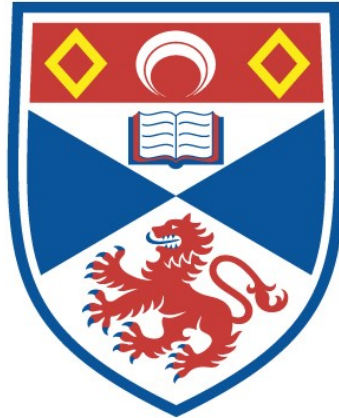


SOME CONTRIBUTIONS TO THE THEORY OF
MATHEMATICAL PROGRAMMING

Chandra P. Saksena

A Thesis Submitted for the Degree of PhD
at the
University of St Andrews



1970

Full metadata for this item is available in
St Andrews Research Repository
at:
<http://research-repository.st-andrews.ac.uk/>

Please use this identifier to cite or link to this item:
<http://hdl.handle.net/10023/13734>

This item is protected by original copyright

Tu 5723

THESIS for Ph.D.

MAY 1970

ProQuest Number: 10167347

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 10167347

Published by ProQuest LLC (2017). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

I hereby declare that this thesis has been composed by myself; that the work of which it is a record has been done by myself; and, that it has not been accepted in any previous application for any higher degree. This research concerning the Theory of Mathematical Programming was undertaken on 16th October 1967, the date of my admission as a research student under Ordinance General No. 12 for the degree of Doctor of Philosophy (Ph.D.)

5th May 1970.
Chandra P. Saksena



I hereby declare that the conditions of the Ordinance and Regulations for the degree of Doctor of Philosophy (Ph.D.) at the University of St. Andrews have been fulfilled by the candidate, Chandra P. Saksena.

Professor A.J. Cole

SOME CONTRIBUTIONS
TO THE THEORY OF
MATHEMATICAL PROGRAMMING

To my parents

and

Maya P. Saksena

Acknowledgments

The work contributed in this thesis on the Theory of Mathematical Programming could be brought to fruition mainly because of the scholarship offered by the University of St. Andrews since it enabled me to come from my distant country (India) to work at St. Andrews in October 1967. I acknowledge with gratitude the financial help thus offered to me by the University Authorities and in particular I am most grateful to Professor A.J. Cole for his recommendation to this effect.

I am grateful and indebted to Professor A.J. Cole also for his valuable guidance and the time and energy that he devoted in supervising my work. I am thankful to him also for the strenuous task of going through the manuscript of this thesis. I am also grateful to Professor O.J. Dahl of the University of Oslo with whom I had some helpful discussions at an early stage in this project.

I would also like to express thanks to my colleagues in the Computing Laboratory, St. Andrews University, for their help and co-operation throughout this work.

My appreciation and thanks are due to Miss Susan M. Watt for her hard work and the excellent typing of the manuscript. I should also like to thank Mrs Cecile Abramson for her valuable help in typing a part of the thesis.

Finally, I am most indebted to my wife for her help in checking the thesis against the manuscript and her cheerful acceptance of a part-time husband during the long period that I remained engaged on this research project.

CONTENTS

CHAPTER 1.	BACKGROUND, HISTORY AND GENERAL INTRODUCTION	1
	Background and history	1
	Optimisation problems	1
	Programming Problems	2
	History	3
	Computer Codes	7
	Introduction	7
CHAPTER 2.	THE GENERAL LINEAR PROGRAMMING PROBLEM	10
	Formulation of the general linear programming problem	10
	Formulation 1	10
	Formulation 2	11
	The Multiplex Method of Linear Programming	12
	The Multiplex flow diagram and the algorithm	19
	The flow diagram	19
	The algorithm	20
CHAPTER 3.	THE BOUNDING HYPERPLANE METHOD OF LINEAR PROGRAMMING - PART I	22
	The Inequalities	22
	Decision rules	27
	The pivotal transformations	31
	The unbounded and non-feasible problems	31

CHAPTER 3.

(Contd.)	Degeneracy and Cycling	33
	Geometrical Interpretation of the Method	36
	Estimation of the running time	45
	Illustrations	45

CHAPTER 4. THE BOUNDING HYPERPLANE METHOD OF LINEAR

PROGRAMMING - PART II 62

Equalities and their relationship to inequalities 62

The equality constraints 64

Techniques 65

The new tableau 68

The final equality tableau 69

Motivation in selecting one of the two techniques 70

The final inequality/functional tableau 72

Directional analysis 75

The unique solution 77

An infinite number of solutions 77

Estimation of running time 78

Illustration 79

CHAPTER 5. FLOW DIAGRAMS AND THE GENERAL ALGORITHM 82

Flow diagrams 82

The general algorithm 83

CHAPTER 6. APPLICATION OF THE B.H.P. TO THE SIMPLEX

METHOD OF LINEAR PROGRAMMING 85

The two phases 85

CHAPTER 6.

(Contd.)	The extended algorithm	86
	Phase 0	86
	New tableau and the evaluations	86
	Phases 1 and 2	88
	Degeneracy	88
	Illustrations	89

CHAPTER 7. COMPARATIVE STUDY OF THE SIMPLEX AND B.H.P.

METHODS	99
Number of iterations required by the B.H.P.M. vis-a-vis Simplex	99
Time taken by the two methods	99

CHAPTER 8. SUGGESTIONS FOR FURTHER RESEARCH

Extensions to quadratic programming	102
Accuracy in linear programming	107
Modifications to reinversion process	107
Modifications to the equality solution techniques	108

BIBLIOGRAPHY	111
--------------	-----

Chapter 1

BACKGROUND, HISTORY AND GENERAL INTRODUCTION

1.1 Background and history.

1.1.1 Optimisation problems. The class of problems that seek to optimise a function of a number of variables (functions) subject to certain constraints on the variables (functions) are, in general, called optimisation problems. These have long interested and intrigued mathematicians; Euclid in the 3rd Century B.C., for example, mentioned a number of optimisation problems in his Elements. Most of the optimisation problems, however, defied rigorous solution until the development of such mathematical disciplines as the calculus of variations, differential calculus and the like. These have subsequently been applied, with considerable success, to the solution of a wide range of classical optimisation problems arising mainly in the fields of engineering sciences, geometry and physical sciences. In the past two decades or so a new class of optimisation problems, termed "programming" problems, have been encountered - arising mainly in the fields of economic theory, business, military and industrial operations. These are usually not amenable to solution by the classical method of calculus.

Classical methods using lagrangian multiplier techniques¹ have however been developed (Klien 1955[50]) for resolving the non-negativity requirements of the variables in the problems with inequality constraints. These are found to be most impractical except for the solution of 'toy' problems (Charnes and Cooper, 1955 [11]).

1.1.2 Programming Problems: The general programming problem is concerned with maximising or minimising a given objective function subject to a set of restrictions; mathematically, it may be stated as under:

Find a vector $X = (x_1, x_2, \dots, x_n)$ which maximises (or minimises)

$$F = f(X) ,$$

subject to

$$g_i(X) \geq 0 \quad , \quad i = 1, 2, \dots, m_1 ,$$

$$h_k(X) = 0 \quad , \quad k = 1, 2, \dots, m_2 ,$$

and

$$X \geq 0 .$$

If the functions f , $\{g_i\}$ and $\{h_k\}$ are all linear the problem (1.1.1) is called the linear programming² problem. If any of these functions is non-linear, it is called the non-linear programming problem; in particular, if f is a quadratic function and $\{g_i\}$ and $\{h_k\}$ are linear, then it is a quadratic programming problem.

1. Developments to Lagrangian multiplier techniques have, however, been continued to be made to tackle both the linear and non-linear programming problems (Everett 1963[23], Falk, 1967 [25]).

2. The term 'linear programming' was suggested by T.C. Koopmans to G.B. Dantzig in 1951 as an alternative to the earlier form 'programming in a linear structure' (Dantzig 1948 [16]).

The linear programming problem may, further, be classified as the general structure or special structure programming problem depending upon whether the coefficient matrices in $\{q_i\}$ and $\{h_k\}$ have the former or the latter structure. This thesis is concerned with the general linear programming structure and quadratic³ programming problems.

1.1.3 History. The famous mathematician J.B.J. Fourier (1826 [28]) appears to be the first to have come across a linear programming problem while attempting to find the least maximum deviation fit to a system of linear equations⁴. Interestingly his suggestion for the solution of the above problem by vertex to vertex descent to the minimum is also the principle behind the Simplex Method of linear programming developed by G.B. Dantzig in 1947. Later the well known mathematician M.Ch.J. de la Vallee Poussin (1911[61]) also, while considering the above problem, suggests a similar solution.

Subsequently, the Russian mathematician L.V. Kantorovich (1939, 1942[47,48]) while considering the application of mathematics to production problems and in particular to transportation problems, seems to be the first to have recognised the well defined mathematical structure of production problems that were amenable to numerical solution. His joint paper with

3. The case when the function f has a positive semidefinite quadratic form.

4. Fourier reduced this problem to finding the lowest point of a polyhedral set.

M.K. Gavurin (1949^[49]) describes in detail the theory of the transshipment problem. The computational algorithm given in this paper is, however, incomplete. During this period Hitchcock 1941^[40] independently formulated and solved the transportation problem⁵ and later G. Stigler (1945^[67]) described the famous 'diet problem'⁶.

However, despite the recognition of linear programming problems in one form or the other, it was only in 1947 that the general linear programming problem was formulated in precise mathematical terms by G.B. Dantzig and others in the U.S. department of the Air Force which then constituted a group called project SCOOP (scientific computation of optimal programs). The most outstanding contribution of project SCOOP was the Simplex technique for the solution of the general linear programming problem. It was presented by G.B. Dantzig (1949 [17]) at the historic conference (June 20 to 24, 1949) held in Chicago by the Cowles Commission for Research in Economics and whose proceedings were brought out under the direction of T.C. Koopmans in 1951. The important results on duality based on unpublished notes of J. Von Neuman were also, for the first time, presented by Dantzig (1951^[63]) and Gale, Kuhn and Tucker (1951^[33]) at the same

5. T.C. Koopmans (1947^[34]) too independently later solved this problem.

6. The diet problem was, in fact, formulated and approximately proposed in 1941 by Jerome Cornfield in an unpublished memorandum and treated by George Stigler by what he calls an 'experimental procedure'.

conference.

Subsequently an important and, in fact, the first ~~symposium in Linear Inequalities and Programming~~ symposium in Linear Inequalities and Programming was held (June 14 to 16, 1951) in Washington under the joint auspices of the Air Force and the National Bureau of Standards. The proceedings of the symposium were published in 1952, under the direction of A. Orden and L. Goldstein[60]. The proceedings of these symposia are of significance in the history of linear programming as they constituted for a number of years the only general source of information and no doubt provided great stimulus to a number of individuals and research organisations for further research, developments and extensions in the field of linear programming.

Subsequently a number of symposia in linear programming (or covering the linear programming field) have been held and also a number of journals from different computing, operational research and allied fields sprang up all over the world. This has mainly been due to remarkable growth in the applications of linear programming to industrial problems and the simultaneous development of new techniques for the solution of linear programming problems. Of the latter, the Dual Simplex Method[56] developed by Lemke in 1954, may particularly be mentioned being concerned with those aspects of the general linear programming problem that, though often peculiar to the former, were not covered by the Simplex Method. As a result the Composite Simplex-Dual Simplex algorithm[18, 59] appeared in 1954. There are a number of other methods which can be used for solving the

linear programming problem. These include: Relaxation Method, Motzkin and Schoenberg [58] ; Projection method, Tompkins [75] the Double Gradient and Multiplex Methods, Frisch[31,30] ; the Hungarian Method, Kuhn [53] ; the Stepping Stone Method, Charnes and Cooper [10] and a refinement to this method⁷ called the MODI Method [26] ; the Ratio-Analysis Method⁸, Fergusson and Seargent [26] ; the Primal-Dual algorithm, Dantzig, Ford and Fulkerson [20] and Ford and Fulkerson [27] ; the Decomposition techniques⁹, Dantzig and Wolfe [21] , Beale [7] , Balas[83,84,85] Branch and Bounding Methods⁹, Land and Doig [55] etc. etc. However, as noted by Hoffman, Mannos, Sokolowsky and Weigmann (1953[42]) and Hoffmann (1955[41a]), most of these methods have not proved as effective in solving the general linear programming problem as the Simplex technique or some version of the Simplex technique - mainly because of the slowness in convergence and the very large number of iterations required. The Multiplex Method [30,86] though has been claimed by its author to be quite effective compared to the Simplex Method for medium sized problems^{9a}. The method has, however, not been extensively coded on computers.

7. and also the Transportation Problem procedure developed by A. Henderson and R. Schlaifer [39].

8. A similar method is described in Waugh & Burrows [80].

9. It may be noted that these techniques are mainly concerned with 'special structured' linear programming problems.

9a. where the number of non-basis variables is below 30 or 50.

1.1.3.1 Computer Codes. The general Simplex Method was first programmed for the SEAC Computer by Alex Orden of the U.S. Air Force and A.J. Hoffman of the National Bureau of Standards in January 1952. Later W. Orchard Hays of the Rand Corporation coded the Simplex Method for the IBM-CPC in 1952, for the IBM-701 in 1954 and the IBM-704 in 1956. Subsequently, with the tremendous growth in applications of linear programming to problems of business and industry, computer codes for the Simplex and other methods have been written on most of the intermediate and large general purpose electronic computers throughout the world.

1.2. Introduction. As stated earlier the Simplex Method (or its variations e.g. Dual Simplex Method) has thus far been the most effective and widely used general method for the solution of linear programming problems. The Simplex Method in its various forms starts initially with a basic feasible solution and continues its moves in different iterations within the feasible region till it finds the optimal solution. The only other notable variation of the Simplex Method, namely the Dual Simplex Method, on the other hand, by virtue of the special formulation of the linear programming problem, starts with an in-feasible solution and continues to move in the in-feasible region till it finds the optimal solution at which it enters the feasible region. In other respects both the Simplex and the Dual Simplex Methods follow essentially the same principle for obtaining the optimal solution. The rigorous mathematical features have been widely discussed in the literature [12, 19, 34, 35, 38, 68, 77] and only those formal aspects of this topic which are closely related

to the subject of this thesis will be outlined.

The Multiplex Method, though reported in the literature [30, 15, 69, 71, 29, 32], is not so well known and has also not been widely coded on electronic computers. It had earlier been programmed for the English Electric's Computer 'DEUCE' by the author [72] and the Ferranti's 'MERCURY' by Ole-John Dahl in 1960 [15]. Later both the above mentioned computers were obsolete and the efforts presently concentrate on coding it for UNIVAC 1100 and IBM 360. The Multiplex Method, as such, has been included in the present thesis and discussed in some detail in chapter 2. The flow diagram and the algorithm for the method is given in section 2.4, chapter 2.

The main body of the thesis consists of developing a new linear programming method which has been called the Bounding Hyperplane Method¹⁰ - Part I. This is explained in detail in chapter 3. The method could initially start with either a basic feasible or in-feasible point and in its subsequent moves it may either alternate between the feasible and the in-feasible regions or get restricted to either of them depending upon the problem. It is applicable as a new phase which we call phase 0 to the Simplex Method, particularly in situations where an initial basic feasible point is not available. In such cases it either results in a feasible point at the end of phase 0 or else yields a 'better' in-feasible point for phase 1 operations of the Simplex Method. Moreover, it is found that the number of

10. It is abbreviated henceforth as either B.H.P. Method or B.H.P.M.

iterations required to reach either the former by the application of phase 0 or the latter by the application of first phase 0 and then phase 1 are, in general, less than those required by following phase 1 alone. This is explained with illustrations in chapter 6. Even when applied alone the method, in general, yields the optimal solution in fewer iterations as compared with the Simplex Method. This is illustrated with examples in chapter 3.

We also develop and illustrate a powerful but straightforward method whereby we first find the solution to the equality constraints and (if the former does not yield an inconsistent solution point) then the transformations to the latter are obtained from the equality solution tableau corresponding to the former. This results in reducing the iteration time appreciably for each iteration of the method. It has been called the B.H.P.M. - part II and is discussed in chapter 4.

To estimate the time taken by the B.H.P. and the Simplex Method, the two codes (written in Fortran) have been run on a number of problems taken from the literature. The results have been summarised in chapter 7.

Finally, the suggestions for further research towards (i) the extensions of B.H.P.M. to the quadratic programming problem where the function in (1.1.1) is positive semi-definite, and (ii) the accuracy of computations in linear programming, in general, are discussed in sections 8.1 and 8.2 respectively of chapter 8.

THE GENERAL LINEAR PROGRAMMING PROBLEM

2. Formulation of the general linear programming¹¹ problem.

2.1 Formulation 1.

Maximise the optimal function

$$F = CX \quad , \quad (2.1.1)$$

subject to the conditions

$$AX \leq P_0 \quad , \quad (2.1.2)$$

and

$$X \geq 0 \quad ; \quad (2.1.3)$$

where

$$X = \begin{vmatrix} x_1 \\ x_2 \\ \vdots \\ x_p \end{vmatrix} \quad , \quad A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1p} \\ a_{21} & a_{22} & \dots & a_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mp} \end{vmatrix} \quad , \quad (2.1.4)$$

$$P_0 = \begin{vmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{vmatrix} \quad , \quad (2.1.5)$$

and

$$C = |c_1 \ c_2 \ \dots \ c_p| \quad . \quad (2.1.6)$$

¹¹Abbreviated as 'L.P. problem' in future discussions.

2.2 Formulation 2.

The above can be formulated also as follows:

Maximise the optimal function (2.1.1) subject to

$$Y = GX \quad , \quad (2.2.1)$$

where¹²

$$0 \leq Y \leq Q_0 \quad , \quad (2.2.2)$$

$$G = \begin{pmatrix} E_n \\ A \end{pmatrix} \quad , \quad (2.2.3)$$

$$\text{and } Q_0 = \begin{pmatrix} L \\ P_0 \end{pmatrix} \quad ; \quad (2.2.4)$$

E_n and L being respectively the identity submatrix ($n \times n$) and the column vector ($1 \times n$) consisting of some very large numbers corresponding to the n elements (variables) of the column vector X ($1 \times n$). Further Y is the column vector ($1 \times N$)

$$Y = \begin{pmatrix} Y^1 \\ Y^2 \end{pmatrix} \quad ; \quad (2.2.5)$$

where

$$Y^1 (=X) = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad , \quad (2.2.6)$$

$$Y^2 = \begin{pmatrix} y_{n+1} \\ y_{n+2} \\ \vdots \\ y_N \end{pmatrix} \quad ,$$

$$\text{and } N = n + m \quad . \quad (2.2.7)$$

¹²The vectors C and X comprise n elements (instead of p) in this formulation; also the matrix A is of the order of ($m \times n$) .

The latter formulation is used in the Multiplex Method [30] for finding the optimal solution to the general linear programming problem. We shall now discuss the Multiplex Method and return to the former formulation for describing the B.H.P. Method and the related work in subsequent chapters.

2.3 The Multiplex Method of Linear Programming.

Let the inequalities (2.2.2), in general, be considered as follows

$$\underline{Y} \leq Y \leq \bar{Y} \quad , \quad (2.2.8)$$

where \underline{Y} and \bar{Y} denote the lower and upper bounds of Y . Let I, J be the set of indices of the rows and columns of $G (= g_{ij}$, an $N \times n$ matrix). Also let $G_i, (i \in I)$ define the row space of G so that

$$G_i = | g_{i1} \quad g_{i2} \quad \dots \quad g_{in} | \quad , \quad i \in I \quad . \quad (2.2.9)$$

Further let $I_{ind} \subset I$, $I_{dep} \subset I$ be the set of indices of the rows of G corresponding to the matrices E_n and A respectively. I_{ind} and I_{dep} then define the 'independent' and 'dependent' sets (of the variables) Y^1 and Y^2 respectively.

Let $I_{opt} \subset I$ be the set of indices i for which the elements $y_i, i \in I_{opt}$ belong to the operation vector¹³ $\hat{Y} (= 1 \times v, v \leq n)$, that is

$$\hat{Y} = | \hat{y}_i | \quad , \quad i \in I_{opt} \quad . \quad (2.2.10 (i))$$

13. Frisch [29, 30] calls this vector the operation set. Initially

\hat{Y} is a null vector.

The vector \hat{Y} is such that

- (i) the elements (variables) \hat{y}_i , $i \in I_{\text{opt}}$ are at their either lower or upper bounds as the case may be and are linearly independent,
- and (ii) if $\hat{y}_i = \bar{y}_i$ then $\rho_i \leq 0$
 or else if $\hat{y}_i = \underline{y}_i$ then $\rho_i \geq 0$ } $i \in I_{\text{opt}}$. (2.2.10 (ii))
- where ρ_i , called the regression coefficients, are the element of the regression vector

$$R = |\rho_i|, \quad i \in I_{\text{opt}}, \quad (2.2.11)$$

given by

$$R = M^{-1} M_{\text{ok}} \quad (2.2.12)$$

The moment matrixⁿ M and the column vector M_{ok} are given by

$$M = (m_{ik}) \quad (2.2.13)$$

$$M_{\text{ok}} = G_k C^T \quad (2.2.14)$$

where

$$m_{ik} = G_i G_k^T, \quad i, k \in I_{\text{opt}}, \quad (2.2.15)$$

14. The moment matrix M is a symmetric matrix in view of (2.2.15) and so also is the inverse moment matrix M^{-1} .

and the superscript¹⁵ \top denotes the transposition to a column vector.

The condition (2.2.10 (ii)) above is termed the sign-correctness of the regression coefficients; if the opposite is the case then the latter are said to be sign-incorrect.

