21.5), we discover the enunciation of program (21.1) where Z^* is the optimum.

Let us observe that (21.4) itself for a given $[\bar{w}]$ is a linear program in $[x]$. Its optimal value $Z_1([\bar{w}])$ is the same as the optimum of its dual program (see 16.121) with precited conditions for this table. We can express the dual program of (21.3) as

$$(21.6)\qquad Z_1([\bar{w}]) = \min_{[y]} (f = ([b]_{m\times 1} - [d]_{m\times p}\cdot[\bar{w}]_{p\times 1})'\cdot[y]_{m\times 1} \big| [a]'_{n\times m}\cdot[y]_{m\times 1} \geqslant [c]_{n\times 1},\ [y]_{m\times 1} \geqslant [0]),$$

where we have assumed

$$(21.7)\qquad f = ([b]_{m\times 1} - [d]_{m\times p}\cdot[\bar{w}]_{p\times 1})'\cdot[y]_{m\times 1}\,,$$

to express the economic function of the dual program.

Let us take **Y** for the set of solutions that satisfy the constraints of the dual program so that

$$(21.8)\qquad \mathbf{Y} = \{[y]_{m\times 1} \big| [a]'_{n\times m}\cdot[y]_{m\times 1} \geqslant [c]_{n\times 1},\ [y]_{m\times 1} \geqslant [0]\}\,.$$

Let us further take $[V_i]$, $i = 1, 2, \dots, T$, for the rays of the convex polyhedron (21.8). Eventually the set of $[V_i]$ may be empty. Finally, let us use $[Y_i]$, i = 1. 2. ..., S, for the extreme points of the convex polyhedron (21.8). Eventually the set of $[Y_i]$ may be empty, but we shall exclude this case since the minimum of (21.3) would then be carried into infinity (see 16.121).

Let us consider linear program (21.3). If, for a vector $[\bar{w}]_{p\times 1}$, this program lacks a solution, then there is a ray $[V_r]_{m\times 1}$, $r \in \{1, 2, \dots, T\}$ of the dual convex polyhedron (21.8). The direction of this ray is obtained by the dual-simplex method, as was shown by Theorem 16.III. We have also shown that the scalar product of the vector slope of the economic function of the dual program and of the vector $[V_r]$ gives a negative number (16.124). For the vector $[\bar{w}]$ with which we are concerned, we have

$$(21.9)\qquad ([b]_{m\times 1} - [d]_{m\times p}\cdot[\bar{w}])'_{p\times 1}\cdot[V_r]_{m\times 1} < 0\,.$$

The same would be true of any vector $[w]_{p\times 1}$ that satisfies (21.9) and where $[w]$ would take the place of $[\bar{w}]$.

If the linear program (21.3), that depends on the value of the components of vector $[w]$, is to have a solution, $[w] \geqslant [0]$ must satisfy the following constraints:

(21.10)

$$([b]_{m\times 1} - [d]_{m\times p}.[w]_{p\times 1})'.[V_i]_{m\times 1} \geqslant 0, \qquad i = 1, 2, \dots, T.$$

For every vector $[w]$ that satisfies these, if the primal linear program has a solution that gives an infinite value to g, we obtain, by solving (21.3) (for example, by the dual-simplex method) a particular extreme point $[Y_i]_{m\times 1}$. In (16.35) we explained how the coordinates of this point could be extracted.

Let us note that the optimum of the dual problem (21.6), if it possesses a solution, is found in one of the extreme points of the dual convex polyhedron (21.1). Hence we do not consider the solutions $[y]_{m\times 1}$ of (21.8) that are not extreme points. Therefore we can say, beginning with 21.6,

(21.11)

$$Z_1([\bar{w}]) = \min_i \left(f = ([b] - [d].[\bar{w}])'_{1\times m}.[Y_i]_{m\times 1} \middle| i = 1, 2, \dots, S\right).$$

If we now return to (21.4) and consider a vector $[w]$ that satisfies (21.10), we can say

$$\text{(21.12)} \qquad \hat{Z}([w]) = [e]'.[w] + Z_1([w]).$$

Let us again consider a vector that satisfies (21.10). By introducing a scalar $Z_2 \in \mathbf{R}$ satisfying

$$\text{(21.13)} \qquad \begin{array}{ll} (1) & Z_2 \leqslant ([b] - [d].[w])'.[Y_1], \\ (2) & Z_2 \leqslant ([b] - [d].[w])'.[Y_2], \\ \vdots & \qquad \vdots \\ (S) & Z_2 \leqslant ([b] - [d].[w])'.[Y_s], \end{array}$$

we can say that

$$\text{(21.14)} \qquad Z_1([w]) = \max_i \left(Z_2 \middle| Z_2 \leqslant ([b] - [d].[w])'.[Y_i],\right.$$
$$\left. i = 1, 2, \dots, S\right).$$

Let there be another scalar $w_0 \in \mathbf{R}$ and let us add $[e]_{1\times p}.[w]_{p\times 1}$ to the right-hand members of the S inequalities. By considering (21.12) we obtain

$$\text{(21.15)} \qquad \begin{array}{ll} (1) & w_0 \leqslant [e]'.[w] + ([b] - [d].[w])'.[Y_1], \\ (2) & w_0 \leqslant [e]'.[w] + ([b] - [d].[w])'.[Y_2], \\ \vdots & \qquad \vdots \\ (S) & w_0 \leqslant [e]'.[w] + ([b] - [d].[w])'.[Y_s]. \end{array}$$

And (21.14) becomes

$$(21.16)\qquad [e'].[w] + Z_1([w]) = \max_i \left(w_0 \middle| w_0 \leqslant [e'].[w] + ([b] - [d].[w]).[Y_i]\right), \quad i = 1, 2, \ldots, S).$$

The left member of (21.16) above is no other than $\check{Z}([w])$ (see 21.12). If we substitute (21.16) in (21.5), it follows that

$$(21.17)\qquad \begin{aligned} &(1)\quad Z^* = \max\ (\hat{Z}([w]) = w_0) \quad \text{subject to} \\ &(2)\quad w_0 \leqslant [e]'_{1\times p}\cdot[w]_{p\times 1} + ([b]_{m\times 1} - [d]_{m\times p}\cdot[w]_{p\times 1})' \\ &\qquad\qquad \cdot[Y_i]_{m\times 1}, \qquad i = 1, 2, \ldots, S, \\ &(3)\quad ([b]_{m\times 1} - [d]_{m\times p}\cdot[w]_{p\times 1})'\cdot[V_i]_{m\times 1} \geqslant 0, \\ &\qquad\qquad i = 1, 2, \ldots, T, \qquad \text{from } (21.10), \\ &(4)\quad [w]_{p\times 1} \geqslant [0]_{p\times 1}. \end{aligned}$$

If we take $[[x^*]\ [w^*]]$ for the optimal solution of program (21.1), then, by following the successive transformations we have made, $[w^*]$ is also the optimal solution of program (21.17) in which vector $[x]$ does not intervene.

If we knew in advance the list of extreme points $[Y_i]$, $i = 1, 2, \ldots, S$, and that of the rays $[V_i]$, $i = 1, 2, \ldots, T$, of (21.8), we could express all the constraints of (21.17) and solve the program for which the optimum is $[w^*]$. It would then be necessary only to solve the linear program (21.3) in $[x]$ alone to obtain the optimal solution $[x^*]$.

This procedure is clearly only theoretical since the number of extreme points and of rays can be very large. Using Benders's method, given below, we progressively calculate the extreme points and rays of which we have need. We shall show that (21.17) can be solved without finding them all.

2. The Case of Programs with Mixed Numbers. Benders's Method

We shall now follow the procedure of Benders [K29] for the very general method of decomposition explained above to solve programs with mixed numbers such, for instance, as

$$(21.18)\qquad \begin{aligned} &(1)\quad Z^* = \max_{[x],[w]}\ (Z = [c]'_{1\times n}\cdot[x]_{n\times 1} + [e]'_{1\times p}\cdot[w]_{p\times 1}), \\ &(2)\quad [a]_{m\times n}\cdot[x]_{n\times 1} + [d]_{m\times p}\cdot[w]_{p\times 1} \leqslant [b]_{m\times 1}, \\ &(3)\quad [x] \in \mathbf{R}^n, \qquad [w] \in \mathbf{N}^p, \\ &(4)\quad [x] \geqslant [0], \qquad [w] \geqslant [0]. \end{aligned}$$

The method of decomposition for linear programs given above applies to the MIP (21.18), since we used the theory of duality (see Section 16) to transform the linear program (21.3) in $[x]$. By referring to (21.17) we obtain the optimal solution of the MIP by solving the following problem in integers[1]:

$$\begin{aligned}
&(1)\quad Z^* = \max_{w_0,\,[w]} \,(Z = w_0)\,, \text{ subject to}\\
&(2)\quad w_0 \leqslant [e]'_{1\times p}\cdot[w]_{p\times 1} + ([b]_{m\times 1} - [d]_{m\times p}\cdot[w]_{p\times 1})'\\
&\qquad\qquad\qquad\qquad\qquad\qquad \cdot[Y_i]_{m\times 1}\,, \qquad i = 1, 2, \dots, S\,,\\
&(3)\quad ([b]_{m\times 1} - [d]_{m\times p}\cdot[w])'_{p\times 1}\cdot[V_i]_{m\times 1} \geqslant 0\,,\quad i = 1, 2, \dots, T,\\
&(4)\quad [w] \in \mathbf{N}^p\,,
\end{aligned}\tag{21.19}$$

the optimal solution of which is $[w^*]_{p\times 1}$, Z^*. For this vector $[w^*]$ we solve the linear program (21.3) in $[x]_{n\times 1}$ only and obtain the optimal solution $[x^*]$. As we have shown $[[x^*]_{n\times 1}[w^*]_{p\times 1}]$, constitutes the optimal solution of (21.18). Let $S_1 < S$ and $T_1 \leqslant T$, and let us now consider the following integer problem that we shall call the *master problem.*

$$\begin{aligned}
&(1)\quad \check{Z}^{(1)} = \max_{w_0,\,[w]} \,(\check{Z} = w_0)\,,\\
&(2)\quad w_0 \leqslant [e]'.[w] + ([b] - [d].[w])'.[Y_i]\,,\quad i = 1, 2, \dots, S_1\,,\\
&(3)\quad ([b] - [d].[w])'.\,[V_i] \geqslant {}_0\,,\qquad i = 1, 2, \dots, T_1\,,\\
&(4)\quad [w]_{p\times 1} \in \mathbf{N}^p.
\end{aligned}\tag{21.20}$$

If problem (21.19) has a solution[2] $[w] = [w^{(1)}]$, $\check{Z} = \check{Z}^{(1)}$, then (21.20) also has a solution since its constraints form a subset of those of (21.19), and we have

$$\check{Z}^{(1)} \geqslant Z^*.\tag{21.21}$$

Let $[w^{(1)}]_{p\times 1}$, $\check{Z}^{(1)}$ be the optimal solution of (21.20). For this $[w^{(1)}]$ we solve the linear program (21.3) that we shall call the *slave problem* to underline that it is solved after (21.20). Program (21.3) becomes

$$\begin{aligned}
Z_1[w^{(1)}] = \max_{[x]} \,(g = [c]'_{1\times n}\cdot[x]_{n\times 1} \,|\, &[a]_{m\times n}\cdot[x]_{n\times 1}\\
&\leqslant [b]_{m\times 1} - [d]_{m\times p}\cdot[w^{(1)}]_{p\times 1}\,, [x] \geqslant [0]_{n\times 1}).
\end{aligned}\tag{21.22}$$

[1] We assume that $[e]$, $[b]$, $[d]$ are matrices with integer elements and that we multiply constraints (2) and (3) of (21.19) by an integer such that the coefficients of these constraints will be integer. This means that w_0 will also be integer.

[2] We use the notation $\check{Z}$ to indicate that it is an upper bound of Z^*. In the same way $\hat{Z}([\bar{w}])$ denotes a lower bound of Z^* since, if $[\bar{w}]$ is not an optimal solution of (21.19), we have $Z^* > \hat{Z}([\bar{w}])$.

Two cases may appear when we solve the *slave* program (21.11). We shall discuss them in succession.

First Case

The slave program (21.22) has no solution for $[w^{(1)}]$. We then obtain a ray V_{T_2}, with $T_2 = T_1+1$ that is such (see 21.9) that we have

(21.23) $$([b]_{m\times 1} - [d]_{m\times p}.[w^{(1)}]_{p\times 1})'.[V_{T_2}]_{m\times 1} < 0.$$

The point $[w^{(1)}]$ does not therefore satisfy one of the constraints of (21.9) that we omitted when solving (21.20). Accordingly we add to (21.20) the constraint

(21.24) $$([b]_{m\times 1} - [d]_{m\times p}.[w])'_{p\times 1}\;.[V_{T_2}]_{m\times 1} \geqslant 0.$$

With this added, the point $[w^{(1)}]$ can no longer be obtained as the solution of (21.20) since we should have both (21.23) and

(21.25) $$([b] - [d].[w^{(1)}])'.[V_{T_2}] \geqslant 0,$$

which is a contradiction.