Additionally, in view of (2.2.10 (ii)) above the direction numbers of y_i , $i \in I_{\text{opn}}$ are set to zero, so as to retain them on their bounds. The direction numbers denoted by d_i , $i \in I$ are given by the^{row} vector

$$D = (D^1, D^2) = |d_i| \quad ; \quad (2.2.16)$$

where

$$D^1 = |d_i^1| \quad , \quad D^2 = |d_i^2| \quad , \quad (2.2.17)$$

and

$$d_i^1 \left[\begin{array}{l} = c_i + \sum_{k \in I_{\text{opn}}} g_{ki} \rho_k \\ = 0 \quad , \quad i \in (I_{\text{opn}} \cap I_{\text{ind}}) \end{array} \right] , i \in I_{\text{ind}} ; \quad (2.2.18)$$

$$d_i^2 \left[\begin{array}{l} = \sum_{j \in J} c_j g_{ij} + \sum_{j \in J} \sum_{k \in I_{\text{opn}}} g_{ij} \rho_k g_{kj} \\ = 0 \quad , \quad i \in (I_{\text{opn}} \cap I_{\text{dep}}) \end{array} \right] , i \in I_{\text{dep}} . \quad (2.2.19)$$

These direction numbers, in turn, are utilised in selecting

y_η , $\eta = i \in I$ for its entry in the operation vector by

15. The superscript \top would henceforth be used to denote the transposition of either a matrix or a row to column vector and vice-versa, as the case may be.

following the criterion below,

$$\lambda_{\min} = \frac{\Delta y_{\eta}}{d_{\eta}} = \text{Min}_{i \in I'} \left(\frac{\Delta y_i}{d_i} \right); \quad (2.2.20)$$

where

$$\Delta y_i \begin{cases} = \bar{y}_i - y_i^{\circ} & \text{for } d_i < 0 \\ = \underline{y}_i - y_i^{\circ} & \text{for } d_i > 0 \end{cases}, \quad (2.2.21)$$

y_i° are the elements of the vector

$$Y^{\circ} = | y_i^{\circ} | \quad (2.2.22)$$

the superscript '0' denoting the initial point which is either readily available¹⁶ from the problem itself (for example, in many situations it could be the lower bound values of the variables) or can be found and I' is defined by

$$I' \cup I_{\text{opt}} = I. \quad (2.2.23)$$

λ_{\min} and y_{η} are termed the breaking out parameter and the variable respectively; the latter is included in \hat{Y} . The new inverse moment matrix required for computing the regression vector R is then obtained from

$$M_{\text{new}}^{-1} = \left| \begin{array}{c|c} M' & Z \\ \hline Z^T & 1/\omega_{\eta\eta} \end{array} \right|; \quad (2.2.24)$$

16. As it happens, in most of the macro-economic problems, to which the Multiplex Method has been mainly applied, an initial point is generally available; however, if it is not available then it can be found using one of the methods discussed in [30].

where

$$M' = M_{old}^{-1} - Q Z^T, \quad (2.2.25)$$

and the column vectors Q , μ , Z and the scalar $\omega_{\eta\eta}$ are given by

$$Q = M_{old}^{-1} \mu; \quad \mu = |m_{\eta i}|, \quad i (\neq \eta) \in I_{ofn}, \quad (2.2.26)$$

$$Z = Q / \omega_{\eta\eta}, \quad \omega_{\eta\eta} = m_{\eta\eta} + Q^T \mu. \quad (2.2.27)$$

Since the regression coefficients ρ_i , $i \in I_{ofn}$ should remain sign correct, after y_η has also entered the operation vector \hat{Y} it is checked that the conditions (2.2.10(III)) are satisfied. If, therefore, one or more variables turn out to be sign-incorrect then the one encountered last as sign-incorrect is removed from \hat{Y} . As often happens, \hat{Y} may no longer consist of sign-incorrect variables; but if this is not the case then we continue to remove, one by one, sign-incorrect variables till \hat{Y} comprise only sign-correct variables. The elements of the inverse moment matrix are updated each time a sign-incorrect variable is removed from \hat{Y} and the new inverse moment matrix elements are obtained (each time) from,

$$M_{new}^{-1} = M_{old}^{-1} - S \frac{S^T}{\xi_{\tau\tau}}, \quad (2.2.28)$$

where τ is the index of the variable y_τ turning out to be sign-incorrect, $\xi_{\tau\tau}$ is the pivotal element of the old inverse moment matrix and S ($= | \xi_{i\tau} | = | \xi_{\tau j} |$; $i, j \in I_{ofn}$) is the τ^{th} column (or row) vector of the old inverse moment matrix. [In computing M_{new}^{-1} from (2.2.28) the elements of those vectors which correspond to the τ^{th} row and column turn out to be null

vectors and hence care is taken to squeeze the matrix correspondingly by one dimension each time a sign-incorrect variable is removed from \hat{Y}] .

The new point at the end of each iteration is given by

$$Y = Y^o + \lambda_{min} \cdot D^T \quad (2.2.29)$$

and this, in turn, becomes the starting point for subsequent iterations.

If, in the course of different iterations, it is found that

$$((\forall i \in I') (d_i = 0)) \quad (2.2.30)$$

then it is implied that the optimal point has been obtained. The general optimal criterion (which would also be satisfied if (2.2.30) is true), however, is that the preference direction number

$$d_o < \epsilon \quad (2.2.31)$$

where

$$d_o = D^1 C^T \quad (2.2.32)$$

and ϵ is the threshold value designed to take into consideration the round off errors as well as errors in the data. The optimal value when (2.2.31) is true is given by

$$F = C Y^1 (= C X) . \quad (2.2.33)$$

At this stage, if

$$v = \eta \quad (\text{refer 2.2.10(i)}) \quad , \quad (2.2.34)$$

then the optimum point obtained is unique. If, on the other hand,

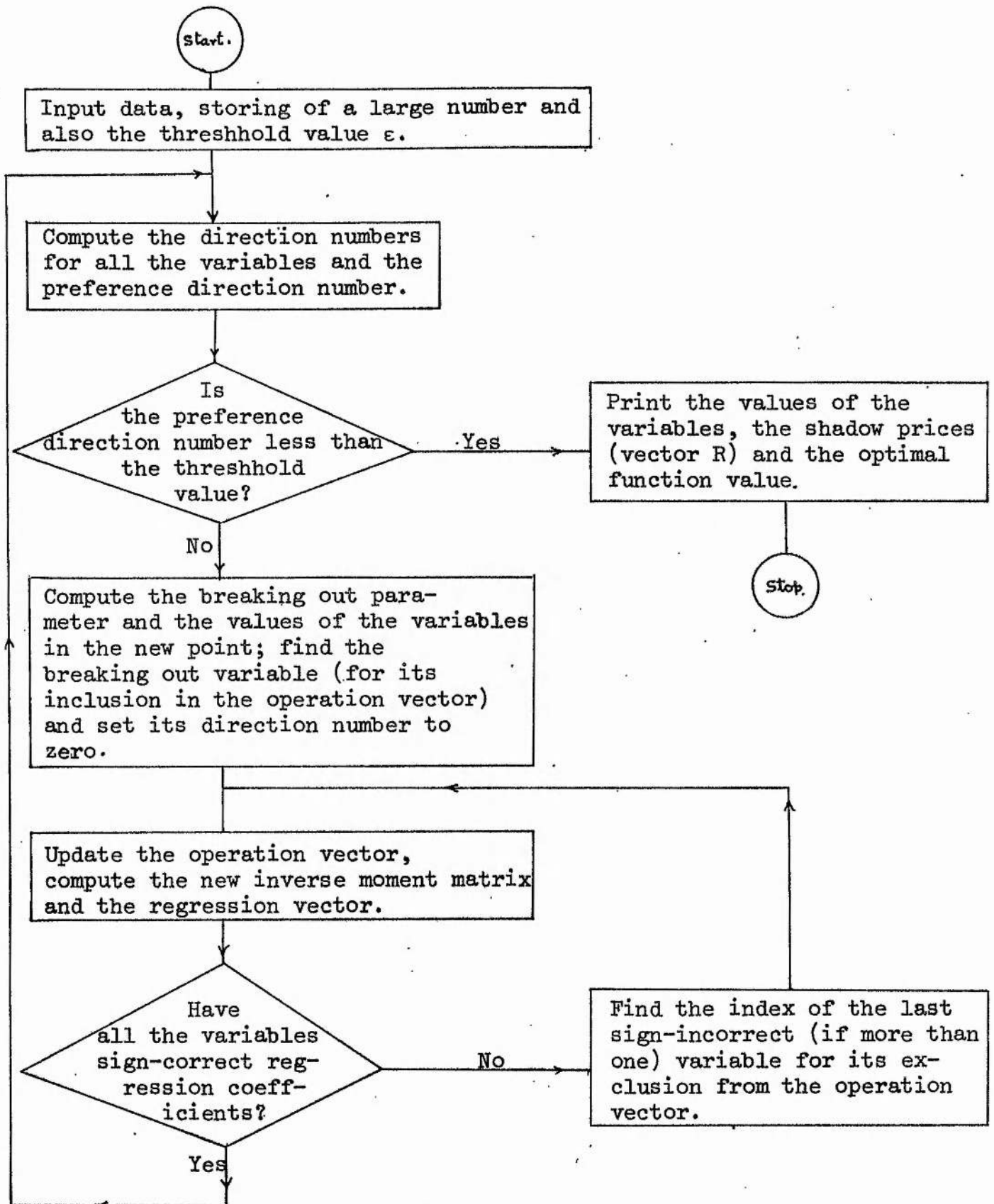
$$v < \eta \quad , \quad (2.2.35)$$

it implies that there exists at least one set of $n-v$ linearly

independent variables which generate a linear manifold such that the value of the optimal function is the same at every point of this manifold.

2.4. The Multiplex flow diagram and the algorithm.

2.4.1. The flow diagram .



2.4.2 The algorithm. For the purpose of describing the algorithm of this section let M denote the section 2.4.2. The algorithm is iterative and detailed as follows.

- M. Find D and d_0 from (2.2.16) to ^{(2.2.19) and} (2.2.32) respectively.
- M.1 Test d_0 :
- M.1.1 if $d_0 < \epsilon$, then print the values of all the variable the shadow prices and the optimal function value given by (2.2.5) ^(2.2.11), (2.2.33) respectively.
- M.1.2 if not, then
- M.1.2.1 (i) compute λ_{min} , η from (2.2.20), Y from (2.2.29) and set $d_\eta = 0$,
- M.1.2.2 (ii) compute the new inverse moment matrix from (2.2.24) or (2.2.28) - as the case may be, the regression vector R from (2.2.11) and update the operation vector.
- M.2 Test ρ_i , $i \in I_{opn}$:
- M.2.1 if $\rho_i \geq 0$ test \hat{y}_i and count i in q_1 :
- M.2.1.1 if $\hat{y}_i = \underline{y}_i$, ρ_i is sign-correct, go to M.3;
- M.2.1.2 otherwise ρ_i is sign-incorrect, go to M.2.2.3;
- M.2.2 if not, test y_i and count i in q_2 :
- M.2.2.1 if $\hat{y}_i = \bar{y}_i$, ρ_i is sign correct, go to M.3;

- M.2.2.2 otherwise ρ_i is sign-incorrect;
- M.2.2.3 $i = \tau$, go[†] to M.1.2.2;
- M.3 test $(q_1 + q_2)$:
- M.3.1 if $(q_1 + q_2) < \epsilon$ go to M.2;
- M.3.2 otherwise go to M.

†. τ is the last sign-incorrect variable encountered in \hat{Y} .

THE BOUNDING HYPERPLANE METHOD OF LINEAR PROGRAMMING - PART I

Let the general l.p. programming be considered as formulated in section 2.1 of chapter 2. We consider the inequality constraints of the system (2.1.1)-(2.1.3) alone in this chapter; the equality constraints can be treated independently of the former and hence are treated separately as part II of the B.H.P. Method in the next chapter.

3.1 The Inequalities. Let the inequality constraints in (2.1.2) be converted to equations by (i) multiplying the constraints of ' \geq ' type (I), if any, by -1 so as to convert them first to constraints of ' \leq ' type (II) and then (ii) adding a positive slack variable to each of the type (II) constraints. The resulting system of equations and the l.p. problem could then be rewritten as

Maximise

$$F = C X , \quad (3.1.1)$$

subject to

$$B X = P_0 , \quad (3.1.2)$$

and

$$X \geq 0 ; \quad (3.1.3)$$

where

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} , \quad (3.1.4)$$

$$B = |P_1, P_2, \dots, P_n| = \begin{vmatrix} b_{11} & b_{12} & \dots & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & \dots & b_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \vdots & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ b_{m1} & b_{m2} & \dots & \dots & b_{mn} \end{vmatrix}, \quad (3.1.5)$$

$$C = |c_1, c_2, \dots, c_p, c_{p+1}, c_{p+2}, \dots, c_n|, \quad c_j \begin{cases} = c_j & \text{for } j = 1, 2, \dots, p \\ = 0 & \text{for } j = p+1, p+2, \dots, n. \end{cases} \quad (3.1.6)$$

P_0 is given by (2.1.5) with signs corresponding to type II inequalities, if any, changed, 0 is a null vector and

$$n = p + m. \quad 3.1.6 \text{ (i)}$$

If we consider the slack variables in the set of equations (3.1.2) we note that they, being a set of m linearly independent vectors corresponding to m structural constraints in the original set (2.1.2), constitute for our study an initial basis. Let $P_{p+1}, P_{p+2}, \dots, P_n$ denote this set of m linearly independent vectors so that the initial basis is given by

$$\beta = |P_{p+1}, P_{p+2}, \dots, P_n| \quad (3.1.7)$$

whence we have,

$$\bar{X} = \beta^{-1} P_0 \quad \text{where} \quad \bar{X} = \begin{vmatrix} x_{p+1} \\ x_{p+2} \\ \vdots \\ x_{p+m} \end{vmatrix}; \quad (3.1.8)$$

$$\begin{aligned} \bar{B} &= |\bar{B}_1, \bar{B}_2, \dots, \bar{B}_n| \\ &= |\beta^{-1} P_1, \beta^{-1} P_2, \dots, \beta^{-1} P_n| \\ &= \beta^{-1} |P_1, P_2, \dots, P_n| \\ &= \beta^{-1} B; \end{aligned} \quad (3.1.9)$$

$$\begin{aligned} F &= C X \\ &= \bar{C} \bar{X} \\ &= \bar{C} \beta^{-1} P_0 \quad \text{where} \quad \bar{C} = |c_{p+1}, c_{p+2}, \dots, c_n|; \end{aligned} \quad (3.1.10)$$

$$\begin{aligned} Z &= \begin{vmatrix} Z_1 & Z_2 & \dots & Z_n \end{vmatrix} \quad (3.1.11(i)) \\ &= \begin{vmatrix} z_1 - c_1 & z_2 - c_2 & \dots & z_n - c_n \end{vmatrix} \\ &= \begin{vmatrix} \bar{C} \bar{B}_1 - c_1 & \bar{C} \bar{B}_2 - c_2 & \dots & \bar{C} \bar{B}_n - c_n \end{vmatrix} \\ &= \begin{vmatrix} \bar{C} \beta^{-1} P_1 - c_1 & \bar{C} \beta^{-1} P_2 - c_2 & \dots & \bar{C} \beta^{-1} P_n - c_n \end{vmatrix} \\ &= \bar{C} \beta^{-1} |P_1, P_2, \dots, P_n| - |c_1, c_2, \dots, c_n| \\ &= \bar{C} \beta^{-1} B - C. \end{aligned} \quad (3.1.11(ii))$$

18. In particular the initial solution \bar{X} given by (3.1.8) may thus be either an infeasible or a feasible solution. It need not necessarily be the latter as is prerequisite with some other methods, for example the Simplex Method. '-' sign associated with $\bar{X}, \bar{B}, \bar{C}$ corresponds to the current point or tableau.

Let the general B.H.P. (method) tableau¹⁹ be given by

$$M = \begin{array}{c|c} \hline C & O \\ \hline \bar{B} & \bar{X} \\ \hline Z & F \\ \hline \end{array} \quad (3.1.12)$$

where the contents of the tableau are defined by (3.1.8) to (3.1.11). Let I, J be the set of indices of the rows and

columns of \bar{B} . Let $I_1 \subset I$ denote the set of indices i for which

$$x_i < 0, \quad i \in I_1 \text{ (if any)}. \quad (3.1.13)$$

Let $J_1 \subset J$ be the set of indices j for which

$$Z_j < 0, \quad j \in J_1. \quad (3.1.14)$$

and let \mathcal{A} and $\text{mod } Z_j, j \in J_1$ respectively denote the space spanned by $P_j, j \in J_1$ and the hyperplane with all $Z_j (j \in J_1)$ coefficients positive. Also let $\bar{B}_i (i \in I)$ define the row space of \bar{B} so that,

$$\bar{B}_i = |\bar{b}_{i1} \bar{b}_{i2} \dots \bar{b}_{in}|, \quad i \in I \quad (3.1.15)$$

and let $\bar{B}_i^j (j \in J_1, i \in I)$ denote the corresponding row elements of $\bar{B}_i (i \in I)$ for which (3.1.14) holds. Further let $I_2 \subset I$ be the set of indices i for which

$$\delta_i < 0, \quad (3.1.16)$$

where²⁰

$$\delta_i = \sum_j \bar{B}_i^j Z_j, \quad j \in J_1. \quad (3.1.17)$$

19. The general tableau M is different from the one considered in the Simplex Method since the negative elements in P_0 are retained under \bar{X} .

20. " δ_i " is proportional to the cosine of the angle between the normals to the $\text{mod } Z_j$ hyperplane (in the direction away from the origin) and the i^{th} hyperplane (in the infeasible direction). Geometrically, by considering (3.1.16) we therefore look for those hyperplanes that have this angle less than 90 degrees; hence, the name 'bounding' for such hyperplanes.

I_2 then defines, what from now on will be termed as, the set of the bounding hyperplanes. We next define a bounding vector Δ comprising of elements Δ_i ($i \in I_2$) such that,

$$\Delta_i = \left| \frac{x_i}{\delta_i} \right|, i \in I_2. \quad (3.1.18)$$

Geometrically Δ_i may be interpreted as a scalar which is proportional to d_i , that is

$$\Delta_i = k d_i, i \in I_2, \quad (3.1.19)$$

where d_i denotes the distance from the origin of the i^{th} bounding hyperplane in the increasing direction of the normal to the mod Z_j ($j \in J_1$) plane and k is a constant (in so far as the current solution point is concerned), given by

$$k = \left(\sum_j Z_j^2 \right)^{-\frac{1}{2}}, j \in J_1. \quad (3.1.20)$$

The bounding vector Δ thus is a function of the distance of the bounding hyperplanes $i \in I_2$, and the particular hyperplane η , for which

$$\Delta_\eta = \min_i (\Delta_i), i \in I_2 \quad (3.1.21)$$

defines, therefore, the nearest bounding hyperplane in the increasing direction of the normal to the mod Z_j ($j \in J_1$) plane. Clearly the optimum point, if $x_\eta \geq 0$, must either lie

on or below the hyperplane η given by (3.1.21), that is, in that half space of the η hyperplane which is towards the origin; similarly, if $x_\eta < 0$, then the optimum point must either lie on or above the hyperplane η , that is, in that half space of the η hyperplane which is away from the origin.

Utilising the above properties we select in each iteration the η hyperplane out of the set of the bounding hyperplanes as observed from the basis point (that is the new origin) reached at the last iteration until we arrive at the optimal solution to the problem by following the decision rules enumerated below.

3.2 Decision rules; class R and S.

Class R. If

$$((\exists j \in J) (z_j < 0)) \quad (3.2.1)$$

then the rules are defined as follows:

Rule R1. Select $i (= \eta) \in I_2$ from (3.1.21) above. This may be termed the Exit Criterion for the Class R.

Rule R2. If F^* denotes the value of the objective function at the new point then, for $i (= \eta) \in I_2$ and $j \in J_1$,

$$F^* = F - x_i (z_j / \bar{B}_i^j) \quad (3.2.2)$$

or
$$\Delta F = F^* - F = -x_i (z_j / \bar{B}_i^j); \quad (3.2.3)$$

where the particular $j \in J_1$ is yet to be specified and ΔF measures the change in the optimal function value from one to the next move. Since we consider infeasible solutions as well in our search for the optimum, the following two possibilities exist -

$$\text{Either (i) } x_{\eta} \geq 0 \quad (3.2.4)$$

$$\text{or (ii) } x_{\eta} < 0 \quad (3.2.5)$$

R2.1. If (3.2.4) is true, select $\Upsilon = j \in J_1$ from

$$\max_j \left| Z_j / \bar{B}_i^j \right|, \quad \bar{B}_i^j > 0, \quad j \in J_1, \quad i = \eta \in I_2, \quad (3.2.6)$$

which gives the maximum increase to the optimal function value.

(Alternatively $\Upsilon = j \in J_1$ could also be selected from

$$\text{either (i) } \min_j \left| Z_j / \bar{B}_i^j \right| \quad \text{or (ii) } \max_j (Z_j \bar{B}_i^j),$$

where in both cases $\bar{B}_i^j > 0, j \in J_1, i = \eta \in I_2$).

R2.2. If (3.2.5) is true then select $\Upsilon = j \in J_1$, for which

$$\max_j (Z_j / \bar{B}_i^j), \quad \bar{B}_i^j < 0, \quad j \in J_1, \quad i = \eta \in I_2; \quad (3.2.7)$$

however, if it happens that

$$((\forall j \in J_1) (\bar{B}_i^j \geq 0)) \quad (3.2.8)$$

(so that Υ satisfying (3.2.7) can not be found) then select

Υ from

$$\min_j \left| Z_j / \bar{B}_i^j \right|, \quad \bar{B}_i^j > 0, \quad j \in J_1, \quad i = \eta \in I_2. \quad (3.2.9)$$

If a tie occurs in the selection of Υ then we choose any of the j for which this tie occurs. This is called the Entry Criterion for class R.

In the former case the value of the optimal function increases as compared with its value in the last iteration but if the

latter were to be true then the value of the optimal function decreases in the next iteration. (This might appear to lead to a situation where an old basis may repeat itself. However, by deciding to move in each iteration on the (nearest) bounding hyperplane alone in the \mathcal{A} space, we believe it is highly unlikely that such a situation can arise. This has been confirmed by the empirical evidence of running the computer program (for B.H.P. method) on a number of both the small and large examples and no case has thus far been encountered where the algorithm does not terminate.)

The above decision rules thus uniquely determine the hyperplane η to which to move and the axis τ along which to move to the former. In particular $\bar{b}_{\eta\tau}$ determines the pivotal element on which the Gaussian eliminational transformations, according to formulae given in section 3.2.1, are performed to obtain the new point and the tableau.

The foregoing rules will therefore either lead to the optimum point wherein

$$((\forall j \in J)(z_j \geq 0) \ \& \ (\forall i \in I)(x_i \geq 0)) \quad (3.2.10)$$

or else a point will be reached which is considered in class S below.

Class S. If (refer also the Dual Simplex Method [56])

$$((\exists i \in I)(x_i < 0) \ \& \ (\forall j \in J)(z_j \geq 0)) \quad (3.2.11)$$

then the rules are defined as follows.

Rule S1. Select $i(-\eta') \in I_1$ for which

$$\Delta'_{\eta'} = \max_i (\Delta'_i) , i \in I_1 , \quad (3.2.12)$$

where

$$\Delta'_i = \left| \frac{x_i}{\delta'_i} \right| , \delta'_i \neq 0 , i \in I_1 , (3.2.12 (i))$$

$$\delta'_i = \sum_j \bar{b}_{ij} z_j , j \in J , i \in I_1 , (3.2.12 (ii))$$

and η' denotes that hyperplane amongst the infeasible hyperplanes which is farthest from the origin in the decreasing direction this time of the normal to the mod $Z_j (j \in J)$ hyperplane for the optimum point must now either lie on or above the η' hyperplane given by (3.2.12). If, however, it is found that

$$((\forall i \in I_1) (\delta_i = 0)) \quad (3.2.13)$$

then we select the η' hyperplane from

$$\max_i \left| x_i / \sqrt{\sum_j \bar{b}_{ij}^2} \right| , j \in J , i \in I_1 \quad (3.2.14)$$

The geometrical interpretation of Δ'_i , δ'_i and η' are analagous to those of Δ_i , δ_i and η discussed earlier with the above mentioned restrictions that are peculiar to the situation in class S. This is called the Exit Criterion for the class S.

Rule S2. Select $\gamma = j \in J_2$ for which

$$\max_j (z_j / \bar{b}_{ij}) , j \in J_2 , \quad i = \eta' (\in I_1) , (3.2.15)$$

where J_2 denotes the set of indices $j \in J$ for which

$$\bar{b}_{ij} < 0, j \in J, i = \eta (\in I_1) \quad (3.2.16)$$

This is called the Entry Criterion for the class S, and is the same as in the Dual Simplex Method [56]. (A tie, if any, is handled as in the case R2.2).

The optimal function value in class S decreases in each move and must in a finite number of steps yield the optimal solution, if it exists, satisfying (3.2.10).

3.2.1 The pivotal transformations. The pivotal transformations, for obtaining the new point and the tableau for either of the above two classes are given by ^{20(a)}

$$\left. \begin{aligned} \bar{b}'_{ij} &= \bar{b}_{ij} - (\bar{b}_{\eta j} / \bar{b}_{\eta \tau}) \bar{b}_{i\tau}, i \neq \eta \\ \bar{b}'_{\eta j} &= \bar{b}_{\eta j} / \bar{b}_{\eta \tau}, i = \eta \end{aligned} \right\}, j \in J+1, \quad (3.2.17)$$

and

where a prime denotes the new values of the elements in the transformed tableau. (The new values of the objective function, the shadow prices and the solution vector are respectively given by,

$$\left. \begin{aligned} \bar{b}'_{i_0} &= \bar{b}'_{m+1, n+1} = F' \\ \bar{b}'_{ij} &= \bar{b}'_{m+1, j} = Z' (= Z'_j), j \in J \\ \bar{b}'_{i_1} &= \bar{b}'_{i, n+1} = \bar{x}' (= \bar{x}'_i), i \in I \end{aligned} \right\} \quad (3.2.18)$$

3.3 The unbounded and non-feasible problems.

The condition

$$\left((\forall i \in I) \left((\forall j \in J_1) \left(\sum_j \bar{B}_i^j |Z_j| \leq 0 \right) \right) \right) \quad (3.3.1)$$

^{20(a)}. $\eta = \eta'$ in (3.2.17), if the pivotal row (hyperplane) is selected following class S decision rules.

which could also be written as

$$\left(\left(\forall i \in I \right) \left(\left(\forall j \in J_1 \right) \left(\sum_j \bar{B}_i^j Z_j \geq 0 \right) \right) \right), \quad (3.3.1 \text{ (i)})$$

or $\left(\left(\forall i \in I \right) \left(\delta_i \geq 0 \right) \right)$, δ_i given by (3.1.17), (3.3.1 (ii))

implies the unbounded problem; and the condition

$$\left(\left(\exists i \in I \right) \left(x_i < 0 \right) \& \left(\forall j \in J \right) \left(\bar{b}_{ij} \geq 0 \right) \right) \quad (3.3.2)$$

implies the non-feasible problem.

In either case there is no optimal solution to the problem.

The Unbounded Problem. Consider the space \mathcal{A} spanned by P_j , $j \in J_1$ for which (3.1.14) holds. Then if θ_i , $i \in I$ denotes the angle between the normals in \mathcal{A} to the mod Z_j hyperplane (in the direction towards the origin) and the i^{th} hyperplane (in the feasible direction) we have

$$\theta_i = \cos^{-1} \left(q_i \sum_j \left(\bar{B}_i^j |Z_j| \right) \right), i \in I, j \in J_1, \quad (3.3.3)$$

where q_i is a positive number given by

$$q_i = k \left(\sum_j \left(\bar{B}_i^j \right)^2 \right)^{-1/2}, j \in J_1, i \in I. \quad (3.3.4)$$

If (3.3.1) or (3.3.1(ii)) holds for each i then (3.3.3) indicates the non-existence of any bounding hyperplanes (or in other words it indicates the existence of an open convex solution set). The value of the optimal function can thus be increased infinitely in the increasing direction of the optimal function and no finite (maximal) optimal solution to the problems exists.

The non-feasible problem. (3.3.2) implies that (2.1.3) can not be satisfied and hence no optimal solution to the problem exists.

3.4 Degeneracy and Cycling.

Let us reconsider the decision rules for the classes R and S. In following the exit criteria in either of the two classes, if the η hyperplane happened to be the one with $\bar{x}_\eta = 0$ then the new solution would be degenerate (a tie in computing $\max_{i \in I_2} (\Delta_i)$ or $\min_{i \in I_2} (\Delta_i)$ from (3.1.21), even if the current solution is non-degenerate, may lead to degeneracy in the new solution). It must be significantly noted at this stage that those basic variables which have values equal to zero but correspond to non-bounding hyperplanes are, in any case, excluded from further consideration as we select only bounding hyperplanes in sub-space A . In effect, therefore, we are left with only such cases for which

$$\bar{x}_i = 0, \quad i \in I_2^d \quad (3.4.1)$$

where $I_2^d \subset I_2$ denotes the set of indices i for which

$$\left. \begin{array}{l} \text{either } \bar{x}_i = 0, \text{ in degenerate cases,} \\ \text{or } \Delta_i = \text{constant}, \quad i \in I_2 \text{ in non-degenerate cases,} \end{array} \right\} (3.4.2)$$

In such a situation the η hyperplane may, in general, correspond to any one of the \bar{x}_i for which (3.4.2) holds. We would, however, proceed as follows.

Since $\Delta_i = 0, \quad i \in I_2^d$, (that is, all such hyperplanes are equidistant from the origin in the increasing direction of the

normal to the mod Z_j ($j \in J_1$) plane), we determine the most convergent hyperplane from

$$\theta_\eta = \min_i (\theta_i), \quad i \in I_2^d. \quad (3.4.3)$$

If there is still a tie in computing the η hyperplane uniquely from (3.4.3) then we determine it from

$$\theta'_\eta = \min_i (\theta'_i), \quad i \in I_2^d; \quad (3.4.4)$$

where

$$\theta'_i = \cos^{-1} \left(g'_i \sum_{j \in J} (\bar{b}_{ij} |z_j|) \right), \quad (3.4.5)$$

and

$$g'_i = \left(\sum_j z_j^2 \right)^{-\frac{1}{2}} \left(\sum_j \bar{b}_{ij}^2 \right)^{\frac{1}{2}}, \quad j \in J.$$

It is, indeed, highly improbable that a tie could still persist in selecting the η hyperplane from either (3.4.3) or (3.4.4) because if for no other reason, rounding errors are introduced during the computations. For this reason and consequent saving

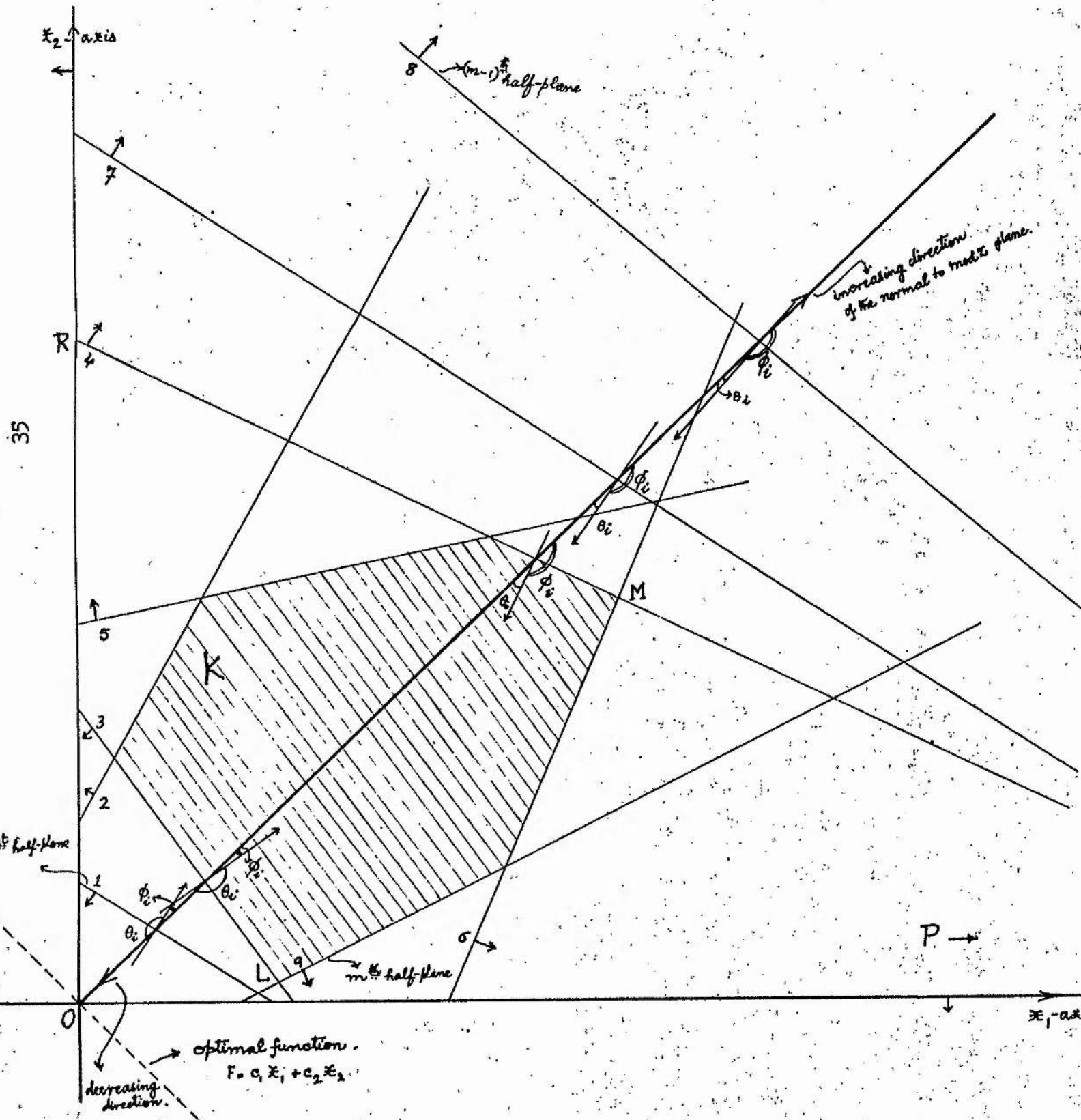


Figure 3.1

in computing time we find that a simpler criterion for selecting the η hyperplane from

$$\delta_\eta = \max_i |\delta_i|, i \in I_2^d, \quad (3.4.6)$$

is efficient and expedient in resolving both the degenerate and/or non-degenerate ties. As we shall show in illustrations given in section 3.7, we effectively resolve the cycling phenomenon in an artificially constructed example (for the Simplex Method) using the Criterion (3.4.6). It may also be stated here that the same criterion has been used for resolving the cycling phenomenon in another example due to Hoffman [41]. The latter is, however, not reported in the illustrations of section 3.7.

3.5 Geometrical Interpretation of the Method.

Let us consider the following two dimensional l.p. problem.

Maximise

$$F = c_1 x_1 + c_2 x_2, \quad (3.5.1)$$

subject to

$$a_{i1} x_1 + a_{i2} x_2 \leq b_i, i=1,2,\dots,m, \quad (3.5.2)$$

and

$$x_1, x_2 \geq 0. \quad (3.5.3)$$

where $b_i, i=1,2,\dots,m$ are unrestricted in sign. We presume without loss of generality that $c_1, c_2 > 0$. Let the $(m+2)$ constraints (3.5.2) and (3.5.3) be as shown in the graph of Fig.3.1. The shaded region K formed by them defines the feasible region as against the infeasible region surrounding K . Let the optimal function

be as shown by the dotted line in the diagram. The optimum point is obtained by moving the optimal function parallel to itself until it reaches the point $M(L)$ where it attains the maximum (minimum) value.

As we see from the diagram in figure 31, certain half-planes converge towards the maximal optimal point M and certain others similarly converge towards the minimal optimal point L . (It may be noted that the $(m-1)^{th}$ and 1^{st} half-planes are 'completely' redundant ones and although they appear to converge towards M and L respectively; they, in fact, are the least helpful in leading towards the optimum point). Since as in a maximising case we are interested in reaching the point M it will obviously be advantageous to search for the former so that by moving in different iterations on one or more of these half-plane we ultimately converge at the point M . To distinguish the two types of the half-planes, we consider the angles (denoted by θ_i , $i = 1, 2, \dots, m$) between the normal in the increasing direction^{20(b)} of the functional $Z = -c_1 x_1 - c_2 x_2$ and the normals to the half-planes given by (3.5.2) in the feasible region. The bounding half-planes that converge towards M are those for which

$$0^\circ \leq \theta_i \leq 90^\circ \quad (3.5.4)$$

^{20(b)}This is same as the decreasing direction of the mod Z_j plane (that is towards the origin).

Similarly the non-bounding half-planes that converge towards L are the ones for which

$$90^\circ \leq \theta_i \leq 180^\circ \quad (3.5.5)$$

From analytical geometry we know that the cosine of the angle (as defined above) between the i^{th} half-plane $a_{i1}x_1 + a_{i2}x_2 \leq b_i$ ($i = 1, 2, \dots, m$) and the functional $Z = -c_1x_1 - c_2x_2$ is given by

$$\cos \theta_i = \frac{-a_{i1}c_1 - a_{i2}c_2}{\sqrt{a_{i1}^2 + a_{i2}^2} \sqrt{c_1^2 + c_2^2}} \quad (3.5.6)$$

$$= k_i (-a_{i1}c_1 - a_{i2}c_2) \quad (3.5.7)$$

where k_i is a positive number given by

$$k_i = (a_{i1}^2 + a_{i2}^2)^{-1/2} (c_1^2 + c_2^2)^{-1/2} \quad (3.5.8)$$

In our search for the optimum we, however, consider (refer *discussions following* (3.1.19)) the increasing direction of the normal to the mod $Z_j (= c_1x_1 + c_2x_2)$ plane instead so that if ^{we} may define ϕ_i to be the angle between the former and the normal to the i^{th} half-plane in the feasible direction then we observe that

ϕ_i is related to θ_i by

$$\phi_i + \theta_i = 180^\circ \quad (3.5.9)$$

whence, in view of (3.5.4), (3.5.5) and (3.5.7), the type of the i^{th} half-plane as bounding or not is determined from

$$-a_{i1} c_1 - a_{i2} c_2 < 0 \quad (3.5.10)$$

or

$$-a_{i1} c_1 - a_{i2} c_2 \geq 0 \quad (3.5.11)$$

respectively. Computationally, therefore, we determine the bounding nature of the i^{th} hyperplane from the negativity of the inner product given by the left hand side term of (3.5.10). We next wish to determine the specific half-plane out of all the bounding half-planes in the system which leads us towards the optimum point M such that the 'completely' redundant ($(m-1)^{\text{th}}$) half-plane is avoided too. Let us at this stage, for simplicity of exposition consider just $m = 9$ half-planes with optimal function as shown by the dotted line. To make the figure (problem) representative for all the types of half-planes we consider the nine half-planes to comprise four non-bounding and five bounding half-planes. These are numbered 1, 2, 3, and 4, 5, 6, 7, 8 respectively in fig. 3.1. Furthermore, of the former number 1 is a redundant non-bounding half-plane and of the latter, numbers 7 and 8 are redundant bounding half-planes. (The redundant half-planes 1, 7 and 8, as may be seen from figure 3.1, are such

that they play no part in defining the feasible region K). The optimal point M is defined by the intersection of two bounding half-plane numbers 4 and 6. If we now look at the increasing direction of the normal to the mod Z plane (indicated by the thick line with an arrow) in figure 31 then we find that the half-plane number 4 happens to be one of these two half-planes; in fact, it is also the nearest (out of all the bounding half-planes) one from the origin O . Thus if we know the distances of all the bounding half-planes in the increasing direction of the normal to the mod Z plane then by selecting the nearest (from the origin) half-plane we are able to determine a plane which meets our requirements. In the present example it, in fact, passes through the optimum point M also. For this purpose, we need, however, not determine the exact distances of all the half-planes. Let us examine the 'static' d_i , $i = 1, 2, \dots, m$ given by

$$d_i = b_i / (-a_{i1} c_1 - a_{i2} c_2) \quad (3.5.12)$$

$$= b_i / \sqrt{a_{i1}^2 + a_{i2}^2} \sqrt{c_1^2 + c_2^2} \cos \theta_i$$

$$= (c_1^2 + c_2^2)^{-1/2} (b_i / \cos(\theta_i))$$

$$= k d_i \quad (3.5.13)$$

where k is a constant given by

$$k = (c_1^2 + c_2^2)^{-1/2} \quad (3.5.14)$$

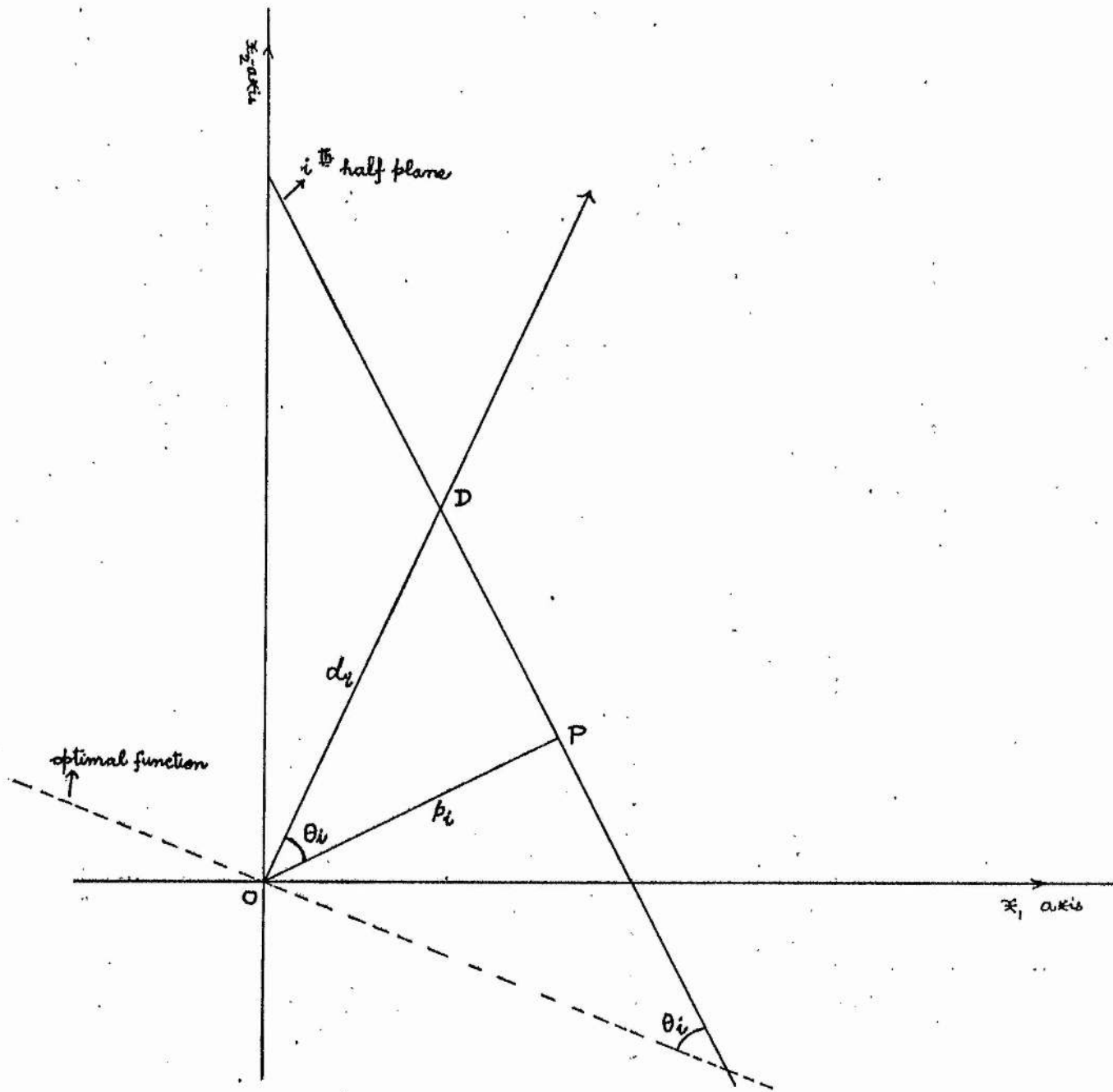


Figure 3.2

and p_i and d_i respectively are (refer figure 3.2) the perpendicular distance and the distance in the increasing direction of the normal to the mod Z_i half-plane of the i^{th} half-plane from the origin. It may significantly be noted that the latter is directly proportional to d_i given by (3.5.12), hence, in the above example, we choose the nearest half-plane number 4 from

$$\min_i (d_i), \quad i = 4, 5, \dots, 8. \quad (3.5.15)$$

The half-plane 4 is found to possess another interesting property namely, if we consider the angle θ_i given by (3.5.4) for the four bounding half-planes then

$$\left. \begin{aligned} \theta_4 &= \min_i (\theta_i), \quad i = 4, 5, 6, 7 \\ \phi_4 &= \max_i (\phi_i), \quad i = 4, 5, 6, 7 \end{aligned} \right\} \quad (3.5.16)$$

(also,

It might be noticed that in determining $\min_i \theta_i$ (or $\max_i \phi_i$) we considered only four bounding half-planes, that is 4, 5, 6 and 7. The redundant half-plane 8 had intentionally been excluded for if it is included in determining $\min_i \theta_i$ (or $\max_i \phi_i$), $i = 4, 5, \dots, 8$ then, as could be seen from figure 3.1, the minimum value now occurs for the half-plane 8, instead of 4. Obviously we would like to guard ourselves against moving to a completely redundant half-plane and hence do not follow this latter criterion in selecting the bounding hyperplanes for our moves in different iterations. (One salient feature, however, emerges from the above discussion, namely that if a half-plane satisfies both (3.5.15) and (3.5.16) then it must pass through the optimum point M).