In the same way $[V_{T_2}]$ can no longer be obtained as a ray of (21.8) by solving (21.3) since we should have as a vector $[w^{(k)}]$, which satisfies (21.24),

(21.26) $$([b] - [d].[w^{(k)}])'.[V_{T_2}] < 0,$$

since $[V_{T_2}]$ is a ray of (21.8) obtained by solving (21.3) for $[\bar{w}] = [w^{(k)}]$, which is a contradiction.

Second Case

The slave program (21.22) has an optimal solution $[x^{(1)}]$ for $[w^{(1)}]$. We then obtain (see (16.35)) an extreme point $[Y_{S_2}]$ of (21.8). Let $Z_1([w^{(1)}])$ be the optimal value of the economic function of (21.22). We calculate $\hat{Z}([w^{(1)}])$ by (21.12) and distinguish two cases (a) and (b):

(21.27) (a) $\check{Z}^{(1)} = \hat{Z}([w^{(1)}])$.

In this case an upper bound of Z^* (in accordance with (21.21)) is equal to a lower bound of Z^* (from (21.5)). Hence we have found the optimum of the MIP (21.18),

(21.28) $$[x^*]_{n\times 1} = [x^{(1)}]_{n\times 1}, \qquad [w^*]_{p\times 1} = [w^{(1)}]_{p\times 1},$$
$$Z^* = \check{Z}^{(1)} = \hat{Z}([w^{(1)}]).$$

(21.29) (b) $\check{Z}^{(1)} > \hat{Z}([w^{(1)}])$.

From (21.11) and (21.12), (21.29) can be expressed

(21.30)

$$\check{Z}^{(1)} > [e]'_{1\times p}\cdot[w^{(1)}]_{p\times 1} + ([b]_{m\times 1} - [d]_{m\times p}\cdot[w^{(1)}]_{p\times 1})'\cdot[Y_{S_2}]_{m\times 1}\,.$$

Hence there is a constraint (2) of (21.19) corresponding to $i = S_2$ omitted from the constraints of (21.20) that is not satisfied by $[w^{(1)}]$ since, if it were satisfied, we should have

(21.31)

$$\check{Z}^{(1)} = \max\,(w_0 | w_0 \leqslant [e]'.[w^{(1)}] + ([b] - [d].[w^{(1)}])'.[Y_{S_2}]),$$

which contradicts[1] (21.30). We therefore add to (21.20) the constraint

$$w_0 \leqslant [e]'.[w] + ([b].[d].[w])'.[Y_{S_2}]. \tag{21.32}$$

The point $[w^{(1)}]$ can no longer be obtained as the solution of program (21.20) to which we have added (21.32), since we should have both (21.30) and (21.31).

Similarly, constraint (2) of (21.19) corresponding to $i = S_2$ cannot be added a second time to the constraints of the program when we obtain the extreme point $[Y_{S_2}]$ again for another solution $[w^{(k)}]$, since we should have

$$w_0 \leqslant [e]'.[w^{(k)}] + ([b] - [d].[w])'.[Y_{S_2}] \tag{21.33}$$

from (21.31), as well as

$$\check{Z} = \check{Z}^{(k)} = \max\, w_0 \tag{21.34}$$

greater than $\hat{Z}([w^k])$ since we assume that we do not have the optimum, with

$$\hat{Z}([w^{(k)}]) = [e]':[w^{(k)}] + ([b] - [d].[w^{(k)}])'.[Y_{S_2}], \tag{21.35}$$

and since we also assume that we again obtain this extreme point when we solve (21.3) for $[\overline{w}] = [w^{(k)}]$. Relations (21.33)–(21.35) are contradictory.

We have therefore proved the following theorem:

Theorem 21.I

All the constraints (2) and (3) of (21.20) obtained with Benders's method are distinct.

This theorem enables us to prove another theorem.

Theorem 21.II

The optimal values $\check{Z}^{(1)}, \check{Z}^{(2)}, \ldots, \check{Z}^{(k)}$ of the economic functions of the various master programs (21.20) for Benders's method are monotone nonincreasing.

Proof

Every time that we solve a slave program (21.3), if we do not obtain the optimum, we then add a constraint to (21.20). In accordance with Theorem 21.I these constraints are distinct. The different problems (21.20) allow of more and more restrictive constraints and their maximal form a nonincreasing sequence that can be expressed, if (21.18) has an optimal solution with Z^*

[1] Need we recall that $a = \max(b\,|\,b \leqslant c)$ means that $a \leqslant c$?

the maximum of the economic function, as

$$(21.36) \qquad \check{Z}^{(1)} \geqslant \check{Z}^{(2)} \geqslant \dots \geqslant \check{Z}^{(k)} \geqslant \dots \geqslant Z^* .$$

3. A Numerical Example of Benders's Method

In this example we shall use a graphical method to solve the *master problem* of Benders's procedure, since $[w]_{p \times 1} \in \mathbf{R}^2$. In the next section a more sophisticated algorithm will be employed for its solution (see [K15] for fuller details).

Given the program

$$(21.37) \qquad \begin{array}{ll} (1) & Z^* = \max\limits_{x_1, x_2, w_1, w_2} \quad (Z = -x_1 - 3x_2 - w_1 - 4w_2), \\ (2) & 2x_1 + x_2 - w_1 + 2w_2 \leqslant -1, \\ (3) & -2x_1 - 2x_2 + w_1 - 3w_2 \leqslant -1, \\ (4) & x_1, x_2 \in \mathbf{R}^+, \qquad w_1, w_2 \in \mathbf{N}. \end{array}$$

The MIP above has the form of (21.1); the reader can easily verify that the constraints (21.8) are expressed as

$$(21.38) \qquad \begin{array}{ll} (1) & 2y_1 - 2y_2 \geqslant -1, \\ (2) & y_1 - 2y_2 \geqslant -3, \\ (3) & y_1, y_2 \geqslant 0. \end{array}$$

The convex polyhedron delimited by these constraints is shown in Fig. 21.1. In it two rays $[V_1]$ and $[V_2]$ of the convex polyhedron $\mathbf{Y}$ appear, which means that w_1 and w_2 may possess values for which program (21.37) has no solution.

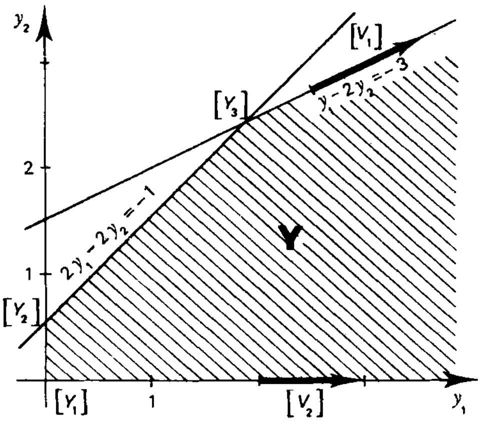


FIG. 21.1

In Benders's method we first solve a master problem for which the constraints are limited to $[w]_{2\times 1} \geqslant [0]_{2\times 1}$; this means that program (21.20) is reduced to

$$(21.39)\qquad \begin{aligned} &(1)\ \ [\mathrm{MAX}]\ \check{Z} = w_0\,,\\ &(2)\ \ w_1\,,\ w_2 \in \mathbf{N}\,. \end{aligned}$$

There is an infinitude of optimal solutions for (21.32), one, for example, being

$$(21.40)\qquad w_1 = 4\,,\qquad w_2 = 10\,,\qquad \check{Z} = w_0$$

equal to a value that approaches the infinite. This infinite value for the economic function is awkward. It is obtained, since there is no bound for w_0 in (21.39), because we have not yet found any constraint such as (21.32). Let us observe that, if in the polyhedron of the constraints of the dual program (21.8) we have $[c]_{n\times 1} \geqslant [0]_{n\times 1}$, then $[Y_1]_{m\times 1} = [0]_{m\times 1}$ is an extreme point of (21.8) We can always transform the MIP (21.1) in such a way as to have

$$(21.41)\qquad c_i \geqslant 0\,,\qquad i = 1, 2, \ldots, n\,.$$

It is sufficient, if $c_i < 0$, to assume $x_i^1 = -x_i$ (the reader will recall our procedure for the method of direct search).

Therefore, after transforming (21.1) so as to have $c_i \geqslant 0$, $i = 1, 2, \ldots, n$, we consider the following initial master program:

$$(21.42)\qquad \begin{aligned} &(1)\ \ \check{Z}^{(1)} = \max_{w_2,\,[w]} (\check{Z} = w_0)\,,\\ &(2)\ \ w_0 \leqslant [e]'_{1\times p}.[w]_{p\times 1} + ([b]_{m\times 1} - [d]_{m\times p}.[w]_{p\times 1})'.[0]_{m\times 1}\,,\\ &(3)\ \ [w] \in \mathbf{N}^p\,. \end{aligned}$$

In the example being treated this program becomes

$$(21.43)\qquad \begin{aligned} &(1)\ \ \check{Z}^{(1)} = \max_{w_0,\,w_1,\,w_2} (\check{Z} = w_0)\,,\\ &(2)\ \ w_0 \leqslant -w_1 - 4w_2\,,\\ &(3)\ \ w_1\,,\ w_2 \in \mathbf{N}\,. \end{aligned}$$

This program is illustrated in Fig. 21.2a, and has an optimal solution:

$$(21.44)\qquad [w_1\quad w_2] = [w_1^{(1)}\quad w_2^{(1)}] = [0\quad 0]\,,\qquad \check{Z} = \check{Z}^{(1)} = 0\,.$$

By incorporating this in the slave program (21.22) and after adding the deviation variables u_1 and u_2, the latter becomes

$$(21.45)\qquad \begin{aligned} &(1)\ \ Z_1([0\ 0]) = \max_{x_1,\,x_2,\,u_1,\,u_2} (g = -x_1 - 3x_2)\,,\\ &(2)\ \ 2x_1 + x_2 + u_1 = -1\,, \end{aligned}$$

$$(3)\quad -2x_1-2x_2+u_2 = -1\,,$$

$$(4)\quad x_1\,,\,x_2\,,\,u_1\,,\,u_2 \geqslant 0\,.$$

Let us now solve the above linear program. The initial table of the type of (16.8) follows in (21.46) and does not provide a solution, since u_2 is negative.

(21.46)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)
(0)			g	x_1	x_2	u_1	u_2	φ_1	φ_2
(1)	g	0	1	1	3	0	0	0	0
(2)	u_1	-1	0	2	1	1	0	1	0
(3)	u_2	-1	0	$\boxed{-2}$	-2	0	1	0	1

.

Table (21.46) has nonnegative elements in the first line and we shall use the dual-simplex method for the iterations, taking line (3) for the pivot. As an exercise, the reader may choose suitable pivots for the dual-simplex method and verify that the following table (21.47) is obtained in two dual operations.

(21.47)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)
			g	x_1	x_2	u_1	u_2	φ_1	φ_2
(0)	g	-9/2	1	0	0	2	5/2	2	5/2
(1)	x_2	2	0	0	1	-1	-1	-1	-1
(2)	x_1	-3/2	0	1	0	1	1/2	1	1/2

.

This table does not represent a solution since the value of x_1 is negative in the column corresponding to the second member of (21.45), but we cannot find a negative element among those for line (3) in columns (1)–(7). Using Theorem 16.III, we obtain the direction $[V_1]$ of an extreme ray of the convex polyhedron (21.35) in the columns of the variables φ_1 and φ_2 in line (1), namely,

$$(21.48)\qquad [V_1]' = [1 \quad 1/2]\,.$$

We can verify from Fig. 21.1 that $[V_1]$ is the direction of an extreme ray of (21.38).

Accordingly, program (21.45) has no solution for $[w^{(1)}]$. The constraint of the type of (21.24) is expressed

$$(21.49)\qquad \left(\underset{[b]}{\begin{bmatrix}-1\\-1\end{bmatrix}}-\underset{[d]}{\begin{bmatrix}-1 & 2\\ 1 & -3\end{bmatrix}}\cdot\underset{[w]}{\begin{bmatrix}w_1\\w_2\end{bmatrix}}\right)'\cdot\underset{[V_1]}{\begin{bmatrix}1\\1/2\end{bmatrix}}\geqslant 0\,;$$

that is, after expansion,

$$(21.50)\qquad w_1 - w_2 \geqslant 3.$$

With this constraint added to the master program (21.43), the latter becomes (see (21.20))

$$(21.51)\qquad \begin{aligned} &(1)\ \check{Z}^{(2)} = \max_{w_0,\, w_1,\, w_2} \quad (\check{Z} = w_0),\\ &(2)\ w_0 \leqslant -w_1 - 4w_2,\\ &(3)\ w_1 - w_2 \geqslant 3,\\ &(4)\ w_1, w_2 \in \mathbf{N}. \end{aligned}$$

This program in integers is shown in Fig. 21.2b.

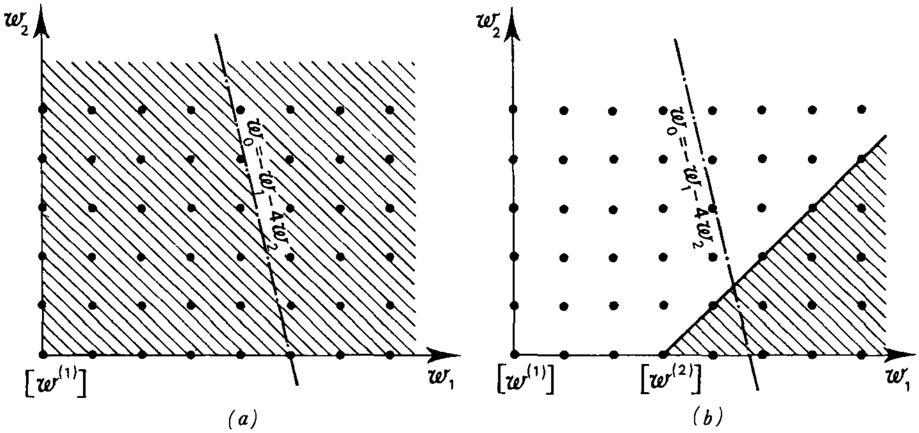


Fig. 21.2

In Fig. 21.2b the optimal solution is

$$(21.52)\qquad [w_1 \quad w_2] = [w_1^{(2)} \quad w_2^{(2)}] = [3 \quad 0], \qquad \check{Z} = \check{Z}^{(2)} = -3.$$

After substituting this in the slave program (21.22) the latter becomes

$$(21.53)\qquad \begin{aligned} &(1)\ Z_1([3\ 0]) = \max_{x_1,\, x_2,\, u_1,\, u_2} \quad (g = -x_1 - 3x_2),\\ &(2)\ 2x_1 + x_2 + u_1 = 2,\\ &(3)\ -2x_1 - 2x_2 + u_2 = -4,\\ &(4)\ x_1, x_2, u_1, u_2 \geqslant 0. \end{aligned}$$

The reader can use the same method as we did to obtain table (21.47) and will find an optimal solution given by the following table:

(21.54)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)
(0)			g	x_1	x_2	u_1	u_2	φ_1	φ_2
(1)	g	-6	1	0	0	2	5/2	2	5/2
(2)	x_2	2	0	0	1	-1	-1	-1	-1
(3)	x_1	0	0	1	0	1	1/2	1	1/2

.

The optimal solution is

(21.55) $[x_1 \quad x_2] = [0 \quad 2], \qquad Z_1([3 \quad 0] = -6 = g.$

By again using (21.12) we can calculate $\hat{Z}([w^{(2)}])$

(21.56) $$\hat{Z}([3 \quad 0]) = \underset{[e]'}{[-1 \quad -4]} . \underset{[w^{(2)}]}{\begin{bmatrix} 3 \\ 0 \end{bmatrix}} + Z_1([3 \quad 0],$$

or again

(21.57) $$\hat{Z}([w^{(2)}]) = \hat{Z}([3 \quad 0]) = -9.$$

Returning to (21.29) we have $\check{Z}^{(2)} > \hat{Z}([w^{(2)}])$ and have not found the optimum for (21.47). From table (21.54) we obtain the extreme point $[Y_3] = [2 \; 5/2]$ of the convex polyhedron (21.38) in the columns of the artificial variables φ_1 and φ_2 in line (1) of (21.54) (refer to Section 16). We then add to the master program (21.51) a constraint of the same type as (21.32) in accordance with our theoretical explanation of Benders's method. We obtain

(21.58)
$$w_0 \leqslant \underset{[e]'}{[-1 \quad -4]'} . \underset{[w]}{\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}} + \left(\underset{[b]}{\begin{bmatrix} -1 \\ -1 \end{bmatrix}} - \underset{[d]}{\begin{bmatrix} -1 & 2 \\ 1 & -3 \end{bmatrix}} . \underset{[w]}{\begin{bmatrix} w_1 \\ w_2 \end{bmatrix}} \right)' . \underset{[Y_3]}{\begin{bmatrix} 2 \\ 5/2 \end{bmatrix}},$$

that, by expansion, becomes

(21.59) $$w_0 \leqslant -9/2 - 3/2 w_1 - 1/2 w_2 .$$

We add (21.59) to (21.51) after multiplying the two members by 2 in order that w_0 shall be integer in the optimum for (21.60):

(21.60) (1) $$\check{Z}^{(3)} = \max_{w_0, w_1, w_2} (\check{Z} = w_0),$$

$$\begin{aligned}
&(2)\quad w_0 \leqslant -w_1 - 4w_2\,,\\
&(3)\quad w_1 - w_2 \geqslant 3\,,\\
&(4)\quad 2w_0 \leqslant -9 - 3w_1 - w_2\,,\\
&(5)\quad w_1\,,\, w_2 \in \mathbf{N}\,.
\end{aligned}$$

By using, for instance, the algorithm for all-integer programming the reader will find that the optimal solution for the master program (21.60) is

$$\text{(21.61)}\qquad [w_1 \quad w_2] = [w_1^{(3)} \quad w_2^{(3)}] = [3 \quad 0]\,, \qquad \check{Z} = \check{Z}^{(3)} = -9\,.$$

Let us observe that $[w^{(3)}] = [w^{(2)}]$, which means that by solving the slave problem (21.22) for $[w] = [w^{(3)}]$ we obtain the same problem as (21.53) and that (see what was done to obtain (21.57)) we also have

$$\text{(21.62)}\qquad \hat{Z}([w^{(3)}]) = \hat{Z}([3 \quad 0]) = -9\,.$$

We shall then, in accordance with (21.27), have shown that the solution

$$x_1^* = 0\,,\ x_2^* = 2\,,\ w_1^* = 3\,,\ w_2^* = 0\,,\ Z^* = \check{Z}^{(3)} = \hat{Z}([w^{(3)}]) = -9\,,$$

is an optimal solution of the MIP (21.37).

Section 22. **Mixed Programming on a Cone**

1. Proof of the Finite Character of the Algorithm Used for the Selection of Masks[1]

We shall express this problem, given in Section 15, in a matrical form. It can be expressed as

$$\text{(22.1)}\qquad \begin{aligned}
&(0)\quad Z^* = \max_{[x],[w]} \left(Z = [0]'_{1\times n}\cdot[x]_{n\times 1} + [e]'_{1\times p}\cdot[w]_{p\times 1}\right),\\
&(1)\quad [a]_{m\times n}\cdot[x]_{n\times 1} + [d]_{m\times p}\cdot[w]_{p\times 1} \leqslant [b]_{m\times 1}\,,\\
&(2)\quad x_i = 0 \quad \text{or} \quad 1\,, \qquad i = 1, 2, \ldots, n\,,\\
&(3)\quad w_i = 0 \quad \text{or} \quad 1\,, \qquad i = 1, 2, \ldots, p\,.
\end{aligned}$$

Here $[0]_{n\times 1}$ is a vector of which all the components are null; in other words, x_i, $i = 1, 2, \ldots, n$, does not appear in the economic function (see (5.56)) and we have

$$\text{(22.2)}\qquad [e]'_{1\times p} = [1 \quad 1 \quad \ldots \quad 1]'_{1\times p}\,.$$

Program (22.1) is one with bivalent variables that can be solved by the algorithm for direct search given in Section 4, but it is preferable to use a

[1] See [K15].

special algorithm that is a modification of Benders's method. We shall first of all define a program that differs slightly from (22.1), namely,

$$(22.3)\qquad \begin{aligned} &(0)\quad Z^* = \max_{[x],[w]} (Z = [0]'_{1\times n}\cdot[x]_{n\times 1} + [e]'_{1\times p}\cdot[w])_{p\times 1}, \\ &(1)\quad [a]_{m\times n}\cdot[x]_{n\times 1} + [d]_{m\times p}\cdot[w]_{p\times 1} \leqslant [b]_{m\times 1}, \\ &(2)\quad [x]_{n\times 1} \geqslant [0]_{n\times 1}, \\ &(3)\quad w_i = 0 \quad\text{or}\quad 1, \qquad i = 1, 2, \ldots, p. \end{aligned}$$

This program in mixed numbers permits the same set of solutions as those of (22.1) but is less constrained owing to (2) being made less restrictive. As a start, let us prove a theorem that will confirm the finite character of the algorithm to be explained.

Theorem 22.1

The first solution $[[x^{(1)}]_{n\times 1}[w^{(1)}]_{p\times 1}]$ of program (22.3) obtained by Benders's method is optimal.

Proof

Using Benders's method, the polyhedron of the constraints of the dual program (see (21.8)) is expressed

$$(22.4)\qquad [a]'_{n\times m}\cdot[y]_{m\times 1} \geqslant [c]_{n\times 1}, \qquad [y]_{m\times 1} \geqslant [0]_{m\times 1}.$$

Here, since $[c] = [0]$, this is,

$$(22.5)\qquad \begin{aligned} &(1)\quad [a]'_{n\times m}\cdot[y]_{m\times 1} \geqslant [0]_{n\times 1}, \\ &(2)\quad [y]_{m\times 1} \geqslant [0]_{m\times 1}. \end{aligned}$$

The convex polyhedron defined by the $n+m$ hyperplanes (1) and (2) of (22.5) only permits one vertex, the point $[Y_1]_{m\times 1} = [0]_{m\times 1}$.

Indeed, whichever m hyperplanes are chosen, their intersection will give $[Y_1] = [0]$. Differently stated, the polyhedron of the constraints of the dual program is a convex polyhedral cone[1] (by widening the definition given in Section 14) and by assuming that there may be more than m hyperplanes passing through the vertex $[Y_1]$ of $\mathbf{R}^m$.

This vertex is the sole extreme point of (22.5). Each extreme point of (22.4) obtained by solving program (21.3) with Benders's method represents a solution of (22.3). Since there is only one extreme point the sole solution $[[x]\ [w]]$ is optimal.

This theorem applies to other problems of MIP differing from that of the selection of masks but having the same structure of constraints.

Let us now give the detailed form of the program (22.1) that enables us to

[1] This explains the name *mixed programming on a cone*.

solve the present problem by referring to Section 5. We have

$$(22.6)\qquad [a]_{m\times n} = \begin{bmatrix} [h]_{(m-p)\times n} \\ [g]_{p\times n} \end{bmatrix}_{m\times n},$$

$$(22.7)\qquad [d]_{m\times p} = \begin{bmatrix} [0]_{(m-p)\times p} \\ [-M]_{p\times p} \end{bmatrix}.$$

And also

$$(22.8)\qquad [b]_{m\times 1} = \begin{bmatrix} [-1]_{(m-p)\times 1} \\ [0]_{p\times 1} \end{bmatrix}_{m\times 1}.$$

The economic function to be minimized is

$$(22.9)\qquad [\mathrm{MIN}]\ \tilde{Z} = [1]'_{1\times p}\cdot[w]_{p\times 1}.$$

By assuming $Z = -\tilde{Z}$ we obtain a problem of maximization allowing the same optimal solution, for which the economic function is expressed as

$$(22.10)\qquad [\mathrm{MAX}]\ Z = -[1]'_{1\times p}\cdot[w]_{p\times 1}.$$

In (22.7), $[-M]_{p\times p}$ is a matrix in which all the elements are null except for those of the diagonal that are equal to $(-M)$; $[-1]_{(m-p)\times 1}$ is a vector in which all the elements are (-1); $[h]_{(m\times p)\times 1}$ is a matrix in which all the nonnull elements equal (-1); $[g]_{p\times n}$ is a matrix in which all the nonnull elements equal 1. Here p is equal to the number of cells, that is, nine for the example in Section 5; $(m-p)$ represents the number of different types of masks (three in the example), and, finally, M must represent a very large positive number greater than the number of nonnull variables x_i, $i = 1, 2, \ldots, n$, in the optimal solution.

Let us now prove the following lemma that is required to prove the finite character of the algorithm to be used.

Lemma 22.II

We consider the case of an MIP in which the matrices $[a]$, $[d]$, $[b]$ have the forms of Eqs. (22.6)–(22.8).

For any solution $[[x]_{n\times 1}[w]_{p\times 1}]_{(n+p)\times 1}$ of (22.3) there is a corresponding solution $[[\tilde{x}]_{n\times 1}[w]_{p\times 1}]_{(n+p)\times 1}$ of program (22.1) with the same vector $[w]$ obtained by rounding off the elements of $[x]$ by transformations (1), (2), and (3) of (22.11).