If we, therefore, follow the criterion given by (3.5.15) then it adequately takes care of excluding the (completely) redundant half-planes and also ensures a move in the relevant²¹ feasible and/or infeasible region in our search for the optimum point. It may also be observed here that to reach the half-plane 4, we have to move (whichever axis we might choose) into the infeasible region. This, however, need not necessarily be the case in all situations. For example if we remove the half-planes numbers 2 and 5 from consideration for a moment (their removal, as we can see from the figure, does not effect the optimum point M) then the point R on the x_2 -axis to which we move is in the feasible region; moving on x_1 -axis (for reaching the half-plane 4) would, however, still lead to an infeasible point P, say, not shown in the figure 3.1. The decision whether to move to the point R or P is guided by consideration of the maximal increase in the optimal function (and not by consideration of movements in the preselected feasible or infeasible region) so that we would move to the (invisible) point P in the above example. The movement to a feasible or an infeasible point will, in each iteration, however, depend on (i) the constraint system and (ii) the functional of the given l.p. problem. The latter perhaps needs some elaboration for clearer understanding. Let us consider that in our example, the optimal function (3.5.1) is changed to ,

²¹This, in general, would include any such redundant bounding half-plane as number 7 in the infeasible region of figure 3.1.

$$F = -c_1 x_1 + c_2 x_2 \quad (3.5.17)$$

instead, so that the functional Z now is

$$Z = c_1 x_1 - c_2 x_2 \quad (3.5.18)$$

The selected half-plane would then in this situation be one that is nearest to the origin O in the increasing direction of the (mod) $Z = c_2 x_2$ plane. It may be noted here that we do not thus (in the first iteration) consider the term ' $-c_1 x_1$ ' of the optimal function (3.5.17) or in other words the term ' $+c_1 x_1$ ' of the functional (3.5.18). This is because the maximum increase (from the origin O) can, in this case, obviously be obtained by moving along the x_2 -axis alone. Moreover, having thus selected the nearest bounding half-plane, we would, in this case, move to it also by the x_2 -axis alone (in general, it would be one of the mod Z_j ($j \in J_1$) axes). This is because we restrict our moves in the sub-space \mathcal{A}' .

To summarise, we move in each iteration of the B.H.P. method from one vertex to another in the sub-space \mathcal{A}' defined by the set of those non-basic variables for which $Z_j < 0, j \in J$. The decision whether the vertex, that we move to, lies in the feasible or the infeasible region is governed entirely by the sub-space \mathcal{A}' . The movements on the bounding hyperplanes in the

increasing direction of the normal to the mod Z_j plane (hyperplane, in general) keeps us directing towards the optimum point in each iteration till we finally reach the latter, if it exists. And as the initial starting point could be either feasible or infeasible we need also not concern ourselves with the problem of starting the moves always from an initial basic feasible solution point (as in the Simplex Method).

3.6 Estimation of the running time.

Excluding the bookkeeping operations, there will approximately be $\frac{1}{2} (5mn + 6m + 5n)$ multiplications and $\frac{1}{2} m (3n-1)$ additions in each iteration. If ν and γ denote the multiply and add times of the Computer respectively then the time for an iteration is about

$$\frac{1}{2} \{ (5mn + 6m + 5n) \nu + m (3n-1) \gamma \} \quad (3.6.1)$$

and if Ω may denote the number of iterations required to reach the optimal solution then the total time required will be of the order of

$$\frac{\Omega}{2} \{ mn (5\nu + 3\gamma) + m (6\nu - \gamma) + 5n\nu \} \quad (3.6.2)$$

3.7 Illustrations.

We next illustrate the method with examples.

3.7.1 Example 1: Consider the problem of maximising

$$F = x_1 + 1.1x_2$$

subject to the conditions,

$$2x_1 + x_2 \geq 4$$

$$2x_1 + 3x_2 \geq 6$$

$$x_1 - 2x_2 \leq 4$$

$$x_1 + 2x_2 \geq 6$$

$$x_1 + 4x_2 \geq 8$$

$$x_1 - x_2 \leq 8$$

$$5x_1 - 3x_2 \leq 50$$

$$4x_1 - x_2 \leq 48$$

$$5x_1 + x_2 \leq 75$$

$$-4x_1 + x_2 \leq 1.5$$

$$-3x_1 + x_2 \leq 4$$

$$-2x_1 + x_2 \leq 5$$

$$-x_1 + x_2 \leq 6$$

$$-2x_1 + 3x_2 \leq 21$$

$$-x_1 + 3x_2 \leq 27$$

$$x_1 + 12x_2 \leq 168$$

$$3x_1 + 13x_2 \leq 169$$

$$-x_1 + 4x_2 \geq 0$$

$$x_1 - 3x_2 \leq 1$$

$$\text{and } x_j \geq 0 \quad j = 1, 2$$

Rewriting the above problem in the form given by (3.1.1) - (3.1.3),

we obtain,

maximise

$$x_1 + 1.1x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6 + 0x_7 + 0x_8 + 0x_9 + 0x_{10} + 0x_{11} + 0x_{12} + 0x_{13} + 0x_{14} + 0x_{15} + 0x_{16} + 0x_{17} + 0x_{18} + 0x_{19} + 0x_{20} + 0x_{21}$$

subject to.

$$\begin{array}{rcl} -2x_1 - x_2 & + x_3 & = -4 \\ -2x_1 - 3x_2 & + x_4 & = -6 \\ x_1 - 2x_2 & + x_5 & = 4 \\ -x_1 - 2x_2 & + x_6 & = -6 \\ -x_1 - 4x_2 & + x_7 & = -8 \\ x_1 - x_2 & + x_8 & = 8 \\ 5x_1 - 3x_2 & + x_9 & = 50 \\ 4x_1 - x_2 & + x_{10} & = 48 \\ 5x_1 + x_2 & + x_{11} & = 75 \\ -4x_1 + 0.5x_2 & + x_{12} & = 1.5 \\ -3x_1 + x_2 & + x_{13} & = 4 \\ -2x_1 + x_2 & + x_{14} & = 5 \end{array}$$

$$\begin{array}{rcl}
 -x_1 + x_2 & +x_{15} & = 6 \\
 -2x_1 + 3x_2 & +x_{16} & = 21 \\
 -x_1 + 3x_2 & +x_{17} & = 27 \\
 x_1 + 12x_2 & +x_{18} & = 168 \\
 3x_1 + 13x_2 & +x_{19} & = 169 \\
 x_1 - 4x_2 & +x_{20} & = 0 \\
 x_1 - 3x_2 & +x_{21} & = 1
 \end{array}$$

and $x_j \geq 0$, $j = 1, 2, \dots, 21$.

The initial point is

$$(0, 0, -4, -6, -8, 8, 50, 48, 75, 1.5, 4, 5, 6, 21, 27, 168, 169, 0, 1),$$

which, it should be noted, is an infeasible (basic) solution.

The value of the objective function at this point is obviously equal to 0. The new point reached in iteration 1 is

$$(56.3, 0.0, 108.7, 106.7, -52.3, 50.3, 48.3, -48.3, -231.7, -177.3, -206.7, 226.8, 173.0, 117.7, 62.3, 133.7, 83.3, 111.7, 0.0, -55.3, -56.3)$$

with the new objective function value equal to 56.33. The point reached is still infeasible; however, in iteration 2, we reach the optimum point,

$$(13, 10, 32, 50, 11, 27, 45, 5, 15, 6, 0, 48.5, 33, 21, 9, 17, 10, 35, 0, 18, 27),$$

which is a feasible point with all $Z_j > 0$ and the optimum value equal to 24. The calculations in each iteration are shown on the right hand side of the tables concerned with the particular iteration and are self-explanatory. Referring to the 2-dimensional diagram in figure 3.3 on page

TABLE 3.2

FIRST ITERATION		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	CALCULATIONS $(\sum_{i \in I} x_i < 0) \&$ $(\sum_{j \in J} x_j > 0) \&$ - CLASS S.	Δ_i	Pivotal Element REMARKS						
i	j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21									
C		1.0	1.1																												
C	Basis	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}	P_{14}	P_{15}	P_{16}	P_{17}	P_{18}	P_{19}	P_{20}	P_{21}									
1	P_3	0.0	7.7	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	108.7				
2	P_4	0.0	5.7	0.0	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	106.7				
3	P_5	0.0	-6.3	0.0	0.0	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-52.3	2.58			
4	P_6	0.0	2.3	0.0	0.0	0.0	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	0.0	0.0	0.0	0.0	0.0	0.0	0.0	50.3				
5	P_7	0.0	0.3	0.0	0.0	0.0	0.0	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	0.0	0.0	0.0	0.0	0.0	0.0	48.3				
6	P_8	0.0	-5.3	0.0	0.0	0.0	0.0	0.0	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	0.0	0.0	0.0	0.0	0.0	-48.3	2.83			
7	P_9	0.0	-24.7	0.0	0.0	0.0	0.0	0.0	0.0	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	0.0	0.0	0.0	0.0	-231.7	2.41			
8	P_{10}	0.0	-18.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	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	0.0	0.0	0.0	-177.3	3.01			
9	P_{11}	0.0	-20.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	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	0.0	0.0	-206.7	3.10			
10	P_{12}	0.0	17.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	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	0.0	226.8				
11	P_{13}	0.0	14.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	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	173.0				
12	P_{14}	0.0	9.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	117.7				
13	P_{15}	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	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	62.3				
14	P_{16}	0.0	11.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	133.7				
15	P_{17}	0.0	7.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	83.3				
16	P_{18}	0.0	7.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	111.7				
17	P_{19}	1.0	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	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	56.3				
18	P_{20}	0.0	-7.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-55.3	2.36			
19	P_{21}	0.0	-8.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-56.3	2.11			
Z_j		0.0	3.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	56.3			

NOTE: THE PIVOTAL ELEMENT IS INDICATED BY A CIRCLE AROUND IT.

$\therefore y_1 = 9$
from (30.12)
and $y = 2$
from (32.15)
Hence the
pivot element
is given by
 $k = 9.2$
 $= -20.7$

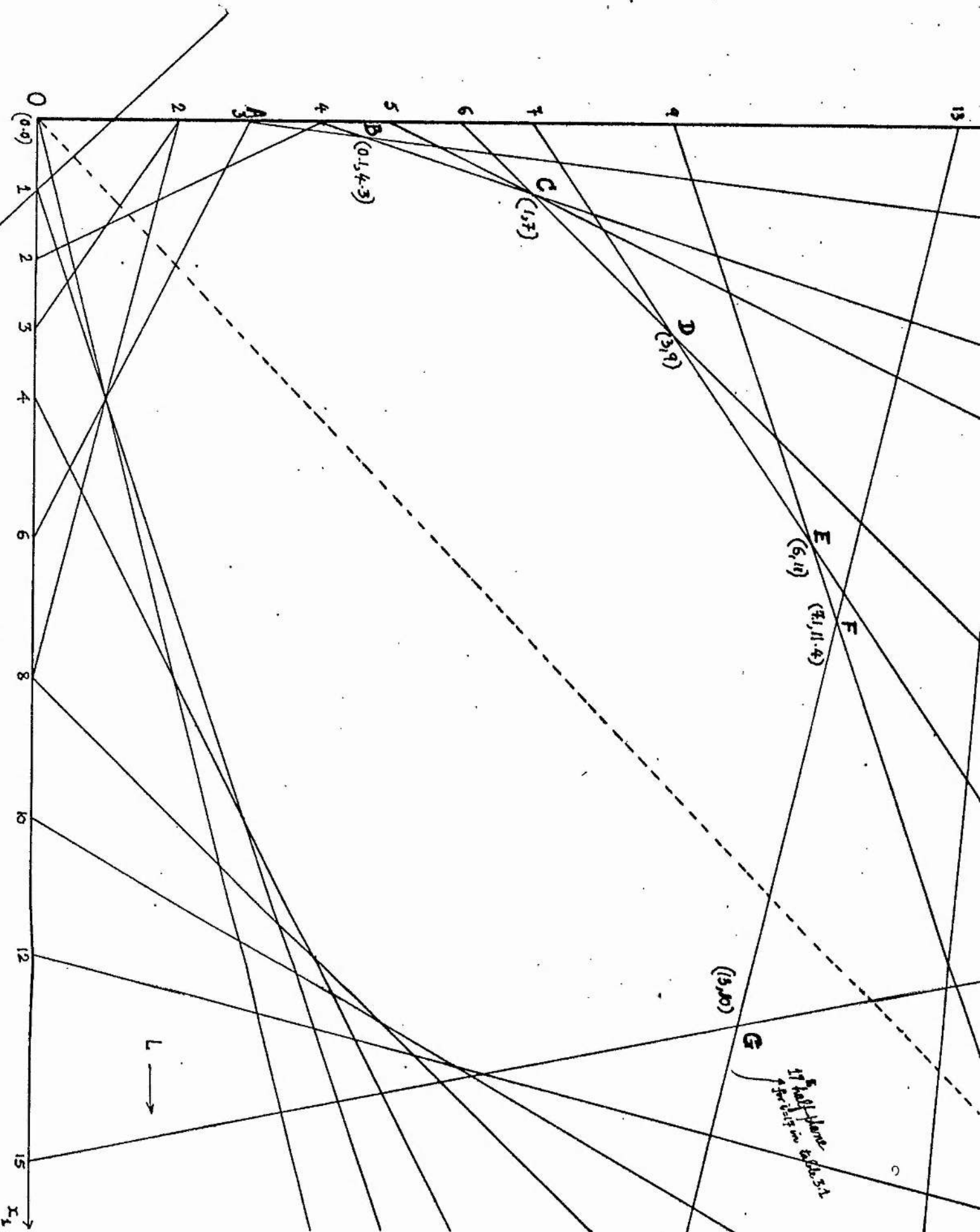


Fig. 3.3

52 we observe that the point reached in iteration 1 is given by ^{21a}L, and that the next move leads us straight to the optimum point G. If, however, we had chosen to move along the (basic) feasible points alone then, in the first place, we would require at least two iterations to get to a feasible point, say, B in the diagram and subsequently require at least 5 iterations to reach the optimum point. Moreover, we would encounter the degeneracy situation in the point C and that consumes an additional iteration to get away from the then current feasible point C to the next feasible point D. We would, therefore, require a total of 8 iterations to reach the optimum point in this case, as against 2 iterations in following the bounding hyperplane method that allows us to move in either the infeasible region alone or the feasible region alone or else both the infeasible and feasible regions, till we reach the optimum point which of course has to be in the feasible region only. In the present example, the moves were all in the infeasible region, till we reached the optimum point G in the feasible region.

37.2 Example 2: Consider another example of minimising

$$-3/4x_1 + 150x_2 - 1/50x_3 + 6x_4$$

subject to the conditions,

$$1/4x_1 - 60x_2 - 1/25x_3 + 9x_4 \leq 0$$

$$1/2x_1 - 90x_2 - 1/50x_3 + 3x_4 \leq 0$$

$$x_3 \leq 1$$

and $x_j \geq 0$, $j = 1, 2, 3, 4$.

^{21a}The point L is the X'n of hyperplane no. 17 with x_1 -axis and is not shown in the figure 3.3.

The above is an unpublished example [35] of cycling in terms of the primal (for the Simplex Method) due to Beale. Since, we have been considering the general linear programming problem as always consisting of a maximising objective function, we convert the minimising function to the maximising one by multiplying the former by -1 ; rewriting the problem in the form (1.7) to (1.9), we have:

maximise

$$0.75x_1 - 150.00x_2 + 0.02x_3 - 6.00x_4 + 0.0x_5 + 0.0x_6 + 0.0x_7$$

subject to the conditions,

$$0.25x_1 - 60.00x_2 - 0.04x_3 + 9.00x_4 + x_5 = 0$$

$$0.50x_1 - 90.00x_2 - 0.02x_3 + 3.00x_4 + x_6 = 0$$

$$x_3 + x_7 = 1$$

and $x_j \geq 0, j = 1, 2, \dots, 7$.

The calculations are shown in the adjoining tableaux, which are self explanatory.

The initial point is $(0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 1.00)$

The point remains stationary in the first iteration. In the next iteration, however, the optimum point given by

$$(0.04, 0.00, 1.00, 0.00, 0.03, 0.00, 0.00)$$

is reached with the optimal value of the objective function equal to $1/20$.

It may be of interest to mention here that the above artificially constructed example to illustrate cycling (with seven bases) in the Simplex Method takes, even after using ϵ -

Table 3.4

i \ j	j						
	1	2	3	4	5	6	7
C	0.75	-150.00	0.02	-6.00	0.00	0.00	0.00
Basis	P ₁	P ₂	P ₃	P ₄	P ₅	P ₆	P ₇
1	0.25	-60.00	-0.04	9.00	1.00	0.00	0.00
2	0.00	-90.00	-0.02	3.00	0.00	1.00	0.00
3	0.00	0.00	1.00	0.00	0.00	0.00	1.00
Z _j	-0.75	150.00	-0.02	6.00	0.00	0.00	0.00

Calculations : $(\exists j \in J) (Z_j < 0)$

$$\delta_1 = -0.1867; \Delta_1 = 0.0;$$

$$\delta_2 = -0.3746; \Delta_2 = 0.0;$$

$$\delta_3 = -0.02; \Delta_3 = 50.0;$$

Since $\Delta_1 = \Delta_2 = 0.00$ (degeneracy),
 using the (Simpler) Criterion
 (3.4-6), we choose η corresponding to
 δ_2 (as $|\delta_2| > |\delta_1|$).

Since $x_2 = 0$, $\tau = 1$ from (3.2.6);
 Pivot = $\bar{b}_{21} = 0.50$.

Iteration 1

1	0.00	P ₅	0.00	-15.00	-0.03	7.50	1.00	-0.50	0.00	0.00
2	0.75	P ₁	1.00	-180.00	-0.04	6.00	0.00	2.00	0.00	0.00
3	0.00	P ₂	0.00	0.00	1.00	0.00	0.00	0.00	1.00	1.00
	Z _j		0.00	15.00	-0.05	10.50	0.00	1.50	0.00	0.00

Calculations: ($34 \in J$) ($Z_4 < 0$)

$$\delta_1 \geq 0$$

$$\delta_2 \geq 0$$

$$\delta_3 = -0.05; \Delta_3 = 20.00$$

$$; \Delta_n = \Delta_3 = 20.00 \text{ from (3.1.21)}; \therefore n = 3 ;$$

Since $x_3 = 0$, $\tau = 3$ from (3.2.6) ;

$$\text{Pivot} = \bar{b}_{33} = 1.00.$$

Table 36

Iteration 2

1	0.00	P ₅	0.00	-15.00	0.00	7.50	1.00	-0.50	0.03	0.03
2	0.75	P ₁	1.00	-180.00	0.00	6.00	0.00	2.00	0.04	0.04
3	0.02	P ₃	0.00	0.00	1.00	0.00	0.00	0.00	1.00	1.00
		Z _j	0.00	15.00	0.00	10.50	0.00	1.50	0.05	0.05

Calculations

57

As the condition (5.2.10) is true the optimum point is reached. The optimum value is 0.05 and the solution point is

$$x_1 = 0.04$$

$$x_5 = 0.03$$

$$x_2 = 0.00$$

$$x_6 = 0.00$$

$$x_3 = 1.00$$

$$x_7 = 0.00$$

$$x_4 = 0.00$$

Note: 1. The figure shown in the bottom extreme right cell gives the value of the objective function in each iteration.

2. The pivotal element is indicated by a circle around it.

perturbation schemes for resolving degeneracy, six iterations to reach the optimum point [35].