Proof

Let us take modified values of x_i, $i = 1, 2, \ldots, n$. Then

$$(22.11)\qquad (1)\ \ \tilde{x}_i = x_i \ \text{ if } \quad x_i = 1 \text{ or } 0, \qquad i = 1, 2, \ldots, n,$$

(2) $\tilde{x}_i = 1$ if $0 < x_i < 1$, $i = 1, 2, \ldots, n$,

(3) $\tilde{x}_i = 1$ if $x_i > 1$, $i = 1, 2, \ldots, n$.

Let us give an explicit expression to the constraints of (22.1):

$$([h]_i)_{1\times n}\cdot[x]_{n\times 1} \leqslant -1, \qquad i = 1, 2, \ldots, (m-p), \tag{22.12}$$

$$([g]_i)_{1\times n}\cdot[x]_{n\times 1} \leqslant Mw_i, \qquad i = 1, 2, \ldots, p. \tag{22.13}$$

Here $([h]_i)$ is a vector in which the nonnull elements equal -1 and $([g]_i)$ has its nonnull elements equal to 1. The form of constraints (22.12) and (22.13) means that a vector $[\tilde{x}]_{n\times 1}$ obtained from $[x]_{n\times 1}$, which itself satisfies these constraints through transformations (22.9)–(22.11) also satisfies the same constraints. Since $[\tilde{x}]_{n\times 1}$ is a vector with components of 0 or 1, the vector $[[\tilde{x}]_{n\times 1}[w]_{p\times 1}]_{(n+p)\times 1}$ is a solution of the program for the selection of masks.

Algorithm for the Selection of Masks

We solve the MIP (22.3) by Benders's method, stopping as soon as a solution $[[x^{(k)}]_{n\times 1}[w^{(k)}]_{p\times 1}]$ is obtained. The optimal solution of (22.1) is obtained by applying the first three transformations of (22.11).

The finite character of this algorithm is ensured by (1) the finite character of Benders's method, (2) Theorem 22.I, and (3) Lemma 22.II.

Before employing this algorithm in an example we shall, however, first explain how to solve the submaster program in integer numbers using Benders's method.

2. Solution of the Subprogram in Integers

The algorithm from which is derived the algorithm for solving the master program in integers (see (21.20)) with Benders's method is the dual-simplex one of Lemke as well as Gomory.

At some stage in the iterations we solve the linear slave program (21.3) for $[w] = [w^{(k)}]$ and if the solution is not optimal we add a *Benders's constraint* (21.24) or (21.32) to the integer program, a constraint that is not satisfied by the last solution $[w^{(k)}]_{p\times 1}$ that was obtained.

In the simplex table used for solving the integer program (21.20) one or more dual-simplex iterations are performed until a point $[w]$ has been obtained that satisfies Benders's constraints. If all the coordinates of this point are not integer, we add a Gomory constraint such as (19.15) and perform dual-simplex iterations, finally adding further Gomory constraints until we have obtained a point with integer coordinates. Except that we introduce some of Benders's constraints as well as those of Gomory, this algorithm is identical with that described in Section 19.2.

We shall now illustrate the use of this algorithm by a very simple example.

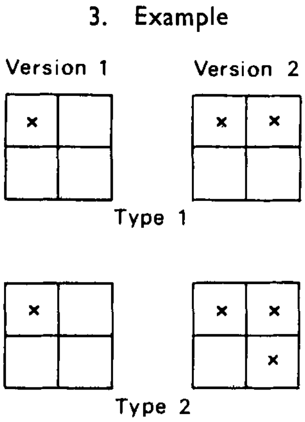


FIG. 22.1

It is obvious that an optimal selection of two different types of mask will include two masks, and we can therefore take $M = 2$ in equation (22.13) where we previously gave it a very large value with the object of proving Lemma 22.II. Program (22.3) can thus be expressed as

$$
(22.14) \qquad
\begin{array}{ll}
(1) & [\mathrm{MIN}]\ \tilde{Z} = w_1 + w_2 + w_3 + w_4 , \\
(2) & x_{11} + x_{12} \geqslant 1 , \\
(3) & x_{21} + x_{22} \geqslant 1 , \\
(4) & x_{22} + x_{11} + x_{12} + x_{21} \leqslant 2w_1 , \\
(5) & x_{12} + x_{22} \leqslant 2w_2 , \\
(6) & x_{22} \leqslant 2w_3 , \\
(7) & w_1 ,\, w_2 ,\, w_3 ,\, w_4 = 0 \quad \text{or} \quad 1 . \\
(8) & x_{11} ,\, x_{12} ,\, x_{21} ,\, x_{22} \geqslant 0 .
\end{array}
$$

Lemma 22.II shows that from a solution $x_{11}, x_{12}, x_{21}, x_{22}$ of this program a solution $\tilde{x}_{11}, \tilde{x}_{12}, \tilde{x}_{21}, \tilde{x}_{22}$ can easily be obtained (see transformations (22.9)–(22.11)).

Let us use Benders's method to solve the above MIP, taking the algorithm explained in the last section to solve the master program in integers such as (21.19).

To begin with, the set of constraints for point $[w]$ is empty and the first master program in pure integer numbers that we solve (see what we showed in (21.42)) is the following:

(22.15)

$$(1)\quad \check{Z}^{(1)} = \max\,(\check{Z} = w_0),$$

$$(2)\quad w_0 \leqslant -w_1 - w_2 - w_3 - w_4,$$

$$(3)\quad w_1, w_2, w_3, w_4 = 0 \quad \text{or} \quad 1,$$

the solution of which is obviously

$$[w^{(1)}]_{1\times 4} = [w_1^{(1)}\ w_2^{(1)}\ w_3^{(1)}\ w_4^{(1)}] = [0\ 0\ 0\ 0] \quad \text{and} \quad w_0^{(1)} = 0.$$

Using Benders's method we now solve the following program, obtained from (22.14), making $[w] = [w^{(1)}]$:

(22.15a)

$$(1)\quad Z_1([w^{(1)}]) = \max_{x_{11},\, x_{12},\, x_{21},\, x_{22}} (g = 0.x_{11} + 0.x_{12} + 0.x_{21} + 0.x_{22}),$$

$$(2)\quad x_{11} + x_{12} \geqslant 1,$$

$$(3)\quad x_{21} + x_{22} \geqslant 1,$$

$$(4)\quad x_{11} + x_{12} + x_{21} + x_{22} \leqslant 0,$$

$$(5)\quad x_{12} + x_{22} \leqslant 0,$$

$$(6)\quad x_{22} \leqslant 0,$$

$$(7)\quad x_{11}, x_{12}, x_{21}, x_{22} \leqslant 0.$$

Let us add deviation variables $u_1, u_2, u_3, u_4, u_5 \geqslant 0$ after changing the direction of inequalities (2) and (3) above. It follows that

(22.16)

$$(1)\quad Z_1([w^{(1)}]) = \max_{x_{11},\, x_{12},\, x_{21},\, x_{22}} (g = 0.x_{11} + 0.x_{12} + 0.x_{21} + 0.x_{22}),$$

$$(2)\quad -x_{11} - x_{12} + u_1 = -1,$$

$$(3)\quad -x_{21} - x_{22} + u_2 = -1,$$

$$(4)\quad x_{11} + x_{12} + x_{21} + x_{22} + u_3 = 0,$$

$$(5)\quad x_{12} + x_{22} + u_4 = 0,$$

$$(6)\quad x_{22} + u_5 = 0,$$

$$(7)\quad x_{11}, x_{12}, x_{21}, x_{22}, u_1, u_2, u_3, u_4, u_5 \geqslant 0.$$

As in (16.92), we construct the simplex table corresponding to program

(22.16), observing that the columns of the five deviation variables form a basis. This table is as follows:

(22.17)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
			g	u_1	u_2	u_3	u_4	u_5	x_{11}	x_{12}	x_{21}	x_{22}	φ_1	φ_2	φ_3	φ_4	φ_5
(0)	g	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(1)	u_1	-1	0	1	0	0	0	0	$\textcircled{-1}$	-1	0	0	1	0	0	0	0
(2)	u_2	-1	0	0	1	0	0	0	0	0	-1	-1	0	1	0	0	0
(3)	u_3	0	0	0	0	1	0	0	1	1	1	1	0	0	1	0	0
(4)	u_4	0	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0
(5)	u_5	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1

The first line of this table is nonnegative, but it does not correspond to a solution, since the column indicated by an arrow contains negative elements. The reader can verify that by performing a dual-simplex iteration with the circled (-1) as pivot (see (16.81)), line (1) being the pivoting line, we obtain the following table:

(22.18)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)
			g	u_1	u_2	u_3	u_4	u_5	x_{11}	x_{12}	x_{21}	x_{22}	φ_1	φ_2	φ_3	φ_4	φ_5
(0)	g	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
(1)	x_{11}	1	0	-1	0	0	0	0	1	1	0	0	-1	0	0	0	0
(2)	u_2	-1	0	0	1	0	0	0	0	0	-1	-1	0	1	0	0	0
(3)	u_3	-1	0	1	0	1	0	0	0	0	1	1	1	0	1	0	0
(4)	u_4	0	0	0	0	0	1	0	0	1	0	1	0	0	0	1	0
(5)	u_5	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	1

The above table does not represent a solution since the line indicated by an arrow is not nonnegative. We could equally take line (2) but will choose line (3) as pivot. In this line there is no negative element which means, as shown in Section 16.3, that the program has no solution. In line (3) at the intersection with columns (11)–(15) we obtain a direction $[V_1]$ of the extreme ray of the polyhedron of the constraints of a dual program such as (21.8), namely,

$$(22.19)\qquad [V_1]'_{1\times m} = [1 \quad 0 \quad 1 \quad 0 \quad 0]\,.$$

In example (22.16) the matrix $[d]_{m\times p}$ and the vector $[b]_{m\times 1}$ of program (22.3) are the following:

$$(22.20)\qquad [d] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -2 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \end{bmatrix}, \qquad [b] = \begin{bmatrix} -1 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

The constraint of the integer program such as (21.24) generated by this iteration will, after replacing (22.19) and (22.20) in (21.24) and after taking the transpose, be

$$(22.21)\qquad \underbrace{[1 \quad 0 \quad 1 \quad 0 \quad 0]}_{[V_1]'}.\left(\underbrace{\begin{bmatrix} -1 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{[b]} - \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -2 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -2 \end{bmatrix}}_{[d]} . \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}}_{[w]}\right) \geqslant 0\,.$$

It follows that

$$(22.22)\qquad -1+2w_1 \geqslant 0\,.$$

We add constraint (22.22) to the constraints of (22.15). Program (22.15), corresponding to (21.20) in the theoretical explanation of Section 21, is expressed

$$(22.23)\qquad \begin{aligned} &(1)\quad \check{Z}^{(2)} = \max_{w_0,\, w_1,\, w_2,\, w_3,\, w_4} [\check{Z} = w_0]\,, \\ &(2)\quad w_0+w_1+w_2+w_3+w_4 \leqslant 0\,, \\ &(3)\quad 2w_1 \geqslant 1\,, \\ &(4)\quad w_1\,, w_2\,, w_3\,, w_4 = 0 \text{ or } 1\,. \end{aligned}$$

In the integer program (22.23) let us observe that the variable w_0 is not constrained to be nonnegative. Since the coefficients of the constraints are integer, the form of constraint (2) and of the economic function (1) of (22.22) ensures that w_0 will be integer for the optimum. To take account, in Gomory's method for example, of the fact that w_0 is not constrained to be nonnegative, it will be sufficient for a solution of (22.22) to consider a simplex table (16.8) in which the basic variables, with the eventual exception of w_0, are nonnegative.

If we add nonnegative deviation variables t_1 and t_2 to constraints (2) and (3) of (22.23) the initial simplex table (see 16.8) becomes, after the transformation of (22.23),

(22.24)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(0)			$\check{z}$	w_0	w_1	w_2	w_3	w_4	t_1	t_2
(1)	$\check{z}$	0	1	-1	0	0	0	0	0	0
(2)	t_1	0	0	$\textcircled{1}$	1	1	1	1	1	0
(3)	t_2	-1	0	0	-2	0	0	0	0	1

.

This table does not correspond to a solution of (22.23) since $t_2 < 0$. In the algorithm we are considering the first simplex iteration is always made by taking as pivot the circled element (1) at the intersection of line (2) and column (2) so as to bring w_0 into the basis. This table then becomes

(22.25)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
			$\check{z}$	w_0	w_1	w_2	w_3	w_4	t_1	t_2
(1)	$\check{z}$	0	1	0	1	1	1	1	1	0
(2)	w_0	0	0	1	1	1	1	1	1	0
(3)	t_2	-1	0	0	$\textcircled{-2}$	0	0	0	0	1

.

As a result of this initial iteration line (1) is always nonnegative. This will occur each time that the vector of cost $[e]_{p\times 1}$ of the integer variables $[w]_{p\times 1}$ is nonpositive in an MIP having the general structure of (21.18). We now perform a dual-simplex iteration, since $t_2 < 0$ and line (1) of the table is nonnegative, the reader being left to verify that the circled element (-2) is the pivot. We obtain

(22.26)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
			$\check{z}$	w_0	w_1	w_2	w_3	w_4	t_1	t_2
(1)	$\check{z}$	-1/2	1	0	0	1	1	1	1	1/2
(2)	w_0	-1/2	0	1	0	1	1	1	1	1/2
(3)	w_1	1/2	0	0	1	0	0	0	0	-1/2

.

The solution given by the column corresponding to the second member of this table is optimal for (22.23) in which we have replaced constraint (4) by $[w]_{4\times 1} \geqslant [0]_{4\times 1}$; indeed, as we have already said, w_0 is not constrained to be nonnegative. We now add a Gomory cut that the reader can easily calculate, in the same way as in (19.23), beginning with line (3) of (22.26). After a dual iteration that the reader is left to perform, this table then becomes

(22.27)

			(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
			$\check{Z}$	w_0	w_1	w_2	w_3	w_4	t_1	t_2	t_3
(1)	$\check{Z}$	-1	1	0	1	1	1	1	1	0	2
(2)	w_0	-1	0	1	0	1	1	1	1	0	2
(3)	w_1	1	0	0	1	0	0	0	0	0	-1
(4)	t_2	1	0	0	0	0	0	0	0	1	-2

.

The above table corresponds to an optimal solution of (22.23), since line (1) is nonnegative and the column corresponding to the second member is composed of nonnegative integers, apart from w_0 that need not be nonnegative. This optimal solution is

$$(22.28) \qquad w_1 = 1, \quad w_2 = 0, \quad w_3 = 0, \quad w_4 = 0, \qquad \check{Z}^{(2)} = w_0 = -1.$$

The reader will continue the algorithm by solving the slave program (22.14), replacing w_1, w_2, w_3, w_4 by the values given by (22.28). The optimal solution of this program is

$$(22.29) \qquad x_{11} = 1, \qquad x_{21} = 1, \qquad x_{12} = 0, \qquad x_{22} = 0.$$

This solution has integer values of 0 or 1 and therefore constitutes an optimal solution of the problem of the selection of masks without the need to apply Lemma 22.II. Hence we choose version 1 of both the first and second type of mask, giving a total of ($\tilde{Z} = -\check{Z}^{(2)} = 1$) defects.

Observation

The above problem represents particular structures of matrices $[d]$, $[a]$, and $[b]$ of program (22.1). In this case, Lemma 22.II shows how, from an optimal solution $[[x]\ [w]]$ of program (22.3) that is more easily solved, we can obtain an optimal solution $[[\tilde{x}]\ [w]]$ of (22.1). Many other structures for these matrices exist for which lemmas of the type of Lemma 22.II are available, a fact justifying the interest taken in these methods for solving certain combinatorial problems. The problem given above is only one of those to which the theories explained in this section can be applied.

Section 23. **Trubin's Algorithm**

1. Introduction

In this section we shall explain the algorithm of V. A. Trubin [K72] that can be applied to certain problems of programming in bivalent variables. This little known algorithm is very neat since it enables us, for example, to solve the problem of the assignment of air crews given in Section 8 in cases where the constraints have the form $[a].[x] = [1]$ with only a slight modification of a standard simplex algorithm. It is given as a supplement because its proof (though not, it should be noted, its use) requires a knowledge of combinatorics that lies outside the scope of this book. We shall, however, in due course refer the reader to more advanced works for these proofs. In any event, the understanding of this algorithm requires a grasp of various properties already explained in this part of the work.

2. Fundamental Theorem

Given the system of relations

$$(23.1)\qquad [a]_{m\times n}\cdot[x]_{n\times 1} = [1]_{m\times 1},$$

where $[1]$ is a vector $m \times 1$ with all its elements equal to 1,

$$(23.2)\qquad x_i = 0 \text{ or } 1, \qquad i = 1, 2, \dots, n.$$

Matrix $[a]$ is such that all its elements are 0 or 1. Let us observe that certain assignment problems such as that in Section 8 have constraints of the form of (23.1) and (23.2). Let a_{ij} represent the element of line i and column j in $[a]$.

Lemma 23.1

If there are two solutions represented by $[x^{(1)}]$ and $[x^{(2)}]$ for (23.1) and (23.2), such that

$$(23.3)\qquad [x^{(1)}]_{n\times 1} + [x^{(2)}]_{n\times 1} = [1]_{n\times 1},$$

then the matrix $[a]_{m\times n}$ is unimodular (refer to the definition in Section 18).

Proof

Let us take $x_i^{(1)}$ for the ith component of the vector $[x^{(1)}]$ and $x_i^{(2)}$ for that of $[x^{(2)}]$.

We can divide the set $\mathbf{S}$ of the columns of $[a]$ into two sets $\mathbf{S}_1$ and $\mathbf{S}_2$. A column $[a]^i$ belongs to $\mathbf{S}_1$ if $x_i^{(1)} = 1$ and to $\mathbf{S}_2$ if $x_i^{(2)} = 1$.

In accordance with (23.2) and (23.3) a column of $[a]$ must belong to one or other of the subsets $\mathbf{S}_1$ or $\mathbf{S}_2$. Let us then say

$$(23.4)\qquad [x^{(3)}] = \tfrac{1}{2}.[x^{(1)}] + \tfrac{1}{2}.[x^{(2)}],$$

a point of which the coordinates are a convex combination of those of $[x^{(1)}]$

and $[x^{(2)}]$. Hence this point belongs to the convex polyhedron defined by (23.1) and we have

$$[a].[x^{(3)}] = [1]. \tag{23.5}$$

That is, by substituting (23.4) in (23.5),

$$\tfrac{1}{2}.[a]_{m\times n}.[x^{(1)}]_{n\times 1} + \tfrac{1}{2}.[a]_{m\times n}.[x^{(2)}]_{n\times 1} = [1]_{m\times 1}. \tag{23.6}$$

Or again, in accordance with (23.3),

$$[a]_{m\times n}.([x^{(1)}] + [x^{(2)}])_{n\times 1} = [2]_{m\times 1}, \tag{23.7}$$

where [2] is a vector in which all the components are equal to 2.

By substituting (23.3) in (23.7) we obtain

$$[a]_{m\times n}.[1]_{n\times 1} = [2]_{m\times 1}, \qquad \text{that is } \sum_{j\times 1}^{n} a_{ij} = 2, \qquad i = 1, 2, \ldots, m. \tag{23.8}$$

In other words, since matrix $[a]$ is formed by elements equal to 0 or 1, each line contains exactly two elements equal to 1.

From this fact and because the set of columns can be divided into two disjoint subsets, in the same way that if two columns have a 1 in the same line they are in two different sets $\mathbf{S}_1$ and $\mathbf{S}_2$, matrix $[a]$ is unimodular. This is proved by Heller–Tompkins's[1] theorem. The above lemma would be equally true if, instead of being composed only of 1, there were 0's in the right member of (23.1).

Lemma 23.II

Let there be two points with integer coordinates $[x^{(1)}] \neq [x^{(2)}]$, (that is to say that they differ by at least one component). These points are both defined by the intersection of k of the $m+2n$ hyperplanes corresponding to the three following constraints. There may be more than two hyperplanes passing through these two points.

Let

$$[a]_{m\times n}.[x]_{n\times 1} = [1]_{m\times 1}, \tag{23.9}$$

$$[x]_{n\times 1} \geqslant [0]_{n\times 1}, \tag{23.10}$$

$$[x]_{n\times 1} \leqslant [1]_{n\times 1}. \tag{23.11}$$

We necessarily have $k \geqslant m$. If $k = m$, condition (23.2) is satisfied and matrix $[a]$ is unimodular. This case is of no consequence and from now on we shall take $k > m$.

Let us assume that it is not possible to find $k+1$ hyperplanes with the three

[1] See the article by Heller and Tompkins, *in* "Linear Inequalities and Related Systems" (Kuhn and Tucker, eds.), Princeton Univ. Press, Princeton, New Jersey, 1956.

given constraints passing through these points (we must have $k > m$, since the planes (23.9) pass there and we exclude $k = m$). Then, all the vertices of the polyhedron defined by these constraints and situated at the intersection of the k hyperplanes of the constraints that pass through the two points are points with integer coordinates. Therefore, a fortiori, they satisfy (23.1) and (23.2).

Proof

Let i_α, $\alpha = 1, 2, \dots, k-m$ be the indices of the components of $[x^{(1)}]$ and $[x^{(2)}]$ that are identical, these points being at the intersection of $k-m$ hyperplanes such as (23.10) and (23.11) and m hyperplanes (23.9), that is, in all, k hyperplanes. We can then say

$$(23.12) \qquad x_{i_\alpha}^{(1)} \neq x_{i_\alpha}^{(2)}, \qquad \alpha = k-m+1, \dots, n.$$

All the vertices $[x]$ that satisfy the three constraints and are situated on the same k hyperplanes as the two points must be such that

$$(23.13) \qquad x_{i_\alpha} = x_{i_\alpha}^{(1)} = x_{i_\alpha}^{(2)}, \qquad \alpha = 1, 2, \dots, k-m.$$

We can also say that relation (23.12) is expressed as

$$(23.14) \qquad x_{i_\alpha}^{(1)} + x_{i_\alpha}^{(2)} = 1, \qquad \alpha = k-m+1, \dots, n.$$

Let us return to Eq. (23.1) and consider the points that satisfy (23.13):

$$(23.15) \qquad [[a]^{i_{k-m+1}} \dots [a]^{i_n}]_{m\times(n-m+k)} . [x_{i_{k-m+1}}, \dots, x_{i_n}]_{(n-m+k)\times 1}$$
$$= [1]_{m\times 1} - [[a]^{i_1} \cdots [a]^{i_{k-m}}]_{m\times(k-m)} . [x_{i_1}, \cdots, x_{i_{k-m}}]_{(k-m)\times 1} .$$

The second member of (23.15) is a constant for all points that satisfy (23.1) and (23.13) that are at the intersection of k of the $2n+m$ hyperplanes defining the polyhedron of the constraints. The above equation provides two solutions: $[x_{i_\alpha}^{(1)}]$ and $[x_{i_\alpha}^{(2)}]$, $\alpha = k-m+1, \dots, n$, by hypothesis and such that

$$(23.16) \qquad x_{i_\alpha}^{(1)} + x_{i_\alpha}^{(2)} = 1, \qquad \alpha = k-m+1, \dots, n,$$

in accordance with (23.14). By using Lemma 23.I, the matrix

$$(23.17) \qquad [[a]_{m\times 1}^{i_{k-m+1}} \quad [a]_{m\times 1}^{i_{k-m+2}} \quad \dots \quad [a]_{m\times 1}^{i_n}]_{m\times(n-m+k)}$$

is unimodular. All the extreme points $[x_{i_{k-m+1}}, \dots, x_{i_n}]$ of the polyhedron defined by Eq. (23.15) and $0 \leqslant x_{i_\alpha} \leqslant 1$, $\alpha = k-m+1, \dots, n$, are therefore integer, since the second member is integer. As all these points are situated on the same k hyperplanes of the polyhedron of constraints (23.1) and (23.2), the lemma is proved.

Let us give an interpretation of this lemma that will make it easy for us to prove the following fundamental theorem. We have just proved that all the vertices of the polyhedron defined by

$$
(23.18) \qquad
\begin{aligned}
&[a]_{m\times n}\cdot[x]_{n\times 1} = [1]_{m\times 1},\\
&[x] \geqslant [0],\\
&[x] \leqslant [1],\\
&x_{i_\alpha} = x^{(1)}_{i_\alpha}; \qquad \alpha = 1, 2, \ldots, k-m,
\end{aligned}
$$

have integer coordinates. Let us observe that the initial polyhedron defined by (23.9)–(23.11) contains the restricted polyhedron defined above, in particular its vertices, which all possess integer coordinates, and its edges. Hence there is always a vertex of the initial polyhedron with integer coordinates adjacent to a vertex $[x^{(1)}]$ with integer coordinates of this initial polyhedron. Also, if $[x^{(2)}]$ is a vertex with integer coordinates there is always a path between adjacent points from $[x^{(1)}]$ to $[x^{(2)}]$.