3.7.3 Example 3. Let us consider Beale's [5] example (in terms of the dual also) that illustrates cycling in the Dual Simplex Algorithm [56]. The problem is to minimise

$$\begin{array}{rcll}
 F = & & x_3 & \\
 \text{subject to} & & & \\
 & -1/4x_1 & - 1/2x_2 & +x_4 = -3/4 \\
 & 8x_1 & + 12x_2 & +x_5 = 20 \\
 & x_1 & + 1/2x_2 & - x_3 +x_6 = -1/2 \\
 & -9x_1 & - 3x_2 & +x_7 = 6
 \end{array}$$

and $x_j \geq 0$, $j=1,2,\dots,7$.

The minimising function could be rewritten as a maximising function in the form

$$F = -x_3$$

This example calls for the direct application of class S rules alone. The calculations are shown in the adjoining tableaux. It may be observed that the optimum point given by

$$(0, 3/2, 5/4, 0, 20, 21/2)$$

is reached in two iterations. The problem otherwise cycles with six bases in the Dual Simplex method and can be solved only by following one of the ϵ -perturbation schemes.

		1	2	3	4	5	6	7	
i \ j									
	C	0	0	-1	0	0	0	0	
	\tilde{C}								
	Basis	P_1	P_2	P_3	P_4	P_5	P_6	P_7	$P_0 (= \tilde{X})$
1	0	-1/4	-1/2	0	1	0	0	0	-3/4
2	0	8	12	0	0	1	0	0	20
3	0	1	1/2	-1	0	0	1	0	-1/2
4	0	-9	-3	0	0	0	0	1	6
	Z_j	0	0	1	0	0	0	0	

Calculations: $((j \in I) \{x_i < 0\} \& (\forall i \in J) \{Z_j \geq 0\})$

Since $x_4 < 0, x_6 < 0$ and $Z_j \geq 0, (j=1, 2, \dots, 7)$, rules of class S will apply, thus:

$\delta_1^1 = 0$ not considered,

$\delta_3^1 = -1, \Delta_3^1 = 1/2; \therefore \eta^1 (= i \in I_1) = 3;$

and $\tau = 3$, from (3.2.15);
 hence, pivot $= b_{33} = -1$.

Table 3.8

Iteration 1.

1	0	P ₄	-1/4	-1/2	0	1	0	0	0	-3/4
2	0	P ₅	8	12	0	0	1	0	0	20
3	-1	P ₃	-1	-1/2	1	0	0	-1	0	1/2
4	0	P ₇	-9	-3	0	0	0	0	1	6
		Z _j	1	1/2	0	0	0	1	0	-1/2

Calculations

Rules of class S still apply, thus:

$$\delta_1 = -1/2, \Delta_1 = 3/2 ; \therefore \eta' = 1 ;$$

Max (-4, -1) corresponds to $r = 2$

From (3.2.15);

hence, pivot = $\bar{b}_{12} = -1/2$.

Table 3.9

Iteration 2

1	0	P_2	1/2	1	0	-2	0	0	0	3/2
2	0	P_5	2	0	0	24	1	0	0	2
3	-1	P_3	-3/4	0	1	-1	0	-1	0	5/4
4	0	P_7	-15/2	0	0	-6	0	0	1	21/2
		Z_j	3/4	0	0	1	0	1	0	-5/4

Calculations

As the condition (3.2.10) is true, the optimum point is reached.
The optimal value is $-5/4$ and the solution point is

$$x_1 = 0 \quad x_5 = 2$$

$$x_2 = 3/2 \quad x_6 = 0$$

$$x_3 = 5/4 \quad x_7 = 21/2$$

$$x_4 = 0$$

- Note: 1. The figure shown in the bottom extreme right cell gives the value of objective function in each iteration.
2. The pivotal element is indicated by a circle around it.

Chapter 4

THE BOUNDING HYPERPLANE METHOD OF LINEAR PROGRAMMING - PART II

4.1 Equalities and their relationship to inequalities.

As stated in the last chapter, we now consider the case when the equalities occur together with the inequalities. In this case it is essential that the former must be satisfied exactly in the optimum point. We, therefore, solve first the equality constraints independently of the inequality constraints and if the 'equality' solution tableau so obtained does not yield an inconsistent solution point then the former is utilised to obtain the 'transformed' inequality tableau (corresponding to the inequality constraints) for the current 'equality' solution point. The set of the transformed constraints thus obtained, corresponding to both the equality and inequality constraints, is now a l.p. problem with only inequality constraints and is solved by the application of the B.H.P. method discussed in chapter 3.

Let us reconsider the l.p. problem formulated in section 2.1, chapter 2. We, for the sake of convenience of description, however, slightly change the notation so that the column vector P_0 is denoted by P_A , and its elements which may be negative and/or non-negative are given by

$$P_A = \begin{array}{c} | \\ a_{1,n+1} \\ | \\ a_{2,n+1} \\ | \\ \cdot \\ | \\ \cdot \\ | \\ \cdot \\ | \\ \cdot \\ | \\ \cdot \\ | \\ a_{m,n+1} \\ | \end{array} \quad (4.1.1)$$

Let us suppose the augmented matrix $(A | P_A) \{ a_{ij} \}$, an m by $n+1$ matrix } to be partitioned into two submatrices $(V | P_V) \{ \beta_{ij} \}$ an m_1 by $n+1$ matrix } and $(W | P_W) \{ \alpha_{ij} \}$ an m_2 by $n+1$ matrix } corresponding to the inequality and the equality constraints respectively where

$$m = m_1 + m_2 \quad (4.1.2)$$

Let I and J be the sets of indices of the rows and columns respectively of $A = \begin{pmatrix} V \\ W \end{pmatrix}$ and let I_V and I_W be the sets of indices of rows of V and W respectively so that

$$I_V \cup I_W = I \quad ; \quad I_V, I_W \subset I \quad (4.1.3)$$

Further, if the submatrix W comprises of the null column vectors, if any, numbering say n' then let $J' (\subset J) / J_{Ea} (\subset J)$ define the sets of indices of columns of W that are/are not the null column vectors so that

$$J' \cup J_{EQ} = J \quad , \quad (4.1.4)$$

and

$$r = n - n' \quad , \quad (4.1.5)$$

where r defines the effective number of columns of W .

4.2 The equality constraints. As stated above we consider first the submatrix W independently, for the time being, of the submatrix V . Since the equality constraints must be satisfied exactly we make the condition that an equality once selected as a pivotal row will not be reconsidered for pivotal row selection in the subsequent iterations. For this purpose, we let

$I_{\bar{W}}^E \subset I_W$ denote the set of indices i for which the equality hyperplanes i ($\in I_W$) have been considered and let $I_{\bar{W}}^{CE} \subset I_W$ denote the set of indices i for which the hyperplane i ($\in I_W$) are yet to be considered so that

$$I_{\bar{W}}^E \cup I_{\bar{W}}^{CE} = I_W \quad , \quad (4.2.1)$$

where \bar{W} ($= \bar{\alpha}_{ij}$) denotes the transformed matrix W at the points reached in different iterations. (The latter may significantly vary depending upon the technique deployed in the solution of the equality constraints for, with the possibility of starting with an infeasible solution point as well in the B.H.P. method we are at liberty to choose any one out of a number of

techniques in the literature). Further if the row vector $Z (= Z_j, j \in J)$ denotes the transformation to the functional '-CX' in different iterations then let $J_1 \subset J_{EQ}$ denote the set of indices j for which

$$Z_j < 0, \quad j \in J_1; \quad (4.2.2)$$

and also if $\bar{W}_i, i \in I_W$ defines the row space of \bar{W} then let $\bar{W}_i^j (i \in I_W, j \in J_1)$ denote the corresponding row elements of \bar{W}_i for which (4.2.2) holds.

4.2.1. Techniques.

We now describe two techniques for the solution of equality constraints. Let $\bar{W}_{\eta\gamma}$ in both the techniques denote, in general, the pivotal element where η , as before, stands for the pivotal row and γ the pivotal column.

(i) Technique 1. The pivotal row $\eta (= i \in I_W^{CE})$ in this approach is chosen by successively considering each of the m_2 rows (starting from the first) in that order. Then if

$$((\forall j \in J_{EQ}) (Z_j \geq 0)) \quad (4.2.3)$$

γ is chosen from

$$\max_j |\bar{\alpha}_{\eta j}|, \quad j \in J_{EQ}; \quad (4.2.4)$$

but if

$$((\exists j \in J_{EQ}) (Z_j < 0)) \quad (4.2.5)$$

then γ is chosen from

$$\max_j \left| \overline{w}_\eta^j \right|, \quad j \in J_{E\alpha}, \quad (4.2.6)$$

if it exists, otherwise it is selected from (4.2.4).

The new tableau is obtained by carrying out the pivotal transformations using formulae (4.2.21). The equality operations terminate as discussed in section 4.2.3.

(ii) Technique 2. In this approach we utilise the concepts developed for the B.H.P. method in chapter 3. We first distinguish, as in the technique 1 above, the two situations given by (4.2.5) and (4.2.3). It may be recalled that these two situations are termed the Classes R and S in the last chapter; refer (3.2.1) and (3.2.11). To maintain analogy we rename them as classes RE and SE (E abbreviation for equality). The pivotal element $\overline{\alpha}_{\eta\gamma}$ is then determined following the rules discussed below for the two classes.

Class RE. $((\exists j \in J_{E\alpha}) (Z_j < 0)) \quad (4.2.7)$

Rule RE.1. Select the pivotal row $\eta (= i \in I_{\overline{w}}^{CE})$ as that hyperplane which is farthest in the increasing direction of the normal to the Z_j ($j \in J_{E\alpha}$) plane from the origin. It is given by

$$\max_i \left| \overline{\alpha}_{i,n+1} / \sum_j \overline{\alpha}_{ij} Z_j \right|, \quad (4.2.8)$$

$i \in I_{\overline{w}}^{CE}, j \in J_{E\alpha}.$

If a tie occurs then we select any of the $i \in I_{\bar{w}}^{CE}$ for which tie occurs. This is termed the exit criterion for the class R.

Next we examine if

$$(i) \quad \bar{\alpha}_{\eta, n+1} \geq 0 \quad , \quad (4.2.9)$$

$$\text{or (ii)} \quad \bar{\alpha}_{\eta, n+1} < 0 \quad . \quad (4.2.10)$$

Rule RE.2. The entry criteria^{21b} for the two cases above are as below.

RE.2.1. If (4.2.9) is true then $\Upsilon (= j \in J_{EQ})$ is chosen from

$$\min_j (Z_j / \bar{\alpha}_{\eta j}) , \bar{\alpha}_{\eta j} < 0 , j \in J_{EQ} ; \quad (4.2.11)$$

but if

$$\left((\forall j \in J_{EQ}) (\bar{\alpha}_{\eta j} \geq 0) \right) \quad (4.2.12)$$

so that Υ cannot be selected from (4.2.11) then we choose it from

$$\min_j (Z_j / \bar{\alpha}_{\eta j}) , \bar{\alpha}_{\eta j} \neq 0 , j \in J_{EQ} ; \quad (4.2.13)$$

RE.2.2. If (4.2.10) is true then $\Upsilon (= j \in J_{EQ})$ is chosen from

$$\max_j (Z_j / \bar{\alpha}_{\eta j}) , \bar{\alpha}_{\eta j} > 0 , j \in J_{EQ} \quad (4.2.14)$$

21b. If a tie occurs then we select any of the $j \in J_{EQ}$ for which the tie occurs.

but if

$$\left((\forall j \in J_{EQ}) (\bar{\alpha}_{\eta j} \leq 0) \right) \quad (4.2.15)$$

then we select Υ alternatively from

$$\max_j (Z_j / \bar{\alpha}_{\eta j}), \bar{\alpha}_{\eta j} \neq 0, j \in J_{EQ} \quad (4.2.16)$$

This is termed the entry criterion for the class RE.

Class SE $\left((\forall j \in J_{EQ}) (Z_j \geq 0) \right) \quad (4.2.17)$

Rule SE.1. Select the pivotal row η as any one of the hyperplanes $i \in I_{\bar{W}}^{CE}$. This is termed the exit criterion for the class SE.

Rule SE.2. Select $\Upsilon (= j \in J_{EQ})$ from

$$\min_j |Z_j / \bar{\alpha}_{\eta j}|, \bar{\alpha}_{\eta j} < 0, j \in J_{EQ}; \quad (4.2.18)$$

but if

$$\left((\forall j \in J_{EQ}) (\bar{\alpha}_{\eta j} \geq 0) \right) \quad (4.2.19)$$

then Υ is chosen from

$$\min_j (Z_j / \bar{\alpha}_{\eta j}), \bar{\alpha}_{\eta j} \neq 0, j \in J_{EQ} \quad (4.2.20)$$

(A tie is handled as in the previous case)

This is termed the entry criterion for the class SE.

4.2.2. The new tableau. This is obtained at the end of each iteration by performing the Gaussian eliminational transformations on the pivotal element $\bar{\alpha}_{\eta\Upsilon}$, from

$$\left. \begin{aligned} \bar{\alpha}'_{ij} &= \bar{\alpha}_{ij} - \frac{\bar{\alpha}_{\eta j}}{\bar{\alpha}_{\eta \tau}} \cdot \bar{\alpha}_{i\tau}, \quad i (\neq \eta) \in I_{W+1} \\ \bar{\alpha}'_{ij} &= \bar{\alpha}_{ij} / \bar{\alpha}_{\eta \tau}, \quad i = \eta \end{aligned} \right\} \begin{array}{l} j \in J_{E\Omega} + 1; \\ (4.2.21) \end{array}$$

where a prime denotes the new values of the elements in the new tableau. The new values of the optimal function F , the functional Z and the Solution vector $\bar{\alpha}_{i, n+1}$, $i \in I_W (= \bar{X}_W)$ respectively are given by

$$\begin{aligned} \bar{\alpha}'_{ij} &= \bar{\alpha}'_{m_2+1, n+1} = F \\ \bar{\alpha}'_{ij} &= \bar{\alpha}'_{m_2+1, j}, \quad j \in J_{E\Omega} = Z (= Z_j), \quad (4.2.22) \\ \bar{\alpha}'_{ij} &= \bar{\alpha}'_{i, n+1}, \quad i \in I_W = \bar{X}_W (= \bar{a}_{i, n+1}). \end{aligned}$$

4.2.3. The final equality $(\bar{W} | P_{\bar{W}})$ tableau.

If $I_{\bar{W}}^{CE} = I_{\bar{W}}^{red}$, that is,

$$I_{\bar{W}}^E \cup I_{\bar{W}}^{red} = I_W, \quad (4.2.23)$$

where $I_{\bar{W}}^{red} \subset I_{\bar{W}}^{CE}$ denotes the set of indices i (redundant hyperplanes) for which

$$\left((\exists i \in I_{\bar{W}}^{CE}) \left((\bar{\alpha}_{i, n+1} = 0) \& (\forall j \in J_{E\Omega}) (\alpha_{ij} = 0) \right) \right) \quad (4.2.24)$$

then the equality operations of the section 4.2 terminate. The

tableau so obtained is termed the final equality tableau. If in this tableau it is found that

$$\left(\left(\exists i \in I_{\bar{w}} \right) \left(\left(\bar{\alpha}_{i, n+1} \neq 0 \right) \& \left(\forall j \in J_{\epsilon a} \right) \left(\alpha_{ij} = 0 \right) \right) \right) \quad (4.2.25)$$

then it implies that the solution to the problem does not exist (inconsistent solution point).

If (4.2.25) is not true and

$$m_1 = 0 \quad (4.2.26)$$

then the optimal solution, if it exists, is obtained as discussed in section 4.4, but if

$$m_1 \neq 0 \quad (4.2.27)$$

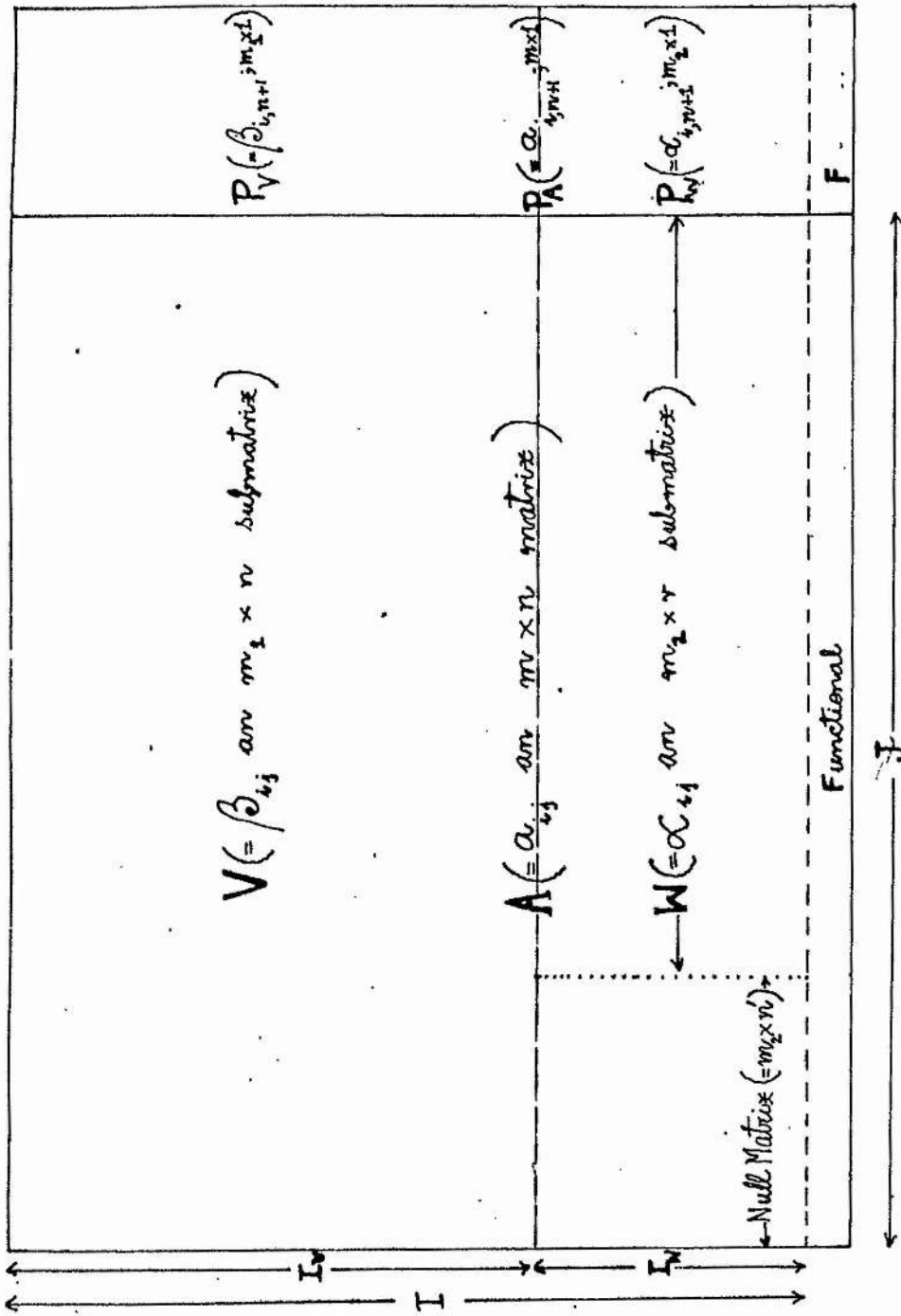
then we develop in section 4.3 the relations for obtaining the transformations to the inequality constraints $(V|P_v)$ from the final equality tableau.

4.2.3.1. Motivation in selecting one of the two techniques (i) and (ii).

It may be useful to mention here the reasons that prompted us for suggesting the two techniques given above. The technique 1 takes care of reducing the rounding errors by selecting the pivot as that element which has the largest magnitude in the pivotal row. This is strictly the case in the situation (4.2.3).

There is, however, a compromise attempted with situation (4.2.5) when the pivotal element is the one which has the largest magnitude corresponding to negative coefficients of the functional (refer (4.2.2)); this is because we know that the (maximum) increase to the optimal function will occur by selecting a variable from

Table. 4.1.



Organisation of the augmented matrix $(A|P)$ and the functional.

the set J_1 defined by (4.2.2). In the technique 2, on the other hand, we have essentially utilised the l.p. concepts developed for the B.H.P.M. in chapter 3. If the equality constraints are not inconsistent we would usually result at the end of equality operations of technique 2 in the situation

(4.2.17) - refer class SE, and this is a considerable advantage in reducing the number of iterations of the problem. (In some cases the solution point obtained at the end of technique 2 may well be the optimum point for the (complete) l.p. problem). If we end up with the class RE at the end of technique 2 then we have obtained the maximum increase in the optimal function or else the minimum decrease (if that be the case) in each iteration and this has been directly related to the (real) optimal function (2.1.1) and not to an auxiliary function as in the Simplex Method.