Theorem 23.III (*Fundamental Theorem*)

Let us take the program with bivalent values defined by

$$
(23.19) \qquad
\begin{aligned}
&(0) \quad [\mathrm{MAX}]\; g = [c]'\cdot[x],\\
&(1) \quad [a]_{n\times n}\cdot[x]_{n\times 1} = [1]_{m\times 1},\\
&(2) \quad x_j = 0 \;\text{ or }\; 1, \qquad j = 1, 2, \ldots, n,
\end{aligned}
$$

and let the polyhedron of the constraints be

$$
(23.20) \qquad
\begin{aligned}
&(1) \quad [a]\cdot[x] = [1],\\
&(2) \quad [0] \leqslant [x] \leqslant [1].
\end{aligned}
$$

If $[x^{(1)}]$ is a nonoptimal solution of this program then there is a vertex with integer coordinates of (23.20) adjacent to $[x^{(1)}]$ that gives a better solution.

Proof

This is directly derived from Lemma 22.I, which shows that there is always at least one vertex with integer coordinates adjacent to $[x^{(1)}]$. If $[x^{(1)}]$ is not the optimal solution this is shown by a better value of the economic function for one of the adjacent vertices with integer coordinates. And there is always a path between adjacent points from $[x^{(1)}]$ to the optimal solution $[\hat{x}]$.

We shall now prove a theorem that shows the difficulty of obtaining an optimal solution with Trubin's algorithm, which will then be explained. A solution of a linear program in $\mathbf{R}^n$ is called *degenerate* if it is the intersection of more than two hyperplanes delimiting the polyhedron of the constraints.

Theorem 23.IV

Every solution with integer values of the program in bivalent variables (23.19) is degenerate. Every solution is, in effect, situated at the intersection of $m+n$ hyperplanes of the polyhedron of the constraints (23.20).

Proof

There are $m+2n$ constraints given by (23.20). Every vertex with integer coordinates equal to 0 or 1 of this polyhedron satisfies the m constraints (1) and the n constraints (2), making a total of $m+n$. Q.E.D.

In the simplex method a basis is associated with n hyperplanes, the intersection of which provides a basic solution. This is the equivalent of saying that for a vertex with integer coordinates of a *Trubin polyhedron* such as (23.20) there are several bases and finally a large number, since each solution is degenerate. Theorem 23.III shows that if a solution $[x^{(1)}]$ is not optimal there is a better adjacent vertex with integer coordinates, but there is no guarantee that this vertex is adjacent in the basis defining $[x^{(1)}]$ and that this will not usually be the case. This will serve to explain Trubin's algorithm that follows.

2. Trubin's Algorithm

This solves a program such as (23.17) by a method derived from the simplex.

(a) Obtain a solution $[x^{(1)}]$ of (23.19), for which it is sufficient to consider the program:

$$\text{(0)}\quad [\text{MAX}]\, g = [c]'.[x] + [M].[\varphi],$$

$$(23.21)\qquad \text{(1)}\quad [a]_{m\times n}.[x]_{n\times 1} + [1]_{m\times m}.[\varphi]_{m\times 1} = [1]_{m\times 1},$$

$$\text{(2)}\quad x_j = 0 \text{ or } 1, \qquad j = 1, \dots, n,$$

$$(23.22)\qquad \varphi_i = 0 \text{ or } 1, \qquad i = 1, \dots, m.$$

This program gives the integer solution $x_j^{(1)} = 0$, $j = 1, \dots, n$ and $\varphi_i^{(1)} = 1$, $i = 1, \dots, m$. If M is a very large cost this will not be an optimal solution. The program defined above provides the same optimal solution as (23.19), namely, $[x] = [\hat{x}]$ and $[\varphi] = [0]$.

(b) A solution $[x^{(1)}]$ of (23.21) is obtained in a simplex table. Theorem 23.III shows that if it is not optimal there is a better solution adjacent to it. In accordance with Theorem 23.IV the solution $[x^{(1)}]$ is degenerate. To consider all the points adjacent to it we should consider all the bases attached to it, an extensive task that the simplex method does not perform. Indeed, a simplex table such as (16.8) only enables us to consider the n points in the basis associated with $[x^{(1)}]$ of the table. In practice we are satisfied with the basis of the simplex method and seek an adjacent integer point in this basis that improves the economic function. If one does not exist we stop. Hence we shall have a local minimum (in relation to the basis of the simplex table for this last stage).

We shall now give an instructional example to illustrate the use of Trubin's algorithm.

Example

Given the program in bivalent variables

$$(23.23)\quad \begin{aligned} &(0)\quad [\mathrm{MAX}]\ g = 4x_1+3x_2+2x_3,\\ &(1)\quad x_1+x_2 \leqslant 1,\\ &(2)\quad x_2+x_3 \leqslant 1,\\ &(3)\quad x_1+x_3 \leqslant 1,\\ &(4)\quad 0 \leqslant x_1, x_2, x_3 \leqslant 1,\\ &(5)\quad x_1, x_2, x_3 \in \mathbf{N}. \end{aligned}$$

This program[1] is shown in Fig. 23.1 and we can see that the vertices with

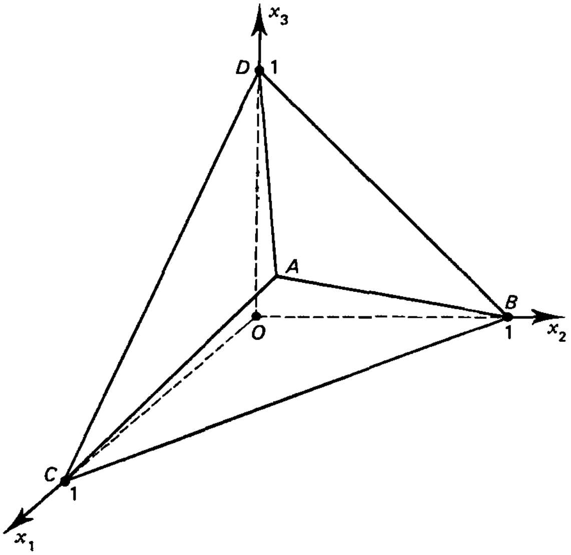


Fig. 23.1

[1] We can easily reproduce the case of program (23.1) and (23.2) by adding deviation variables μ_i, $i = 1, 2, 3$, such that $0 \leqslant \mu_i \leqslant 1$ to constraints (1), (2), and (3) of (23.23).

integer coordinates marked by heavy dots have at least one adjacent vertex with integer coordinates. If constraint (5) of (23.23) were replaced by

(23.24) $$x_1, x_2, x_3 \in \mathbf{R}^+,$$

the optimum would be point $A(x_1 = 1/2, x_2 = 1/2, x_3 = 1/2, g = 4\,1/2)$. Starting from point D, Trubin's algorithm, that is a modification of an ordinary simplex, does not lead to A (which is not integer) but to C.

At this point there is no adjacent point with integer coordinates that improves the value of the economic function, and the algorithm is stopped. The optimal solution of (23.23) is

(23.25) $$\hat{x}_1 = 1, \qquad \hat{x}_2 = 0, \qquad \hat{x}_3 = 0; \qquad \hat{g} = 4.$$

CONCLUSION

This third volume has dealt with a particularly difficult subject and we hope that the majority of our readers will have grasped the material propounded without too great an effort, for the subject is well worth the labor. Problems of a combinatorial or diophantine character, that is to say, those that have integer solutions, occur in most planning and operations research studies of the present day. We now possess the means to attack and to solve these problems, and it is only to be hoped that engineers are ready to make use of them.

We have made a very important effort in instruction, greater even than in the previous volumes. To be sure, the mathematical theory of the second part is at times difficult to grasp, but thanks to a number of examples the path should have been relatively easy. Some readers of the first two volumes, both in France and in the world generally, since these works have been translated into numerous languages, have told us that the second volume was less easy to understand than the first for those without the necessary mathematical knowledge. This is true and is to some extent accentuated in the case of the present volume. Volume 1 constituted a very elementary introduction, Volume 2 introduced moie complicated methods and models that were more specifically based on the new mathematics, and so on. But we believe that the reader of the first two volumes who has improved his knowledge in the interval may reap advantage from the present one.

Volume 4 is now in course of preparation and our small team has been augmented for it by D. Coster, Engineer I.M.A.G., who has been given the responsibility for a number of important chapters. We have decided that this volume will deal with nonlinear programming, a difficult subject for which educational literature is not extensive despite the presence of nonlinear problems in a great many economic phenomena. As in the first three volumes, the first part will present problems in their actual context, while the second part will be devoted to the difficulties, the proofs and the more complicated calculations.

If possible we shall not end with Volume 4: there are so many new models and attractive methods that make their appearance every year in operations research.

A few years ago analysts, economists and informaticians were beginning to say, "Operations research is out of date." In fact, it has never ceased to spread under various guises and has remained the scientific basis for management and administration. Informatics has not replaced operations research for they are not competitors but constitute two different branches of research and its application that should work together; from their combination scientific management is born. Now that informatics has reached the stage of systems, the arrangement of its methods and their utilization has become a whole group of problems that can most frequently be solved by the most advanced methods of operations research. Management, administration, informatics, piloting of systems, regulation, bionics, and so forth, are domains of cybernetics in an even wider sense than that envisaged by Norbert Wiener. The use of mathematics is no passing fashion but the very essence of science. Indeed, some very interesting efforts are being made to reconcile formal reasoning with indistinct concepts from which there is derived the growing success of the *fuzzy sets theory*. This has been used the better to express and analyze human behavior, so difficult to specify and to measure, and also to produce a fuller understanding of thought.

The critics of mathematics and especially of the new mathematics cannot properly have understood them. Why else should they oppose the precision and the ever increasing generalization and the economy of thought? To be sure we must not exaggerate and regard mathematics as a language that would have charmed the Sphinx, but should be ready to make use of its more precise symbolism and its capacity to avoid omissions and redundancies.

It is the power of abstraction that constitutes the superiority of man over the animal and this is derived from language and communication. A word is already a mathematical formula and a phrase is a model. The progress of thought depends on the quality of these formulas and these models. Now that conversation has been extended from man with man to that of man with the machine, thereby increasing the possibilities of communication, operations research will multiply its means and its results. It is needed for our progress toward a better moral and material existence.

Appendix. OPERATIONS ON MODULO 1 EQUATIONS

1. Modulo 1 Addition of Two Real Numbers

Definition

The modulo 1 addition of two real numbers a and b, expressed as $a +_1 b$ is the remainder of the division by 1 of their common addition $a+b$.

Examples

(A1.1) $$3.57 +_1 1.86 = 0.43,$$

(A1.2) $$(-3.88) +_1 5.64 = 0.76,$$

(A1.3) $$(-3.52) +_1 (-2.31) = 0.17,$$

(A1.4) $$(-6.15) +_1 6 = 0.85.$$

Notation

We give the notation $\langle a \rangle$ to the largest integer less than or equal to a real number a. The noninteger part $a - \langle a \rangle$ will be indicated by $\{a\}$.

Thus

(A1.5) $$\{a\} = a - \langle a \rangle.$$

For a matrix

(A1.6) $$[a]_{m\times n} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix},$$

we shall use the notation

(A1.7) $$\{[a]_{m\times n}\} = \begin{bmatrix} \{a_{11}\} & \{a_{12}\} & \cdots & \{a_{1n}\} \\ \{a_{21}\} & \{a_{22}\} & \cdots & \{a_{2n}\} \\ \cdots & \cdots & \cdots & \cdots \\ \{a_{m1}\} & \{a_{m2}\} & \cdots & \{a_{mn}\} \end{bmatrix}.$$

Examples

(A1.8) (1) $a = 3.28, \qquad \langle a\rangle = 3, \qquad \{a\} = a-\langle a\rangle = 0.28,$

(A1.9) (2) $a = -3.63, \qquad \langle a\rangle = -4, \qquad \{a\} = a-\langle a\rangle = 0.37,$

(A1.10) (3) $a = -2, \qquad \langle a\rangle = -2, \qquad \{a\} = a-\langle a\rangle = 0.$

(A1.11) (4) $[A] = [5.3 \quad -1.8 \quad -2 \quad 0{,}1],$

(A1.12) $\langle [A]\rangle = [5 \quad -2 \quad -2 \quad 0],$

(A1.13) $\{[A]\} = [0.3 \quad 0{,}2 \quad 0 \quad 0{,}1].$

(A1.14) (5) $[B] = \begin{bmatrix} 0{,}31 & -2{,}53 \\ 8.60 & -0{,}12 \end{bmatrix},$

(A1.15) $\langle [B]\rangle = \begin{bmatrix} 0 & -3 \\ 8 & -1 \end{bmatrix},$

(A1.16) $\{[B]\} = \begin{bmatrix} 0.31 & 0{,}47 \\ 0.60 & 0.88 \end{bmatrix}.$

Property 1

If the real number a is an integer, we have $\{a\} = 0$.
This is obvious from the definition of the notation $\{a\}$.

(A1.17) $\{a+b\} = \{b\}$ if a is an integer.

Property 2

(A1.18) $\{\{a\}\} = \{a\}$

since the largest integer less than or equal to $\{a\} = 0$.