4.3 The final inequality/functional²² tableau.

Let the constraints (2.1.2) and the functional Z be represented as shown in table 4.1. Also let the final equality matrix (tableau) be as shown in the middle portion of table 4.2. The transformations to the function F and the coefficients $(\bar{\beta}_{ij})$ of the final inequality tableau are then obtained, using the coefficients $(\bar{\alpha}_{ij})$ of the final equality tableau, from

22. Abbreviated as I/F tableau for future reference.

Table 4.2

$\bar{\beta}_{ij}(\neq 0)$	$\bar{\beta}_{ij} (= 0)$	$\bar{\beta}_{ij}$	$\bar{\beta}_{i,n+1}$
Null Matrix			$\bar{\alpha}_{ij}$
$\bar{z}_i = z_i$	$\bar{z}_j = 0$	\bar{z}_j	\bar{F}
I_1		I_2	
I		I_3	
n'		n_1	
J'		J_{eq}	
J		$r - n_1$	

Final equality (after phase 1 of the Simplex Method) and the Inequality/Functional tableau.

$$\bar{\beta}_{ij} = \beta_{ij} - \sum_{k=1}^{n_1} \beta_{i, g_k} \bar{\alpha}_{h_k, j} \quad ; i \in I_V, j \in J, (4.3.1)$$

$$\bar{\beta}_{i, n+1} = \beta_{i, n+1} - \sum_{k=1}^{n_1} \beta_{i, g_k} \bar{\alpha}_{h_k, n+1} \quad , i \in I_V ; (4.3.2)$$

where

$$\left. \begin{array}{l} G = |g_i| \\ H = |h_i| \end{array} \right\} 1 \leq i \leq n_1, \quad (4.3.3)$$

n_1 ($\leq r$) is the column rank of W and g_i and h_i , $i = 1, 2, \dots, n_1$ are the elements of the index vectors G and H that record the column and row respectively of the successive elements of the W matrix used as pivots at each stage of the operations of section 4.2.1. The coefficients $\bar{\beta}_{ij}$ (for the inequality part of the table 4.2) computed from (4.3.1) relate, however, to only those columns in the final equality tableau that correspond to the column nullity $r - n_1$ ($n_1 < r$) in the original submatrix W . This is because the coefficients in the I/F tableau corresponding to the null matrix, if any, in the W submatrix (or equality tableau) remain unchanged as the terms

$$\sum_{k=1}^{n_1} \beta_{i, g_k} \bar{\alpha}_{h_k, j} = 0, \quad i \in I_V, j \in J' \quad (4.3.4)$$

on the right hand side of (4.3.1) hence,

$$\bar{\beta}_{ij} = \beta_{ij} \quad , \quad j \in J' \quad , \quad i \in I_V \quad (4.3.5)$$

And the coefficients in the I/F tableau corresponding to the unit matrix coefficients in the final equality tableau are necessarily zero (that is a null matrix).

We next consider if, in the I/F tableau,

$$\left(\left(\exists i \in I \right) \left(\left(\bar{\beta}_{i, n+1} < 0 \right) \& \left(\forall j \in J \right) \left(\bar{\beta}_{ij} \geq 0 \right) \right) \right) \quad (4.3.6)$$

which implies that the optimal solution to the problem does not exist: If (4.3.6) is not true then we concern ourselves with relevant parts of the direction analysis discussed generally below.

4.4 Directional analysis. As mentioned earlier, we distinguish two classes of situation in the given linear programming problem,

Class a: $m_2 \neq 0$, $m_1 = 0$ (equalities only);

Class b: $m_2 \neq 0$, $m_1 \neq 0$ (both equality and inequality constraints).

The following cases may exist in both the classes:

$$\left. \begin{array}{l} \text{(i)} \quad m_2 > r \\ \text{(ii)} \quad m_2 = r \\ \text{and (iii)} \quad m_2 < r \end{array} \right\} \quad (4.4.1)$$

We consider the three cases in each class separately,

Class a. (i) If it is found that $d (= m_2 - r)$ hyperplanes are redundant (that is the number of unit column vectors is exactly equal to r) then a unique solution to the problem exists; proceed to 4.4.1. However if

$$d > m_2 - r, \quad (4.4.2)$$

then an infinite number of solutions exist in which case proceed to 4.4.2.

(ii) If there is/are no redundant hyperplanes then a unique solution to the problem exists; proceed to 4.4.1. However, if there is at least one redundant hyperplane (that is the number of unit column vectors is less than r) then an infinite number of solutions exist; proceed to 4.4.2.

(iii) An infinite number of solutions exist; proceed to 4.4.2.

Class b:

Let $J_2 \subset J_{\text{eq}}$ denote the set of indices that correspond to those columns (or variables) of the \mathcal{W} submatrix that have column nullity $r - n_1$ and let J'' define the set given by

$$J'' = J' \cup J_2. \quad (4.4.3)$$

Then if

$$((\forall i \in I_V)((\forall j \in J'')(\bar{\beta}_{ij} = 0))) \quad (4.4.4)$$

the considerations in class a apply exactly to

this situation and we follow the directions of the three cases as discussed therein. If (4.4.4) is not true then there exists an infinite number of solutions and we proceed to 4.4.2.

4.4.1 The unique solution: There is an optimal solution to the problem only if

$$((\forall i \in I)(\bar{a}_{i,n+1} \geq 0) \& (\forall j \in J)(z_j \geq 0)) \quad (4.4.5)$$

where

$$\bar{a}_{i,n+1} = \bar{\beta}_{i,n+1} \quad , i \in I_v \quad , \quad (4.4.6)$$

$$\bar{a}_{i,n+1} = \bar{\alpha}_{i,n+1} \quad , i \in I_w \quad ; \quad (4.4.7)$$

otherwise the optimal solution does not exist.

4.4.2 An infinite number of solutions. These are categorised in the following three mutually exclusive and collectively exhaustive cases which uniquely direct the next move:

(i) if (4.4.5) is true,

then the current solution point given by

$$\bar{x}_{i,n+1} = \bar{a}_{i,n+1} \quad , i \in I \quad (4.4.8)$$

is the optimal point, where $\bar{a}_{i,n+1}$ are given by (4.4.6) - (4.4.7) .

(ii) If $((\forall j \in J)(z_j \geq 0) \& (\exists i \in I)(a_{i,n+1} < 0))$, (4.4.9)

then apply the class S decision rules of the B.H.P. Method chapter 3 or the Dual Simplex Method [56] to obtain the optimal solution.

$$(iii) \text{ If } ((\exists j \in J) (z_j < 0)) \quad (4.4.10)$$

then apply the class R decision rules of the B.H.P. Method, ch.3 to obtain the optimal solution of the problem.

4.5 Estimation of running time.

In view of initially restricting the pivotal transformations to only the equality constraints and then developing the transformed inequality constraints from the former we have been able to effectively reduce the time taken in each iteration in the 'equality operations'. An estimate of the running time for the equality operations alone will be

$$\frac{1}{6} m_2 [3v \{5n(m_2+1) - 3m_2(m_2-1) + 2\} + \gamma \{3n(1+3m_2) + m_2(6-5m_2) - 1\}] \quad (4.5.1)$$

This estimate for the solution of a l.p. problem that comprises both the equality and inequality constraints but does not consider the former independently of the latter, is given by

$$\left. \begin{aligned} & \frac{1}{6} m_2 [3v \{5n(m_2+1) - 3m_2(m_2-1) + 2m_1(2n - m_2 + 1) + 2\} \\ & \quad + \gamma \{3n(1+3m_2) + m_2(6-5m_2) + 3m_1(2n - m_2 + 1) - 1\}] \\ & \quad + \frac{\Omega}{2} [(n-m_2) \{(5v+3\gamma)m + 5v\} + m(6v-\gamma)] \end{aligned} \right\} \quad (4.5.2)$$

where Ω , v , γ are as defined in chapter 3. If, however, we followed the approach discussed in this chapter then (4.5.2) reduces to

$$\left. \begin{aligned} & \frac{1}{6} m_2 [3v \{5n(m_2+1) - 3m_2(m_2-1) + 2\} + \gamma \{3n(1+3m_2) + m_2(6-5m_2) - 1\}] \\ & \quad + \frac{\Omega}{2} [(n-m_2) \{(5v+3\gamma)m + 5v\} + m(6v-\gamma)], \text{ approximately} \end{aligned} \right\} \quad (4.5.3)$$

and an estimated saving in computer running time to the order of

$$\frac{1}{2} m_1 m_2 (2n - m_2 + 1) (2\gamma + \gamma) \quad (4.5.4)$$

is achieved.

4.6 Illustration. We illustrate the method using technique 2 in example 1 below.

Example 1.

Consider the problem of maximising,

$$F = x_1 + x_2 + x_3$$

subject to the conditions,

$$5x_1 + 2x_2 + 5x_3 \leq 10$$

$$3x_1 + 3x_2 + x_3 = 3$$

$$2x_1 + 2x_2 + 3x_3 = 6$$

The matrix $(A | P_A) = \left(\begin{array}{c|c} V & P_V \\ W & P_W \end{array} \right)$ is shown in table 4.3.

The subsequent iterations tableau are shown in tables 4.4. to 4.6 . We may observe that using technique (ii) for the solution of the equality constraints independently of the inequality constraints in the problem we obtain in the first iteration (tableau 4.4) the class SE situation (4.2.17) where,

$$Z_j \geq 0 \quad , \quad j \in J_{EQ} (= J, \text{ in this case}).$$

We retain this situation by considering the class SE (decision) rules now and obtain the new solution point as given in tableau 4.5. Next using the formulae (4.3.1)-(4.3.2) we obtain the I/F

Table 4.3

$i \backslash j$	1	2	3		Calculations/Remarks	
1 (1)	5	2	5	10	$\frac{\bar{c}_i}{\bar{a}_{ij}}$, refer (4.2.8). From (4.2.8) and (4.2.13) $\eta = 3$ and $\gamma = 1$; hence the pivotal element is given by \bar{a}_{31} .	
2 (1)	3	3	1	3		
3 (2)	②	2	3	6		
Z_j	-1	-1	-1	0		

Table 4.4

Iteration 1

$i \backslash j$	1	2	3		Calculations/Remarks	
1 (1)	5	2	5	10	clearly η , in this case, is for $i = 2$; using (4.2.18) $\gamma = 3$; hence the pivotal element is \bar{a}_{23} .	
2 (1)	0	0	⑦/2	-6		
3 (2)	1	1	3/2	3		
Z_j	0	0	1/2	3		

Table 4.5

Iteration 2

$i \backslash j$	1	2	3		Remarks	
1 (1)	5	2	5	10	All the equalities are, at this stage satisfied; the transformation to inequality constraints, using formulae (4.3.1)-(4.3.4) are shown in tableau 4.6.	
2 (1)	0	0	1	12/7		
3 (2)	1	1	0	3/7		
Z_j	0	0	0	13/7		

Table 4.6

$i \backslash j$		1	2	3	4	
	Basis	P_1	P_2	P_3	P_4	$P_0(\bar{x})$
1	P_4	0	-3	0	1	-5/7
2	P_3	0	0	1	0	12/7
3	P_1	1	1	0	0	3/7
	Z_j	0	0	0	0	13/7

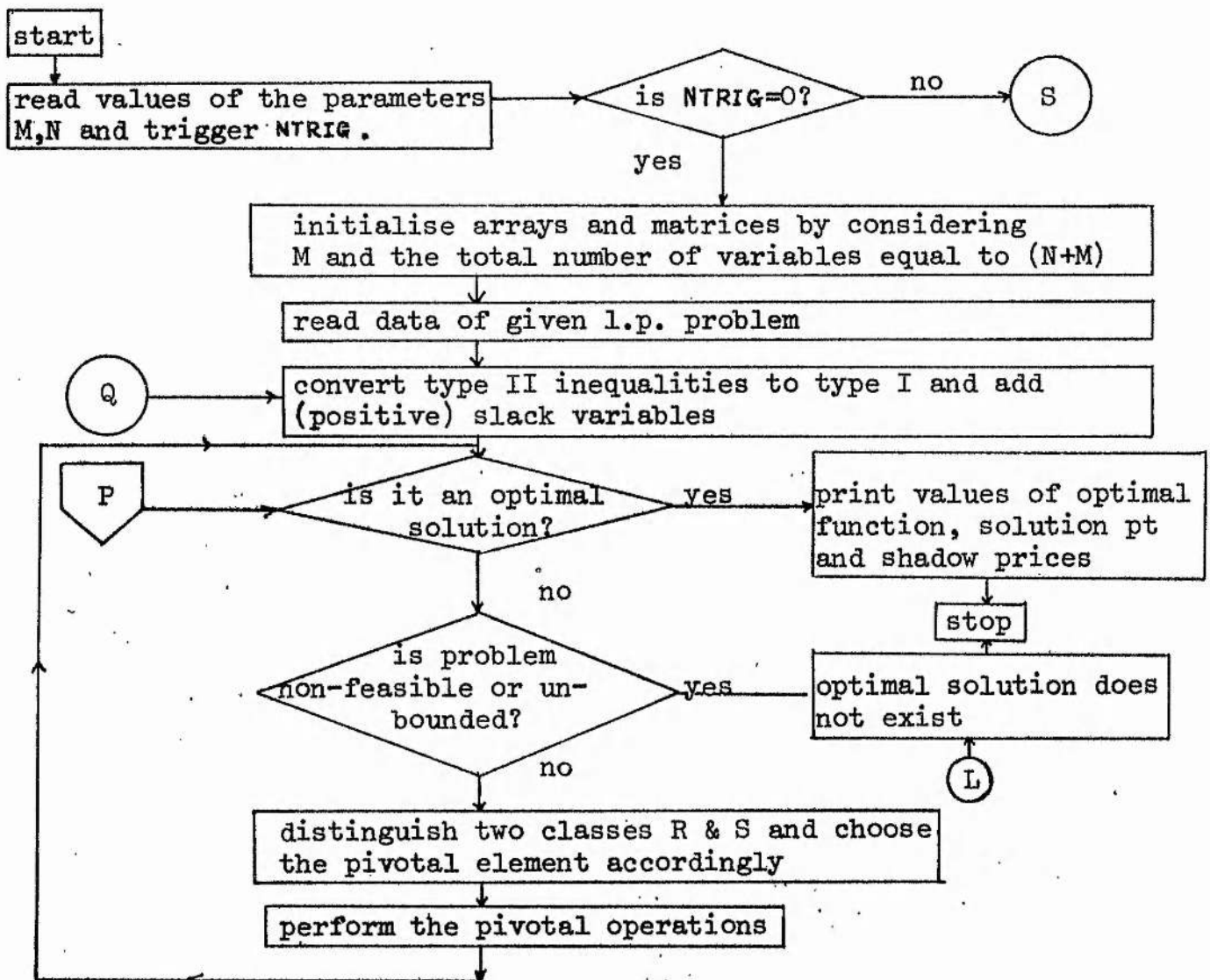
The B.H.P. tableau at the end of equality operations.

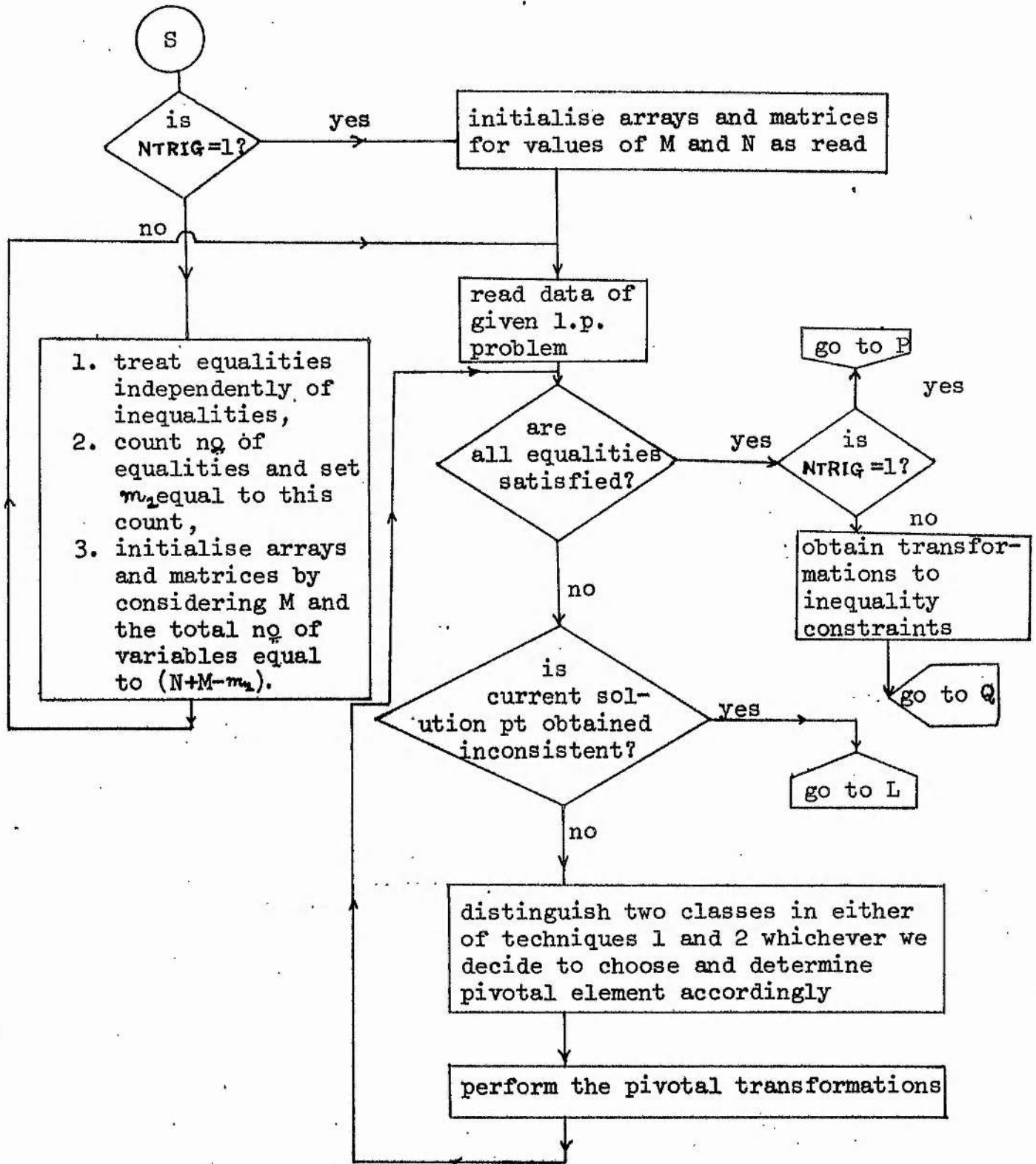
tableau (which is the initial tableau for BHPMas described in chapter 3). This tableau is given in table 4.6 and as we can see we have, by using the equality technique (ii), obtained the point of class S (chapter 3) alone. The optimum tableau is obtained (not shown) in the next iteration.

Chapter 5

FLOW DIAGRAMS AND THE GENERAL ALGORITHM

5.1 Flow diagrams. The flow diagram given below illustrates all three possibilities; that is when the l.p. problem consists of either inequalities alone (chapter 3) or equalities alone (chapter 4) or both the equalities and inequalities (chapter 4). These are respectively indicated by the value of the trigger 'NTRIG' equal to 'zero', 'one' or 'two'. M and N in the flow diagram below denote the 'number of equations+1' and 'number of variables (without addition of slack variables)+1' in the l.p. problem.





5.2

The general algorithm. Corresponding to the system of constraints (2.1.2) consisting of either (a) inequalities alone or (b) equalities alone or (c) both equalities and inequalities, set the trigger 'NTRIG' equal to zero, one or two respectively. Also set another trigger 'trig' equal to one or two according as we may wish to use either the equality solution technique 1 or 2 of chapter 4. Further let A denote the general algorithm description that follows. We are now ready to institute the complete algorithm.

```

A      Test 'NTRIG' (for zero or nonzero):

A1     If 'NTRIG' ≠ 0, set ctr to zero (where ctr is an abbreviation for the counter that keeps a count of the number of equalities
      satisfied up to a given iteration)
      ctr+1 → ctr; test*ctr:
      if ctr > m2, go to A1.2;
      otherwise test 'trig':
      if 'trig'=1, η (=δ)=ctr, i ∈ Ice; test Zj, j ∈ Jea:
          if Zj > 0 for each j, select γ from (4.2.4)
          otherwise select γ from (4.2.6)/(4.2.4) as the case
          may be. The pivotal element is  $\bar{\alpha}_{\eta r}$  (in technique 1),
          go to A.3;
          if not, test Zj, j ∈ Jea:
              if Zj > 0 for each j then choose η as any one i ∈ Ice - exit criterion, class SE;
              and select γ from either (4.2.18) or (4.2.20) as the case may be -
              entry criterion, class SE. The pivotal element is  $\bar{\alpha}_{\eta r}$  (in technique 2), go to A.3;
              otherwise select η from (4.2.8) - exit criterion class RE and select γ from
              either (4.2.11)/(4.2.13) or (4.2.14)/(4.2.16) as the case may be -
              entry criterion, class RE. The pivotal element is  $\bar{\alpha}_{\eta r}$  (in technique 2),
              go to A.3;
          test for the inconsistency, if any (in the equality solution point reached - refer (4.2.25)):
          if inconsistent, the optimal solution does not exist.
          otherwise, test 'NTRIG' (for 1 or 2):
              if NTRIG=1 then proceed as in section 4.4, class a; set NTRIG=0, go to Aj;
              if not, proceed according to section 4.4, class b; obtain transformations to the
              inequality constraints, if necessary, from (4.3.1)-(4.3.5); set NTRIG=0 and go to Aj;
      if not, set 'FS'=0 (FS is a trigger abbreviated for feasible solution);
      test  $\bar{x}_i$  (≡  $\alpha_{i, n+1}$ ) also with reference to chapter 4), i ∈ I:
      if  $\bar{x}_i < 0$  for some i, test  $\bar{r}_{ij}$ , j ∈ J:
          if  $\bar{r}_{ij} > 0$  for all j, non-feasible problem (refer (3.3.2)) and the optimal solution does not exist.
          otherwise go to A2.1.2.1;
      if not, set FS=1 (to mark for a feasible solution);
      test Zj, j ∈ J:
      if Zj > 0 for each j, test 'FS':
          if FS ≠ 0, print optimal value, solution point and shadow prices - output.
  
```

A1122
A11221
A11222

A11
A111
A112
A1121
A11211
A11212

A12
A121
A122
A1221
A1222

A2
A21
A211
A2111
A2112
A212
A2121
A21211
A212111

```

A212112                otherwise, find  $\eta$  ( $=i \in I$ ) from (3.2.12)/(3.2.14) - exit criterion, class S;
                        and find  $\gamma$  ( $=j \in I_2$ ) from (3.2.15) - entry criterion, class S; pivotal element is  $\bar{r}_{\eta\gamma}$ 
                        go to A.3;
A21212                if not, test  $\delta_i, i \in I_2$ :
A212121                if  $\delta_i > 0$  for each  $i$ , unbounded problem - refer (3.3.1(ii)), the optimum solution does not exist.
A212122                otherwise, find  $\eta$  ( $=i \in I_2$ ) from (3.1.21) - exit criterion, class R;
                        and find  $\gamma$  ( $=j \in I_1$ ) from (3.2.6) or (3.2.7)/(3.2.9) as the case may be -
                        entry criterion, class R; pivotal element is  $\bar{r}_{\eta\gamma}$ ;
A3      test 'trig':
A31     if 'trig'  $\neq 0$ , the pivotal transformations are given by (4.2.21), go to A.1.1;
A32     otherwise, the pivotal transformations are given by (3.2.17)-(3.2.18), go to A.2.1;

```

23. NTRIG is set to 0, 1 or 2 (as the case may be) by reading a parameter card that has the appropriate value of NTRIG punched in it.
24. The number of equalities ($=m_2$) is counted at the time of reading the data of the given l.p. problem and is stored suitably for testing purposes as required subsequently by the algorithm. This and other minor details are not shown in the algorithm above.