Property 3

(A1.19) $\{-a\} = 1-\{a\}.$

Indeed

(A1.20) $-a = \langle -a\rangle + \{-a\}$

and also

(A1.21) $-a = -\langle a\rangle - \{a\}.$

Then

(A1.22) $\langle -a\rangle = -\langle a\rangle - 1$ if a is a noninteger

(for instance, $\langle -2, 6\rangle = -3 = -\langle 2, 6\rangle - 1$).

Thus, by comparing (A1.20) and (A1.21) and by then taking account of (A1.22), we can say

(A1.23) $$\begin{aligned}\langle -a\rangle + \{-a\} &= -\langle a\rangle - \{a\}\\ &= \langle -a\rangle + 1 - \{a\}\,.\end{aligned}$$

By eliminating $\langle -a\rangle$ in (A1.23) we find (A1.19).

Property 4

(A1.24) $$a +_1 b = \{a+b\}\,.$$

Indeed

(A1.25) $$a+b = \langle a+b\rangle + \{a+b\}\,.$$

Since $(a+b)$ is an integer its remainder after division by 1 is 0, which means that the remainder of the division of $a+b$ is the equivalent of the remainder of $\{a+b\}$, since $0 \leqslant \{a+b\} < 1$.

Property 5

(A1.26) $$a +_1 b = \{\{a\} + \{b\}\}.$$

From Property 4 we have

$$a +_1 b = \{a+b\} = \{\langle a\rangle + \{a\} + \langle b\rangle + \{b\}\}\,.$$

But from Property 1, since $\langle a+b\rangle$ is an integer, the remainder is $\{\{a\}+\{b\}\}$.

Property 6

(A1.27) $$\{a\} +_1 \{b\} = a +_1 b\,.$$

From Property 4 we have

(A1.28) $$\{a\} +_1 \{b\} = \{\{a\} + \{b\}\}\,,$$

by making $a = \{a\}$ and $b = \{b\}$. Hence, in accordance with Property 5, we have

(A1.29) $$\{a\} +_1 \{b\} = \{\{a\} + \{b\}\} = a +_1 b\,.$$

We can sum up Properties 4–6 as

(A1.30) $$a +_1 b = \{a+b\} = \{\{a\} + \{b\}\} = \{a\} +_1 \{b\}.$$

Property 7

(A1.31) $$a +_1 b = \{\{a\} +_1 \{b\}\}\,.$$

In fact, from (A1.30) we have

(A1.32) $$\{\{a\} +_1 \{b\}\} = \{\{a+b\}\}\,,$$

which is equal to $\{a+b\}$ in accordance with Property 2 and is also equal to $a+_1 b$ from (A1.30).

To sum up,

$$(\text{A1.33}) \qquad a +_1 b = \{a + b\} = \{\{a\} + \{b\}\} = \{a\} +_1 \{b\}$$
$$= \{\{a\} +_1 \{b\}\}.$$

Property 8

It can easily be proved, as the reader can do if he wishes, that if the real number b is an integer, we have

$$(\text{A1.34}) \qquad \{a.b\} = \{\{a\}.b\} .$$

2. Associativity of Modulo 1 Addition

To prove

$$(\text{A1.35}) \qquad (a +_1 b) +_1 c = a +_1 (b +_1 c),$$

it is sufficient to make use of the definition of modulo 1 addition.

3. Abelian Group of the Noninteger and Zero Parts of Real Numbers for Modulo 1 Addition

To show that these parts form an abelian group for modulo 1 addition, for any three noninteger parts $\{a\}$, $\{b\}$, and $\{c\}$, we must prove that we can verify associativity, that a unit exists, that every noninteger part has an inverse, and, finally, that there is commutativity.

The associativity

$$(\text{A1.36}) \qquad (\{a\} +_1 \{b\}) +_1 \{c\} = \{a\} +_1 (\{b\} +_1 \{c\}),$$

is verified for these three noninteger parts since this property is true for the modulo 1 addition of every real number. At the same time we make the simple check that there is commutativity:

$$(\text{A1.37}) \qquad \{a\} +_1 \{b\} = \{b\} +_1 \{a\} .$$

The unit element is $\{0\} = 0$; indeed,

$$(\text{A1.38}) \qquad \{a\} +_1 0 = \{\{a\} + 0\} \qquad \text{from Property 4}$$
$$= \{\{a\}\} = \{a\} \qquad \text{from Property 2.}$$

Hence we have

$$(\text{A1.39}) \qquad \{a\} +_1 0 = \{a_1\} = 0 +_1 \{a_1\} ,$$

in accordance with the commutativity. Thus 0 is indeed the unit.

The inverse of $\{a\}$ is $\{-a\}$. Indeed,

$$\begin{aligned} (\mathrm{A}1.40) \qquad \{a\} +_1 \{-a\} &= \{\{a\} + \{-a\}\} && \text{from Property 4} \\ &= \{\{a\} + 1 - \{a\}\} && \text{from Property 3} \\ &= \{1\} = 0 && \text{from Property 1.} \end{aligned}$$

Hence, by taking the commutativity into account, we have

$$(\mathrm{A}1.41) \qquad \{a\} +_1 \{-a\} = \{-a\} +_1 \{a\} = 0.$$

4. Modulo 1 Addition of Matrices

This is defined in the same way as this operation for the real numbers that form the elements of these matrices. Thus

$$(\mathrm{A}1.42) \qquad [1.2 \quad 1.3 \quad -0.4] +_1 [0.6 \quad 0.2 \quad 1] = [0.8 \quad 0.5 \quad 0.6],$$

$$(\mathrm{A}1.43) \qquad \begin{bmatrix} 1.2 & 1.3 \\ -0.4 & 0.6 \end{bmatrix} +_1 \begin{bmatrix} 2.4 & 1 \\ 0 & 2.3 \end{bmatrix} = \begin{bmatrix} 0.6 & 0.3 \\ 0.6 & 0.9 \end{bmatrix}.$$

5. Solution of Modulo 1 Equations

Let us consider the following equation where x, a, $b \in \mathbf{R}$:

$$(\mathrm{A}1.44) \qquad \{x+a\} = \{b\}\ .$$

From Property 1, the general solution of (A1.44) is

$$(\mathrm{A}1.45) \qquad x = b-a+k, \qquad \text{with} \quad k \in \mathbf{Z}.$$

For example, the equation

$$(\mathrm{A}1.46) \qquad \{x+1.2\} = \{4.8\},$$

has for its solutions $x = 0.6+k$, that is also

$$(\mathrm{A}1.47) \qquad x = \dots, -1.4, \quad -0.4, \quad 0.6, \quad 1.6, \quad 2.6, \dots .$$

Solution of Optimization Problems in Modulo 1 Equations

We shall now give an algorithm for dynamic programming that enables us to solve such problems as the asymptotic problem (see (20.18)) with the following structure

$$(\mathrm{A}1.48) \qquad \begin{aligned} &(1) \quad [\mathrm{MIN}]\, f = \bar{c}_1 x_1 + \bar{c}_2 x_2 + \dots + \bar{c}_n x_n, \\ &(2) \quad \left\{\frac{a_1}{\delta} x_1 + \frac{a_2}{\delta} x_2 + \dots + \frac{a_4}{\delta} x_n\right\} = \left\{\frac{\lambda}{\delta}\right\}, \end{aligned}$$

with

$\bar{c}_i, i=1,\dots,n$; $a_i, i=1,\dots,n$; $x_i, i=1,\dots,n$; $\delta \in \mathbf{N}$; $\lambda \in \mathbf{Z}$ and $a_i < \delta, i=1,\dots,n$.

Let $\Lambda_{n-1}(\xi)$ be the optimal value of the economic function of the following program:

$$(\text{A1.49})\qquad \begin{array}{ll} (1) & \Lambda_{n-1}(\xi) = [\text{MIN}]\, f = c_1 x_1 + \dots + c_{n-1}x_{n-1}\,, \\ (2) & \left\{\dfrac{a_1}{\delta}x_1 + \dots + \dfrac{a_{n-1}}{\delta}x_{n-1}\right\} = \{\xi\}\,, \qquad x_i \in \mathbf{N}, \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad i = 1, \dots, n-1\,. \end{array}$$

In accordance with (A1.45), line (2) of Eq. (A1.48) can be expressed for any second member $\{\xi\}$

$$(\text{A1.50})\qquad \frac{a_1}{\delta}x_1 + \dots + \frac{a_{n-1}}{\delta}x_{n-1} = \xi - \frac{a_n}{\delta}x_n + k\,, \qquad k \in \mathbf{Z}\,.$$

That is, by using Property 1,

$$(\text{A1.51})\qquad \left\{\frac{a_1}{\delta}x_1 + \dots + \frac{a_{n-1}}{\delta}x_{n-1}\right\} = \left\{\xi - \frac{a_n}{\delta}x_n\right\}.$$

Let us use $\Lambda_n(\xi)$ for the optimal value of the economic function of program (A1.48) in which b/δ in the second member of line (2) has been replaced by ξ.

Then by using Bellman's[1] condition for optimality as well as (A1.51), the program can be expressed

$$(\text{A1.51a})\qquad \begin{array}{ll} (1) & \Lambda_n\left(\dfrac{b}{\delta}\right) = \underset{x_n}{[\text{MIN}]}\left(\bar{c}_n x_n + \Lambda_{n-1}\left(\dfrac{b}{\delta} - \dfrac{a_n}{\delta}x_n\right)\right), \\ (2) & x_n \in \mathbf{N}\,. \end{array}$$

We can easily give a general form to this relation of recurrence, and we have

$$(\text{A1.52})\qquad \Lambda_p(\xi) = \underset{x_p}{[\text{MIN}]}\ \bar{c}_p x_p + \Lambda_{p-1}\left(\xi - \frac{a_p}{\delta}x_p\right), \qquad 1 \leqslant p \leqslant n\,.$$

Let us observe that relation (A1.52) is identical except for a few minor details with the one established for the problem given on page 86 of Volume 2 and also on page 86 of the present volume. We shall now determine the set of the values of ξ for which (A1.52) has to be evaluated. In the first place, ξ must be of the form

$$(\text{A1:53})\qquad \xi = \alpha/\delta\,, \qquad \alpha \in \mathbf{Z}\,,$$

in order that line (2) of Eq. (A1.49) may have a solution. Further, $\{\xi\} = \{\alpha/\delta\}$ can only take δ values $0, 1/\delta, 2/\delta, \dots, (\delta-1)/\delta$; we shall also take ξ

[1] See Volume 2, multistage optimization, page 331.

having the form

(A1.54) $\quad \xi = \alpha/\delta, \qquad \alpha = 0, 1, 2, \ldots, \delta - 1,$

and

(A1.55) $\quad x_p = 0, 1, 2, \ldots, \delta - 1.$

Example

Let us take the optimization problem:

$$\text{(A1.56)} \quad \begin{array}{ll} (1) & [\text{MIN}]\, f = 5x_1 + 2x_2 + 7x_3 , \\ (2) & \{1/3x_1 + 2/3x_2 + 2/3x_3\} = \{1/3\} . \end{array}$$

Using the notation $x^*(\xi)$ for the optimal solution of (A1.52), we successively calculate $\Lambda_1(\xi), \Lambda_2(\xi), \Lambda_3(1/3)$ and group the calculations in tabular form:

(A1.57)

	(1)	(2)	(3)	(4)	(5)	(6)	(7)
(0)	ξ	$\Lambda_1(\xi)$	$x_1^*(\xi)$	$\Lambda_2(\xi)$	$x_2^*(\xi)$	$\Lambda_3(\xi)$	$x_3^*(\xi)$
(1)	0	0	0	0	0	0	0
(2)	1/3	5	1	4	2	4	0
(3)	2/3	10	2	2	1	2	0

For example, in table (A1.57) we have

$$\text{(A1.58)} \quad \Lambda_2(0) = \text{MIN}(\Lambda_1(0-2/3.0) + 2.0,\ \Lambda_1(0-2/3.1) + 2.1,\ \Lambda_1(0-2/3.2) + 2.2).$$

That is, by substituting the values of $\Lambda_1(\xi)$ taken from column (2) of (A1.57),

(A1.59) $\quad \Lambda_2(0) = 0 \qquad \text{for } x_2(0) = 0.$

Lastly we obtain $\Lambda_3(1/3) = 4$ as the minimal value of the economic function of (A1.56) for $x_3^*(1/3)$. For $x_3 = 0$ we have $\Lambda_2(1/3-2/3.0) = 4$ for $x_2^*(1/3) = 2$ With $x_2 = 2$ we have $\Lambda_1(1/3-2/3.0-2/3.2) = \Lambda_1(-3/3) = \Lambda_1(0/3) = 0$ for $x_1^*(0/3) = 0$.

Hence the optimal solution is

(A1.60) $\quad x_1 = 0, \qquad x_2 = 2, \qquad x_3 = .0, \qquad f = 4.$

Let us observe that table (A1.57) will also give us the solution of problem (A1.56) for other values $\{0/3\}$ and $\{2/3\}$ of the second member of line (2) of Eq. (A1.56).

BIBLIOGRAPHY

This bibliography is a continuation of that in Volumes 1 and 2.

XII. **Programs with Integer and Mixed Values**[1]

1. Works

[K1] Beckenbach, E. F., "Applied Combinatorial Mathematics." Wiley, New York, 1964.

[K2] Berge, C., "Graphes et hypergraphes." Dunod, Paris, 1970.

[K3] Berge, C., and Ghouila-Houri, A., "Programmes, jeux et réseaux de transport." Dunod, Paris, 1962.

[K4] Birkhoff, G., "Lattice Theory." Amer. Math. Soc., Providence, Rhode Island.

[K5] Busacker, R. G., and Saaty, T. L., "Finite Graphs and Networks." McGraw-Hill, New York, 1965.

[K6] Carvallo, M., "Monographie des treillis et algèbre de Boole." Gauthier-Villars, Paris, 1962.

[K7] Carvallo, M., "Principes et applications de l'analyse booléenne." Gauthier-Villars, Paris, 1965.

[K8] Dantzig, G. B., "Linear Programming and Extensions." Princeton Univ. Press, Princeton, New Jersey, 1960. [French translation: Dunod, Paris, 1966.]

[K9] Dermiane, J. C., and Pair, C., "Problèmes de cheminement dans les graphes." Dunod, París, 1971.

[K10] Faure, R., and Heurgon, E., "Structures ordonées et algèbres de Boole." Gauthier-Villars, Paris, 1971.

[K11] Faure, R., Denis-Papin, M., and Kaufmann, A., "Aide-mémoire de mathématiques nouvelles" (2 volumes). Dunod, Paris, 1964.

[K12] Faure, R., Denis-Papin, M., and Kaufmann, A., "Cours de calcul booléen appliqué." Albin-Michel, Paris, 1964.

[K13] Flegg, H. G., "L'algèbre de Boole et son utilisation." Dunod, Paris, 1967.

[K14] Hammer, P. L., and Rudéanu, S., "Boolean Methods in Operations Research and Related Areas." Springer Publ., New York, 1970. [French edition: Dunod, Paris.]

[K15] Henry-Labordère, A., "Partitioning Algorithms in Mixed and Pseudo Mixed Integer Programming." Ph.D. Thesis, Rennsselaer Polytechnic Inst., Troy, New York, 1968.

[1] When the present volume was originally published the bibliography of books on this subject was very limited, in contrast to the numerous articles dealing with it. Almost all the works cited here are concerned with modern mathematics.

[K16] Heurgon, E., "Programmation linéaire en nombres entiers." Thèse 3^{e} cycle, Math. Appl. Analyse Numérique, Fac. Sciences, Paris, 1967.

[K17] Hu, T. C., "Integer Programming and Network Flows." Addison-Wesley, Reading, Massachusetts, 1970.

[K18] Kaufmann, A., "Introduction à la combinatorique en vue des applications." Dunod, Paris, 1968.

[K19] Kaufmann, A., and Coster, D., "Exercices de combinatorique avec solutions" (3 volumes). Dunod, Paris, 1971.

[K20] Kaufmann, A., and Précigout, M., "Cours de Mathématiques nouvelles pour le recyclage des ingénieures et cadres." Dunod, Paris, 1966.

[K21] Kaufmann, A., and Cullmann, G., "Mathématiques nouvelles pour le recyclage des parents" and "Problèmes simples de mathématiques nouvelles pour le recyclage des parents." Dunod, Paris, 1970.

[K22] Kuntzmann, J., "Algèbre de Boole." Dunod, Paris, 1965.

[K23] Roy, B., "Algèbre moderne et Théorie des graphes" (2 volumes). Dunod, Paris, 1970.

[K24] Thiriez, H. M., "Air Line Crew Scheduling: A Group Theoretic Approach." Ph.D. Thesis, M.I.T., FTL R.69.1, Cambridge, Massachusetts, 1969.

2. Articles

[K25] Balas, E., An additive algorithm for solving linear programs with 0–1 variables. *J. Oper. Res. Soc. Amer.* **13**(4), 1965.

[K26] Balas, E., Discrete programming by the filter method. *J. Oper. Res. Soc. Amer.* **15**(5), 1967.

[K27] Balas, E., Integer programming and convex analysis: intersection cuts from outer polars. *Math. Programming* **2**(3), 330–382, 1972.

[K28] Balinski, M. L., Integer programming, methods, uses, computations. *Management Sci.* **12**(3), 253–313, 1965.

[K29] Benders, J. F., Partitioning procedures for solving mixed variables programming problems. *Numeriske Mathematik* **4**, 238–252, 1962.

[K30] Bertier, P., Phomg Truan Nghiem, and Roy, B., Programmes linéaires en nombres entiers et procédures S.E.P. *METRA* **4**(3), 1965.

[K31] Delmas, L., and Henry-Labordère, A., Solution exacte du problème de l'enlèvement des modules. *METRA* **10**(3), 453–458, 1971.

[K32] Fiorot, J. C., and Gondran, M., Résolution des systèmes linéaires en nombres entiers. *Bull. Direction Études Recherches EDF, Sér. C*, no. 2, pp. 65–116, 1969.

[K33] Fisk, C. J., Caskey, D. L., and West, L. E., ACCEL, Automated Circuit Card Etching Layout. *Proc. IEEE* **55**(11), 1771-1982, 1967.

[K34] Garfinkel, R. S., and Nemhauser, G. L., The set partitioning problem: set covering with equality constraints. *J. Oper. Res. Soc. Amer.* **17**, 848–856, 1969.

[K35] Geoffrion, A. M., An improved implicit enumeration approach for integer programming. The Rand Corp. RM 5644-PR., June 1968.

[K36] Geoffrion, A. M., and Martens, R. E., Integer programming algorithms: a framework and state-of-the-art survey. *Management Sci.* **18**(9), 465–491, 1972.

[K37] Glover, F., A multiphase dual algorithm for the 0–1 integer programming problem. Case Inst. of Tech., Cleveland, Ohio. *Management Sci. Rep.* no. 25, 1965.

[K38] Glover, F., An algorithm for sǫlving the linear integer programming over a finite additive group with extensions to solving general linear and certain nonlinear integer programs. Univ. of California Oper. Res. Center, Berkeley, California, WP 29, June 1966.

[K39] Glover, F., Cut–search methods in integer linear programming. *Math. Programming* **2**(3), 330–382, 1972.

[K40] Gomory, R. E., All-integer programming algorithm. IBM Corp., RC 189, 1960.

[K41] Gomory, R. E., On polyhedra related to some combinatorial problems. IBM Corp., RC 2145, 1968.

[K42] Gomory, R. E., An algorithm for integer solutions to linear programs. *In* "Recent Advances in Math. Programming" (P. Wolfe and R. Graves, eds.). McGraw-Hill, 1962.

[K43] Gomory, R. E., and Hoffmann, A. J., On the convergence of an integer programming process. *Naval Res. Logist. Quart.* **10**, 121–123, 1963.

[K44] Gomory, R. E., On the relation between integer and noninteger solutions to linear programs. *Proc. Nat. Acad. Sci.* **53**, 260–295, 1965.

[K45] Gomory, R. E., and Johnson, E. L., Some continuous functions related to convex polyhedra and their applications to integer programming. *Math. Programming* **3**(1), 23–85, 1972.

[K46] Guignard, M., and Spielberg, K., Search techniques with adaptive functions for certain integer and mixed integer programming problems. *Proc. I.F.I.P.*, Edinburgh, 1968. Mathematical series.

[K47] Healy, W. C. Jr., Multiple choice programming. *J. Oper. Res. Soc. Amer.* **12**, 122–138, 1964.

[K48] Henry-Labordère, A., Optimal removal of logic modules in printed circuits board rework by linear programming. IBM Corp., TROO-1719, April 1968.

[K49] Henry-Labordère, A., and Chandra, R., Optimal mask selection by cone integer programming. Memo. IBM Corp., Poughkeepsie, Dept. B24, April 1968.

[K50] Henry-Labordère, A., and Delmas, L., Solution exacte du problème de l'enlèvement des modules. *METRA* **10**(3), 453–458, 1971.

[K51] Henry-Labordère, A., and Zerhouni, C. M., Décisions bayésiennes avec information incomplète. *METRA* **11**(4), 1972.

[K52] House, R. W., Nelson, L. D., and Rado, T., Computer studies of a certain class of linear integer problems. *In* "Recent Advances in Optimization Techniques" (A. Lavi and T. Vogl, eds.). Wiley, New York, 1966.

[K53] Johnson, S. M., Optimal two- and three-stages production schedule with set-up times included. *Naval Res. Logist. Quart.* **1**, 61–86, 1954.

[K54] Land, A. H., and Doig, A. G., An automated method for solving discrete programming problems. *Econometrica* **28**, 497–520, 1960.

[K55] Lathrop, J. W., Harrell, S. A., Ables, B. D., and Streater, S., Computer impact on photomask technology. Kodak Photolithographic Symposium, June 1, 1967.

[K56] Lawler, E. L., and Bell, M. D., A method for solving discrete optimization problems. *J. Oper. Res. Soc. Amer.* **14**(6), 1966.

[K57] Lee, C. Y., An algorithm for path connections and its applications. *Trans. Electronic Computers*, **EC-10**, 346-365, 1961.

[K58] Lemke, C. E., The dual method for solving the linear programming programs. *Naval Res. Logist. Quart.* **1**, 36–47, 1954.

[K59] Lemke, C. E., and Speilberg, K., Direct search algorithm for zero-one and mixed integer programming. *J. Oper. Res. Soc. Amer.* **15**(5), 1967.

[K60] Pierce, J. F., Application of combinatorial programming to a class of all zero-one integer programming problems. *Management Sci.*, **15**, 191–209, 1968.

[K61] Roth, R., Computer solutions to minimum cover problems. *J. Oper. Res. Soc. Amer.* **17**, 455–466, 1969.

[K62] Roy, B., and Benayoun, R., Programmes linéaires en variables bivalentes et continues sur un graphe (Program Poligami). *METRA* 6(4), 1967.

[K63] Salkin, H., An adaptive algorithm for integer binary programming. Masters Thesis, Dept. of Operations Research, Cornell Univ., Ithaca, New York, June 1968.

[K64] Salkin, H., and Spielberg, K., Adaptive binary programming. IBM New York Scientific Center, Tech. Rep. 320-2951, June 1968.

[K65] Shapiro, J. F., Dynamic programming algorithms for the integer programming problem. I: The integer programming problem viewed as a knapsack-type problem. *J. Oper. Res. Soc. Amer.* **16**, 1968.

[K66] Shapiro, J. F., Group theoretic algorithms for the integer programming problem. II: Extension to a general algorithm. *J. Oper. Res. Soc. Amer.* **16**(5), 1968.

[K67] Spielberg, K., An algorithm for the simple plant location with some side conditions. IBM New York Scientific Center, Tech. Rep. 2900, May 1967.

[K68] Spielberg, K., Ennumerative methods for integer and mixed integer programming. IBM New York Scientific Center, Tech. Rep. 320-2928, March 1968.

[K69] Spielberg, K., and Guignard, M., The state enumeration method for mixed zero-one programming. *Proc. Symp. Programmation Math.* (*7th*), La Haye, 1970.

[K70] Thiriez, H. M., Implicit enumeration applied to the crew scheduling problem. Dept. of Aeronautics, M.I.T., Cambridge, Massachusetts, 1968.

[K71] Thiriez, H. M., The set covering problem: a group theoretic approach. *Rev. Française Informat. Recherche Opérationnelle* **3**, 84–103, 1971.

[K72] Trubin, V. A., On a method of solution of integer linear programming problem of a special kind. *Soviet Math. Dokl.* **5**, 1544–1546, 1969.

[K73] White, W. W., On a group theoretic approach to linear integer programming. Univ. of California, Berkeley, California, ORC 66-27, September 1966.

3. Additional Bibliography

[K74] Donath, W. E., Statistical properties of the placement of a graph. *SIAM J. Appl. Math.* **16**, 1968.

[K75] Steinberg, L., The blackboard wiring problem of placement algorithm. *SIAM Rev.* **3**, 37–50, 1961.

[K76] Gilmore, P. C., Optimal and suboptimal algorithm for the quadratic assignment problem. *SIAM J. Appl. Math.* **10**, 305–313, 1962.

[K77] Gondran, M., and Fiorot, J. C., Résolutions des systèmes linéaires en nombres entiers. *Bull. Direction Études Recherches EDF, Sér. C*, no. 2, pp. 65–116, 1969.

[K78] Henry-Labordère, A., A partitioning primal–dual solution to very large assignments problems. IBM Corp., TR00-1704, February 1968.

INDEX

A
B 7
C 8
D 9
E 0
F 1
G 2
H 3
I 4
J 5