Chapter 6

APPLICATION OF THE B.H.P. TO THE SIMPLEX METHOD OF LINEAR PROGRAMMING.

6.1 The two phases. Let us consider the l.p. problem as formulated in section 2.1 of chapter 2. As we know, the Simplex method works in two phases. Phase 1 determines a basic feasible solution (satisfying the equality constraints) and phase 2 then leads from the solution so obtained to the optimal solution. In considering the application of the B.H.P. to the Simplex method we will, as explained in chapter 4, treat first the equalities independently of the inequalities. For this purpose we consider the application of either (a) one of the two techniques of chapter 4 or alternatively (b) the artificial variable technique of the Simplex method. If the latter were to be utilised then, as stated, we restrict its application to the equality constraints alone. The auxiliary function, as in the Simplex method, is constructed and the usual Simplex rules for the entry and exit criterion are followed. At the end of the amended phase 1 as above we obtain a feasible solution point satisfying the equality constraints. The transformations carried out on the inequality constraints and the functional using the formulae (4.3.1)-(4.3.2) of chapter 4 and

$$\left. \begin{aligned} \bar{Z}_j &= Z_j - \sum_{k=1}^{n_1} Z_{g_k} \bar{\alpha}_{h_k, j} \\ \bar{F} &= F - \sum_{k=1}^{n_1} Z_{g_k} \bar{\alpha}_{h_k, n+1} \end{aligned} \right\}, j \in J \quad (6.1.1)$$

respectively yield us the enlarged linear programming problem with only inequality constraints and in which all the equalities are satisfied. This problem is in the same format as envisaged in the I/F tableau obtained by considering the techniques in (a) above (refer chapter 4).

The solution point for the enlarged problem may, however, not necessarily be a feasible one; in fact, we would for generality presume that it is an infeasible point. We next proceed (as discussed in chapter 3 and 4) to convert the type I ' \geq ' type constraints to type II ' \leq ' constraints and then add a positive slack variable to each constraint so as to obtain the B.H.P. tableau of (3.1.12). We are now ready to institute the extended algorithm for the Simplex method as explained in the following section.

6.2 The extended algorithm. class $R; ((\exists j \in J) (Z_j < 0))$.

6.2.1 Phase 0. It pertains to decision rules for the class R . It is termed phase 0 so as to distinguish it from phases 1 and 2 of the Simplex method.

6.2.1.1 Criterion for a variable to leave the basis. This corresponds to the selection of the η hyperplane ($= i \in I$) according to (3.1.21), chapter 3, class R .

6.2.1.2 Criterion for a variable to enter the basis. This corresponds to the selection of the axis γ ($= j \in J$) according to either the rule R2.1, (3.2.6) or the rule R2.2, (3.2.7)/(3.2.9) of chapter 3, as the case may be.

6.2.2 New tableau and the evaluations. Perform the usual Gaussian

eliminational transformations, using formulae (3.2.17)-(3.2.18) to obtain the new tableau. We then evaluate the sign conditions of $Z_j, j \in J$ and $x_i, i \in I$; the following four mutually and collectively exhaustive cases exist which direct the next move in the algorithm. These are

(i) if $((\forall j \in J) (Z_j \geq 0) \& (\forall i \in I) (x_i \geq 0))$

then the optimal solution has been found;

(ii) if $((\exists j \in J) (Z_j < 0) \& (\exists i \in I) (x_i < 0))$

then, if ΔF (given by (3.2.3), chapter 3) is non-negative proceed to 6.2.1, phase 0. above;

otherwise proceed to 6.2.3, phase 1 below. The infeasible point now obtained will, in general, yield (on application of phase 1 of the Simplex method) a feasible point which is nearer to optimum point than the feasible point that would have been obtained from the initially available infeasible point. Refer example 1, section 6.3, that illustrates this point.

(iii) if $((\exists j \in J) (Z_j < 0) \& (\forall i \in I) (x_i \geq 0))$ then a feasible

point has been obtained (this feasible point too would, in general, be nearer to the optimum point than the one obtained from the initially available infeasible point).

Refer example 2 of section 6.3.

(iv) if $((\forall j \in J) (Z_j \geq 0) \& (\exists i \in I) (x_i < 0))$

then the optimal solution is obtained by the application

of either the class S decision rules of B.H.P. Method (refer (3.2.11) chapter 3) or the Dual Simplex Method (Lemke [56]).

6.2.3 Phases 1 and 2.

(i) Phase 1. Since, in considering the application of the Simplex method, the (right hand side) elements of the vector P_0 must be non-negative we first change the signs of all the elements of the rows corresponding to the negative (basic) variables $x_i, i \in I$.

(It may be noted that the positive slack variables corresponding to the negative basis variables get thus converted to negative surplus variables by following the above step). Next we proceed as in the Simplex Method, construct the auxiliary function and choose the pivotal element according to the entry and exit criteria²⁵ of phase 1.

(ii) Phase 2. The criteria²⁵ for the selection of the η and τ variables remain as in this phase of the Simplex Method.

It may be noted that once we enter either phase 1 or phase 2 of the Simplex Method we would not leave the latter. The existence of the non-feasible or unbounded solution is checked according to Simplex criteria for these situations. The optimal criterion of the B.H.P. and the Simplex methods are identical.

6.2.3.1 Degeneracy. The B.H.P. method could contribute significantly to the resolution of degeneracy (or ties) in selecting the pivotal row in situations where the hyperplanes for which the ties

²⁵Refer Dantzig [19] or any standard text book cited in the bibliography.

occur have at least one bounding hyperplane amongst them. (When this is not the case then though the degeneracy rules of B.H.P.M. could still be used by considering (3.4.3) or (3.4.6) for non-bounding hyperplanes (that is $\varepsilon_i > 0$), their significance in resolving degeneracy is not known). Thus, for example, Beale's cycling example no.3.7.2, chapter 3 requires, even after taking recourse to the ε -perturbation technique six iterations whereas by following the former technique (used in the B.H.P.M.) the optimum solution is obtained in two iterations. Moreover, as we know, the cycling phenomenon in practical problems, never occurs, but the ties or degenerate situations do occur in almost every problem. It may be worth attempting to try to see the effect on the speed of the Simplex algorithm by following the B.H.P. technique referred to above in each situation when a tie or degeneracy occurs (in different iterations).

6.3 Illustrations:

6.3.1 Example 1. Consider the example
 maximise

$$F = x_1 + 2x_2$$

subject to

$$x_1 + x_2 \geq 3$$

$$2x_1 + x_2 \geq 4$$

$$-x_1 + 3x_2 \leq 12$$

$$5x_1 - x_2 \leq 10$$

and

$$x_1, x_2 \geq 0$$

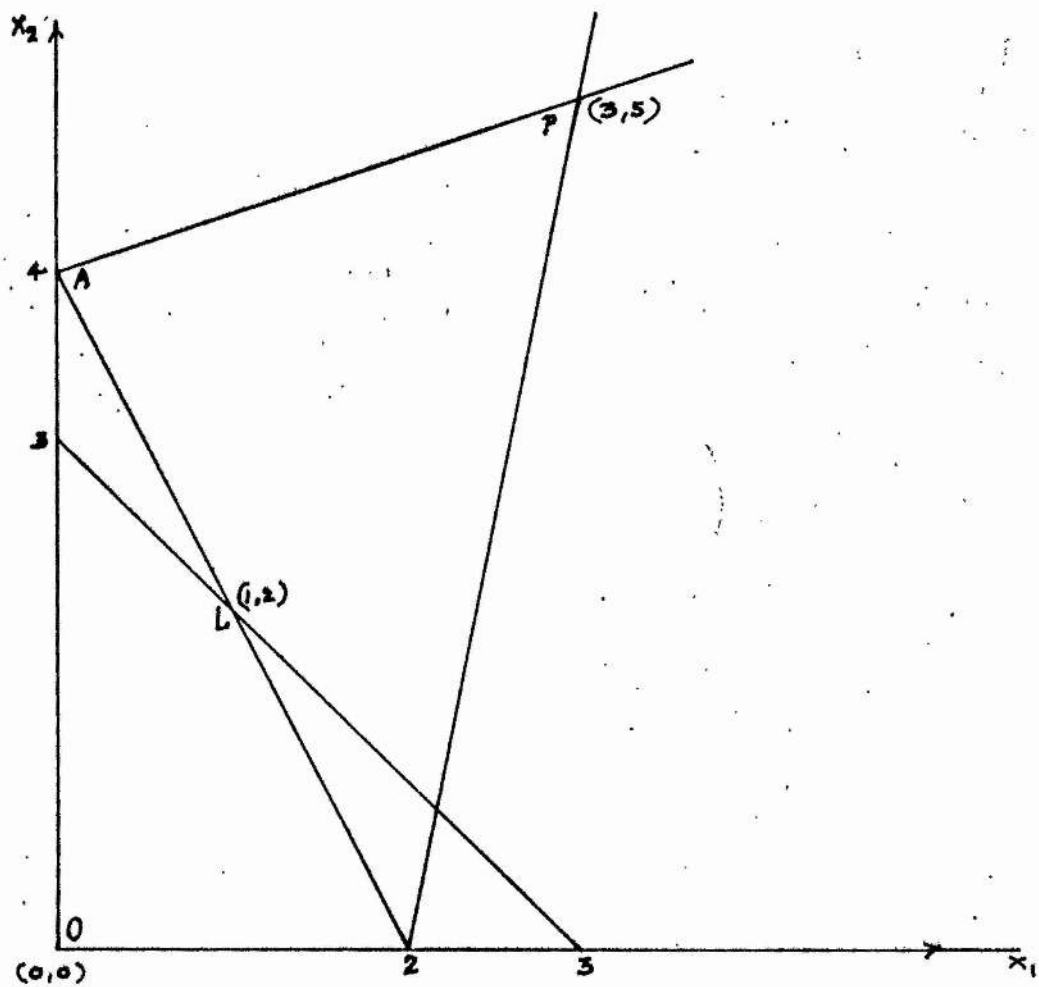


Fig. 6.1 - Diagram corresponding to example 1.

Table 61

Initial tableau for phase 0 of example 1.

i \ j									
		1	2	3	4	5	6		
C		1	2	0	0	0	0		
	\bar{C}	Basis	P_1	P_2	P_3	P_4	P_5	P_6	$P_0 (= \bar{X})$
1	0	P_3	-1	-1	1				-3
2	0	P_4	-2	-1		1			-4
3	0	P_5	-1	③			1		12
4	0	P_6	5	-1				1	10
Z_j			-1	-2	0	0	0	0	0 (=F)

Table62

Tableau after one iteration for phase 0/2 of example 1

i \ j									
		1	2	3	4	5	6		
C		1	2	0	0	0	0		
	\bar{C}	Basis	P_1	P_2	P_3	P_4	P_5	P_6	$P_0 (= \bar{X})$
1	0	P_3	-4/3		1		1/3		1
2	0	P_4	-7/3			1	1/3		0
3	2	P_2	-1/3	1			1/3		4
4	0	P_5	(14/3)				1/3	1	14
Z_j			-5/3	0	0	0	2/3	0	8 (=F)

Rewriting the above problem as envisaged in the B.H.P.M.;

maximise

$$F = x_1 + 2x_2 + 0x_3 + 0x_4 + 0x_5 + 0x_6$$

subject to

$$-x_1 - x_2 + x_3 = -3$$

$$-2x_1 - x_2 + x_4 = -4$$

$$-x_1 + 3x_2 + x_5 = 12$$

$$5x_1 - x_2 + x_6 = 10$$

and $x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$

The B.H.P.M. tableau is then as given in Table 6.1.

Phase 0:

From (3.1.21),

$$\text{Min } (12/5, 10/3)$$

is obtained for the hyperplane $i = \eta = 3$

And from (3.2.6), the axis by which to move to the bounding

hyperplane η is given by $\gamma = 2$. The pivotal element is thus

$\bar{b}_{32} = 3$ as shown by the circle around it, in the above tableau.

Performing the usual Gaussian eliminational transformation, the new tableau is obtained as shown in table 6.2.

The new point obtained is the basic feasible point A in figure 6.1.

Phase 2 of the Simplex method takes over at this point and yields

the optimal point P (3, 5, 7, 5) in one step, with the optimal

value of the objective function equal to 13. However, if

initially phase 1 of the Simplex Method was followed (instead of

Table 63

Initial Tableau for phase 0 of Example 2

i \ j			1	2	3	4	5	6	7	
	C		1	2	0	0	0	0	0	
	\bar{C}	Basis	P_1	P_2	P_3	P_4	P_5	P_6	P_7	$P_0 (= \bar{X})$
1	0	P_3	-1	-1	1					- 3
2	0	P_4	-2	-1		1				- 4
3	0	P_5	-1	③			1			12
4	0	P_6	5	-1				1		10
5	0	P_7	-5	-2					1	- 9
	Z_j		-1	-2	0	0	0	0	0	0(=F)

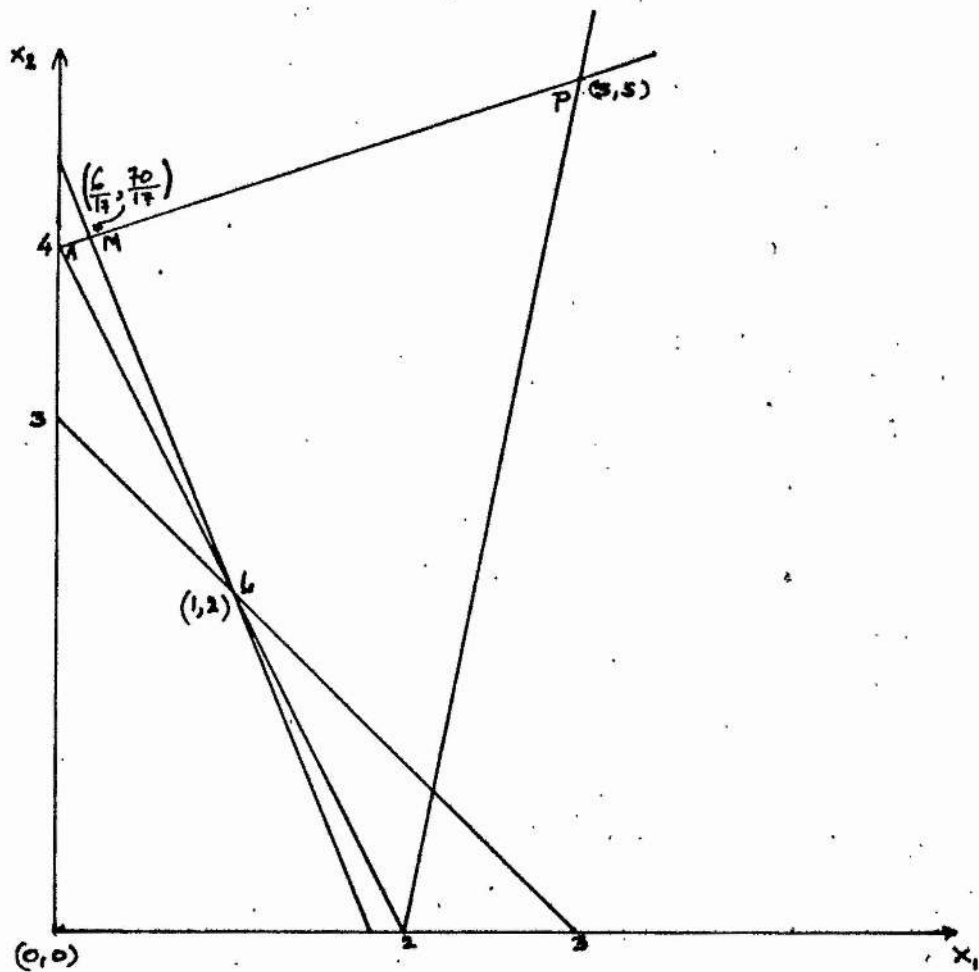


Fig. 6.2 - Diagram corresponding to example 2.

Phase 0) then it would take 2 iterations to arrive at the first basic feasible point L and subsequently it takes another two iterations to reach the optimal point P. Hence, we save two iterations (one each from the operations of the phase 1 and phase 2) if we follow phase 0 prior to the applications of the Simplex Method.

6.3.2 Example 2. Consider again the example 1 with an additional constraint

$$5x_1 + 2x_2 \geq 9$$

The first two B.H.P.M. tableaux are as given in Tables 6.3 and 6.4²⁶. The computations for finding the pivot are exactly the same as in Example 1. The new point obtained is still A; however, it is this time infeasible (refer figure 6.2.). If we may now for the sake of simplicity of exposition, let phase 1 of the Simplex Method take over²⁷, then in one step we reach the point M which is a basic feasible point; phase 2 of the Simplex Method next takes over and in one more step yields the optimal point P with optimal value equal to 13. We thus required three iterations to

26. The tables 6.3 and 6.4 are the same as in example 1 except for the additional row (last one) corresponding to the additional constraint in example 2 above.

27. If we do not let phase 1 take over at this point then the optimal point P is reached in this example in the very next step of phase 0; we, therefore, let phase 1 take over so as to illustrate the point that a 'better' infeasible point is reached by following phase 0.

Table 64

Tableau after one iteration for phase 0/2 of example 2

i \ j										
		1	2	3	4	5	6	7		
C		1	2	0	0	0	0	0		
\bar{C}	Basis	P_1	P_2	P_3	P_4	P_5	P_6	P_7	$P_0 (= \bar{X})$	
1	0	P_3	-4/3	1		1/3			1	
2	0	P_4	-7/3		1	1/3			0	
3	2	P_2	-1/3	1		1/3			4	
4	0	P_6	14/3			1/3	1		14	
5	0	P_7	-17/3			2/3		1	-2	
Z_j			-5/3	0	0	0	2/3	0	0	$g (= F)$

reach the optimal point P.

On the other hand, if we had initially applied phase 1 (instead of phase 0) directly then it takes as in Example 1 two iterations to reach the basic feasible point L and an additional two iterations to reach the optimal point P using phase 2. Apart from the fact that it takes more iterations, if we use the Simplex Method alone, to reach the optimal point, the main point to be observed here is that we arrive at a 'better' basic feasible point M by following phase 0 prior to phase 1 of the Simplex Method. As indicated above it should also be noted that if we had continued to follow the B.H.P.M. alone then the optimal point P would, in this example, be reached in just two iterations.

Chapter 7

COMPARATIVE STUDY OF THE SIMPLEX AND B.H.P. METHODS

In making a comparative study we have been restricted by the number of examples available to us. Most of the examples have been drawn from the text books, different computer reference manuals and private sources. The study, though of a limited nature, however, throws significant light on the power of the B.H.P. method over the Simplex. We have attempted to compare the methods by (i) the number of iterations and (ii) the time taken to solve the different sized problems by running the two computer (all-core) programs on 16 examples.

7.1 Number of iterations required by the B.H.P.M. vis-a-vis Simplex.

Columns 3 and 4 of table 7.1 provide the detailed information in this respect. As we observe, the B.H.P. method takes less iterations in most of the examples. In none of the examples presented did we find it to take more iterations as compared to Simplex (this may however not be true generally). Furthermore the number of iterations taken by the B.H.P. method appear to reduce on comparison with Simplex method as the problem size increases. It is not known if this would generally be true. Extensive application of the method could only reveal the general situation in this respect.

7.2 Time taken by the two methods.

As could be expected the time taken by the B.H.P. method in a single iteration would, in general, be more than that required by the Simplex method. A comparison of the two 'times' given in

columns 5 and 6 of table 7.1 also points to this fact. For the (single) large problem number 16 the time taken by the B.H.P.M. is appreciably less. Again it is not known if this would generally be true. We need to run the method on very large problems to investigate this aspect fully. In general it appears, however, that the time taken by the B.H.P.M. in all cases reported in table 7.1 is either less or is comparable with the time taken by the Simplex method. It may be stated here that the total time taken by the B.H.P. method could be reduced by (i) improving the existing B.H.P.M. computer program and (ii) incorporating the techniques discussed in sections 4.2 and 4.3, chapter 4. The single iteration time though would still generally remain more than that required by the Simplex method; the B.H.P.M. however has the advantage of detecting, in general, the 'unbounded'/'non-feasible' state of the given l.p. problem earlier than the Simplex method. This is because we, in each iteration, consider (i) the whole sub-space \mathcal{A}^j ($j \in J_1 \subset J$) and not just one column (of this sub-space) that is the one corresponding to $\max |z_j|, j \in J_1$ (as with Simplex) for unboundedness and (ii) each infeasible constraint ($i \in I_1 \subset I$) for non-feasibility (till we find at least one feasible solution point).

Table.7.1

Serial number.	Size of the problem. ($m \times n$)	Number of iterations taken by the two methods,		Time taken by the two methods in secs.,	
		B.H.P.	Simplex.	B.H.P.	Simplex.
1	2	3	4	5	6
1	4 x 4	2	6	3.6	3.5
2	4 x 5	4	4	4.3	3.5
3	4 x 5	2	4	4.4	4.2
4	4 x 6	3	5	4.8	4.3
5	4 x 6	2	2	4.6	3.5
6	5 x 7	3	9	5.8	5.9
7	5 x 7	4	12	6.0	6.4
8	7 x 6	3	4	7.2	7.1
9	8 x 5	4	10	5.6	6.1
10	8 x 5	3	10	5.5	6.1
11	8 x 5	4	7	5.9	6.2
12	9 x 6	1	4	6.4	6.8
13	15 x 8	3	12	12.1	14.0
14	19 x 21	18	20	18.8	19.3
15	20 x 3	2	11	10.1	12.5
16	54 x 69	49	79	57.0	71.6

Note 1. The size of the problem given in column 2 above does not include the slack (and surplus in case of Simplex) variables added to the constraints.

2. The times given in column 4 are obtained by running the computer program for the 'standard' Simplex method; both the times in columns 3 and 4 are the execution times ^(inclusive of input and output times) obtained by using a sub-routine KLOCK [87] on the I.B.M. 360/44 computer.

SUGGESTIONS FOR FURTHER RESEARCH

8.1^v Extensions to quadratic programming. Consider the quadratic programming²⁸ problem given by

$$\text{maximise } F(x) = Cx - \frac{1}{2} x^T D x \quad , \quad (8.1.1)$$

subject to linear inequalities

$$Ax \leq P_0 \quad , \quad (8.1.2)$$

and

$$x \geq 0 \quad , \quad (8.1.3)$$

where the vectors C , x , P_0 and the matrices A and D are respectively given by

$$C = |c_1 \ c_2 \ \dots \ c_n| \quad , \quad (8.1.4)$$

$$x = \begin{vmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{vmatrix} \quad , \quad (8.1.5)$$

$$P_0 = \begin{vmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{vmatrix} \quad , \quad (8.1.6)$$

$$A = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{vmatrix} \quad , \quad (8.1.7)$$

²⁸. The quadratic programming problem will henceforth be abbreviated as 'q.p. problem'.

$$D = \begin{vmatrix} d_{11} & d_{12} & \cdot & \cdot & d_{1n} \\ d_{21} & d_{22} & \cdot & \cdot & d_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ d_{n1} & d_{n2} & \cdot & \cdot & d_{nn} \end{vmatrix} \quad (8.1.8)$$

Further D is assumed to be a symmetric positive semi-definite matrix. Introducing the m non-negative slack variables X_s to the inequalities (8.1.2) the above q.p. problem could be rewritten as,

$$\text{maximise} \quad \phi(Y) = f Y - \frac{1}{2} Y^T Q Y, \quad (8.1.9 \text{ (i)})$$

$$\text{subject to} \quad B Y = P_0, \quad (8.1.9 \text{ (ii)})$$

$$\text{and} \quad Y \geq 0, \quad (8.1.9 \text{ (iii)})$$

where the $(n+m)$ vectors Y and f and the $(k$ by $(n+m))$ matrix B and $((n+m)$ by $(n+m))$ matrix Q are respectively given by

$$Y = \begin{vmatrix} X \\ X_s \end{vmatrix} \quad (8.1.10)$$

$$f = \begin{vmatrix} C & 0 \end{vmatrix} \quad (8.1.11)$$

$$B = \begin{vmatrix} A & I \end{vmatrix} \quad (8.1.12)$$

and

$$Q = \begin{vmatrix} D & 0 \\ 0 & 0 \end{vmatrix} \quad (8.1.13)$$

Let us now consider the Kuhn-Tucker Theorem [90]. It states that the vector Y is a solution to the q.p. problem (8.1.9(i)-(iii)) if and only if Y is feasible and there exists a $(n+m)$ vector

$$U = \begin{vmatrix} Y \\ W \end{vmatrix} \quad (8.1.14)$$

such that

$$U = QY + B^T W - f^T \geq 0 \quad (8.1.15)$$

and

$$U^T Y = 0 \quad (8.1.16)$$

The condition (8.1.16) implies that at most $(n+m)$ elements out of $(2n+2m)$ elements are non-zero which, in turn, means that we need to examine only basic solutions to the system (8.1.9(i)-(iii)) in our search for the optimum to the given q.p. problem (Barankin and Dorfman [4]). Thus in moving from one basic solution point to the next we enforce the condition that if a Y variable is in the basis then the corresponding U variable is not in the basis (the tableau corresponding to a basic solution point, with the above condition adhered to, is said to be in the 'standard form', Beale [6]).

If, therefore, a basic feasible solution is obtained such that the tableau corresponding to this basic solution is in 'standard form' and $U \geq 0$, then we have found the optimum point.

Let us now reconsider the optimal function (8.1.9(i)).

Substituting for QY given by (8.1.15) we obtain [78]

$$\begin{aligned} \phi(Y) &= f^T Y - \frac{1}{2} Y^T (U - B^T W + f^T) \\ &= \frac{1}{2} f^T Y - \frac{1}{2} U^T Y + \frac{1}{2} W^T B Y, \end{aligned}$$

which after substitution for BY from (8.1.9(ii)) gives

$$\begin{aligned} &= \frac{1}{2} f Y - \frac{1}{2} U^T Y + \frac{1}{2} P_0^T W, \\ &= \frac{1}{2} (f Y + P_0^T W) \text{ in view of (8.1.16);} \end{aligned} \quad (8.1.17)$$

that is, presuming that we consider the tableau in 'standard form'.

Thus if

$$\Phi = 2 \phi(Y), \quad (8.1.18)$$

we have

$$\Phi = f Y + P_0^T W. \quad (8.1.19)$$

Equation (8.1.15), (8.1.9(ii)) and (8.1.19) could, in view of the special structure of matrices Q, B and f , be rewritten as

$$-DX - A^T W + C^T \leq 0, \quad (8.1.20)$$

$$AX \leq P_0, \quad (8.1.21)$$

and

$$CX + P_0^T W = F^*, \quad (8.1.22)$$

where

$$F^* = 2F(X). \quad (8.1.23)$$

The system (8.1.20)-(8.1.22) is the usual l.p. problem and could compactly be written as

$$\left. \begin{array}{l} \text{maximise} \quad H \xi, \\ \text{subject to} \quad G \xi \leq R, \\ \text{and} \quad \xi \geq 0, \end{array} \right\} \quad (8.1.24)$$

where H, G, R and ξ are given by

$$H = \left| C \ P_0^T \right|, \quad (8.1.25)$$

$$G = \begin{vmatrix} -D & -A^T \\ A & 0 \end{vmatrix} \quad (8.1.26)$$

$$R = \begin{vmatrix} -C^T \\ P_0 \end{vmatrix} \quad (8.1.27)$$

and

$$\xi = \begin{vmatrix} x \\ w \end{vmatrix} \quad (8.1.28)$$

The dual l.p. problem to the (primal) system (8.1.24) is then given by,

$$\text{minimise} \quad R^T \xi \quad (8.1.29)$$

$$\text{subject to} \quad G^T \xi \geq H^T \quad (8.1.30)$$

$$\text{and} \quad \xi \geq 0 \quad (8.1.31)$$

Converting (8.1.29) to a maximising problem and the type I (\geq) inequalities in (8.1.30) to type II (\leq), the 'dual' could be rewritten as

$$\begin{aligned} \text{maximise} \quad & -R^T \xi \\ \text{subject to} \quad & -G^T \xi \leq H^T \end{aligned} \quad (8.1.32)$$

$$\text{and} \quad \xi \geq 0$$

We are now ready to apply the B.H.P.M. with restrictions that (i) the tableau is maintained in each iteration in the 'standard form' and that (ii) the hyperplane (pivotal row) selected in the immediately preceding iteration is not considered for selection in the current iteration. The latter condition may be required to be imposed in the q.p. algorithm for the B.H.P.M. on account of the former condition (so as to avoid a possible case of cycling).

Since in the primal problem the optimal solution is

obtained when we reach a basic feasible solution, the corresponding situation in the dual case is achieved when the coefficients of the functional Z given by

$$Z = R \quad (8.1.33)$$

are non-negative.

8.2 Accuracy in linear programming.

8.2.1 Modifications to reinversion process. Let us consider the optimal solution obtained by the application of the B.H.P.M. (or the Simplex or any other techniques) to a given l.p. problem. We know that certain constraints in the optimal point are satisfied exactly, and that the pivotal elements lie on these constraints; let these be called the pivotal constraints. Utilising this information we reconstruct the original l.p. problem wherein the pivotal constraints are equalities and the remaining inequalities. Since we also know the pivotal elements lying on each pivotal constraint we first obtain the solution to the equality (pivotal) constraints by performing the Gaussian eliminational transformations on these pivotal elements. Next we determine the transformations to the vector $P_v (= \bar{\beta}_{i,n+1})$ corresponding to the inequality constraints from (4.3.2) of section 4.3, chapter 4. The coefficients of the functional Z will (in all probability) be non-negative to indicate that the optimal point, given by

$$\begin{pmatrix} \bar{\beta}_{i,n+1} \\ \bar{\alpha}_{i,n+1} \end{pmatrix} \quad (8.2.1)$$

has been reached. The (new) optimal point, if different from the earlier one, would satisfy the pivotal constraints 'more'

exactly. This is on account of considerable reduction in the computations required otherwise. The (new) optimal tableau, if desired, can be obtained from (4.3.1), chapter 4. The above procedure is essentially the reinversion process [63] modified by the techniques developed in chapter 4. For further reducing the effects of rounding errors we may consider the extensions discussed generally in the section below.

8.2.2 Modifications to the equality solution techniques. Let us reconsider the techniques of section 4.2, chapter 4. We may along with finding the solution to the equality constraints obtain the inverse²⁹ of the matrix corresponding to the pivotal elements and check for its accuracy by determining its product with the original matrix; the discrepancy between the latter and the unit matrix shows the extent of inaccuracy in the results.

Thus if the rank of the equality submatrix W is n_1 ($\leq r$) then let \hat{W} denote the leading non-singular matrix of order $n_1 \times n_1$ obtained, as is always possible, by rearranging the equations and unknowns (variables). Let S_0 denote the inverse of the submatrix \hat{W} obtained as above (or by any other process of inversion) and let M_0 ($= \lambda_{ij}$) denote the discrepancy

$$M_0 = I - \hat{W} S_0, \quad \|M_0\| \leq \rho < 1 \quad (8.2.2)$$

29. The inverse could easily be obtained in the process of obtaining solution to the equality constraints by appending an artificial (unit) vector (as in the Simplex technique) to each of the n_2 equality constraints. The inverse corresponds to the transformations to the artificial vectors.

where $\|M_0\|$, considering the first matrix norm³⁰ (sections 5, 13 [241]), is given by

$$\max_i \sum_{k=1}^{n_1} |\phi_{ik}|, \quad i = 1, 2, \dots, n_1. \quad (8.2.3)$$

The accuracy of the elements of the inverse matrix \hat{W}^{-1} ($= \theta_{ij}$, an $n_1 \times n_1$ matrix) may then be increased to as high a degree as desired by following the iterative process ([89]; also sec. 13 in [241])

$$S_q = S_{q-1} + S_{q-1} (E - \hat{W} S_{q-1}); \quad (8.2.4)$$

where

$$S_q = S_{q-1} (E + M_{q-1}) \quad (8.2.5)$$

$$M_q = E - \hat{W} S_q \quad (8.2.6)$$

(It can be proved [241] that

$$M_q = M_0^{2^q}, \quad (8.2.7)$$

(8.2.6) then reduces to

$$S_q = \hat{W}^{-1} (E - M_0^{2^q}), \quad (8.2.8)$$

which shows that S_q approaches \hat{W}^{-1} within the accuracy desired.)

Having thus obtained a more accurate value for the inverse matrix \hat{W}^{-1} , we utilise it to obtain the final equality tableau of Section 4.2.3 by operating the former on (a) those columns

30. Hotelling, however, considered the norm given by

$$\sqrt{\sum_{i=1, j=1}^{n_1} |\phi_{ij}|^2}$$

of the original submatrix w that have nullity $r-n_1$ and (b) the column vector P_w ; thence the elements of the final equality tableau are given by

$$\bar{\alpha}_{ij} = \sum_{k=1}^{n_1} \theta_{ik} \alpha_{kj}, \quad i \in I_w, j \in J_1 \quad (8.2.9)$$

and

$$\bar{\alpha}_{i,n+1} = \sum_{k=1}^{n_1} \theta_{ik} \alpha_{k,n+1}, \quad i \in I_w \quad (8.2.10)$$

where $J_2 (\subset J_{Ea})$ is the set of indices j that correspond to (a) above, refer Section 4.4 of chapter 4. It is clear that the submatrix of the final $(W|P_w)$ tableau corresponding to w is a unit matrix and hence these elements are not computed.

The elements of the I/F tableau are then obtained from (4.3.1 - 4.3.5) as discussed in Section 4.3 of chapter 4. The coefficients thus obtained in the final equality and I/F tables will be more accurate than the ones obtained by following the techniques of chapter 4.

BIBLIOGRAPHY

- 1 Ackoff, R.L., (Ed.), Progress in Operations Research, Vol 1, ORSA No. 5, John Wiley & Sons, Inc., New York, 108-201 pp.
- 2 Balas, E., "Un algorithme additif pour la resolution des programmes lineaires en variables bivalents," Compt. Rend. Acad. Sci., Vol. 258, 1964, pp. 3817-3820.
- 3 -----, Extension de l'algorithme additif a la programmation en nombres entiers et a la programmation nonlineaire, Compt. Rend. Acad. Sci., Vol. 258, May 25, 1964, pp. 5136-5139.
- 4 Barankin, E.W., and R. Dorfman, "On Quadratic Programming," University of California Publications in Statistics, Vol. 2, No. 13, University of California Press, Berkeley, California, 1958, pp. 285-318.
- 5 Beale, E.M.L., "Cycling in the Dual Simplex Algorithm," Naval Res. Logist. Quart., Vol. 2, No. 4, December, 1955, pp. 269-276.
- 6 -----, "On Quadratic Programming," Naval Res. Logist. Quart., Vol. 6, No. 3, September, 1959, pp. 227-243.
- 7 Beale, E.M.L., P.A.B. Hughes, and R.E. Small, "Experience in using a decomposition Program," The Computer Journal, Vol. 8, April 1965 to Jan. 1966, pp. 13-18.
- 8 Charnes, A., "Optimality and Degeneracy in Linear Programming," Econometrica, Vol. 20, No. 2, April, 1952, pp. 160-170. Published in Operational Res. Quart., Vol. 3, No. 3, September, 1952, p. 54 (abstract).
- 9 Charnes, A., and W.W. Cooper, "Generalizations of the Warehousing Model," Operational Res. Quart., Vol. 6, 1955, pp. 131-172.
- 10 -----, "The Stepping Stone Method of Explaining Linear Programming Calculations in Transportation Problems," Management Sci., Vol. 1, No. 1, 1954, pp. 49-69.
- 11 -----, "Such Solutions Are Very Little Solved," Vol. 3, No. 4, 1955, pp. 345-346.
- 12 Charnes, A., W.W. Cooper, and A. Henderson, An Introduction to Linear Programming, John Wiley & Sons, Inc., New York, 1953. 74 pp.
- 13 Chung, An-Min, Linear Programming, C.E. Merrill, Columbus, Ohio, 1963.
- 14 Curtes, H.A., "Use of Decomposition Theory in the Solution of the State Assignment Problem of Sequential Machines, J. Assoc. Computing Mach., July, 1963, p. 386.
- 15 Dahl, Ole-John, "Linear Programming on the MERCURY Computer, The Multiplex Method," Internal Report F-0375, Norwegian Defense Research Establishment, Kjeller, Lillestrom, Norway, March 1959.

- 16 Dantzig, George B., 1948-1. "Programming in a Linear Structure,"
Comptroller, USAF, Washington, D.C., February, 1948.
- 16(a) -----, "A Proof of the Equivalence of the Programming Problem
and the Game Problem," in T.C. Koopmans (ed.), Activity
Analysis of Production and Allocation, John Wiley &
Sons, Inc., New York, 1951, pp. 330-335.
- 17 -----, "Maximization of a Linear Function of Variables
Subject to Linear Inequalities," in T.C. Koopmans (ed.),
Activity Analysis of Production and Allocation, John
Wiley & Sons, Inc., New York, 1951, pp. 339-347.
- 18 -----, Notes on Linear Programming: Part XI, Composite
Simplex-Dual Simplex Algorithm-I, The RAND Corporation,
Research Memorandum RM-1274, April 26, 1954, 15 pp.
Also The RAND Corporation, Paper P-516.
- 19 -----, "Linear Programming and Extensions," Princeton
University Press, New Jersey, 1963, 632 pp.
- 20 Dantzig, George B., Lester R. Ford, Jr., and Delbert R.
Fulkerson, "A Primal-Dual Algorithm for Linear Programs,"
in H.W. Kuhn and A.W. Tucker (eds.), Linear Inequalities
and Related Systems, Annals of Mathematics Study,
No. 38, Princeton University Press, Princeton, New
Jersey, 1956, pp. 171-181.
- 21 Dantzig, George B., and P. Wolfe, "The Decomposition Algorithm
for Linear Programming," *Econometrica*, Vol. 29, No. 4,
October, 1961.
- 22 Dorn, W.S., "On Lagrange Multipliers and Inequalities,"
Operations Research, Vol. 9, No. 1, 1961, pp. 95-104.
- 23 Everett, H., "Generalized Lagrange Multiplier Method for
Solving Problems of Optimum Allocation of Resources,"
Operations Research, Vol. 11, 1963, pp. 399-417.
- 24 Faddaeva, V.N., *Computational Methods of linear Algebra*
(translated by Curtis D. Benster), Dover Publications,
Inc., New York, 1959, 252 pp.
- 25 Falk, J.E., "Langrangian Multipliers and Non-Linear Program-
ming," *J. Math. Anal. Appl.*, Vol. 19, No. 1, July 1967.
- 26 Ferguson, R.O., and L.F. Sargent, *Linear Programming: Funda-
mentals and Applications*, McGraw-Hill Book Company,
Mc, New York, 1958, 342 pp; (chapters 3 and 6 refer to
the MODI and the Ratio Analysis Method respectively).
- 27 Ford, Lester R., Jr., and Delbert R. Fulkerson, "A Primal-
dual Algorithm for the Capacitated Hitchcock Problem,"
The RAND Corporation, Paper P-827, March 23, 1956; also
The RAND Corporation, Research Memorandum RM-1798,
September 25, 1956. Published in *Naval Res. Logist.*
Quart., Vol. 4, No. 1, March 1957, pp. 47-54.

- 28 Fourier, Jean Baptiste Joseph, "Solution d'une question
 particuliere du calcul des inegalites," 1826, and
 extracts from "Histoire de l'Academie," 1823, 1824,
 Oeuvres II, pp. 317-328.
- 29 Frisch, Ragnar, "Multiplex Method of Linear Programming,"
 Sankhya, Vol. 18, 1955, pp. 329-362.
- 30 -----, "Multiplex Method for Linear Programming," Memorand-
 um, Institute of Economics, University of Oslo,
 Norway, Sept. 1958, pp.
- 31 -----, Principles of Linear Programming (with Particular
 References to the Double Gradient Form of the Lograth-
 mic Potential Method, Memorandum, Institute of Economics,
 University of Oslo, Norway, 1954.
- 32 -----, "Mixed Linear and Quadratic Programming by the
 Multiplex Method," in Hugo Hegeland (ed.), Money,
 Growth, and Methodology and other Essays in Economics,
 Cwk Gleerup Lund, Sweden, March 1961, pp. 373-392.
- 33 Gale, David, H.W. Kuhn, and A.W. Tucker, "Linear Programming
 and the Theory of Games," in T.C. Koopmans (ed.),
 Activity Analysis of Production and Allocation, John
 Wiley & Sons, Inc., New York, 1951. Also the RAND
 Corporation, Report R-193, June 1951.
- 34 Garvin, Walter W., Introduction to Linear Programming,
 McGraw-Hill Book Company, Inc., New York, 1960, 281 pp.
- 35 Gass, Saul I, Linear Programming: Methods and Applications,
 McGraw-Hill Book Company, Inc., New York, 1958, 223 pp.
- 36 Graves, Robert L., and Philip Wolfe (eds.), Recent Advances
 in Mathematical Programming, McGraw-Hill Book Company,
 Inc., New York, 1963.
- 37 Hadley, G., Nonlinear and Dynamic Programming, Addison-Wesley
 Publishing Co., Reading, Massachusetts, Inc., 1964.
- 38 -----, Linear Programming, Addison-Wesley Publishing Co.,
 Inc., 1962.
- 39 Henderson, A., and R. Schlaifer, "Mathematical Programming,"
 Harvard Business Rev., Vol. 32, May-June, 1954, pp.73-
 100.
- 40 Hitchcock, Frank L., "Distribution of a Product from Several
 Sources to Numerous Localities," J. Math. Phys., Vol.
 20, 1941, pp. 224-230.
- 41 Hoffman, A.J., "Cycling in the Simplex Algorithm," National
 Bureau of Standards, Report No. 2974, December 16,
 1953. 7 pp.
- 41(a) -----, "How to Solve a Linear Programming Problem," in
 H. Antosiewicz (ed.), Proceedings of the Second Sympos-
 ium in Linear Programming, Vols. 1, and 2, Directorate
 of Management Analysis, DCS/Comptroller, U.S. Air
 Force, Washington, D.C., Jan., 1955, pp. 397-424.

- 42 Hoffman, A.J., M. Mannos, D. Sokolowsky, and N. Wiegmann, "Computational Experience in Solving Linear Programs," J. Soc. Indust. Appl. Math., Vol. 1, No. 1, 1953, pp. 17-33.
- 43 Houthakker, H.S., "The Capacity Method of Quadratic Programming," in P. Wolfe (ed.), The RAND Symposium on Mathematical Programming, The RAND Corporation, Report R-351, March 16-20, 1959.
- 44 I.B.M. Manuals ; I.B.M. Manual, E20-817-0, I.B.M. World Trade Corporation, 821, United Nations Plaza, New York 10017, pp. 23-31.
- 45 -----, I.B.M. 1130 Linear Programming - Mathematical Optimisation Sub-routine System, Program Reference Manual H20-0345-0, pp. 3-11.
- 46 -----, I.B.M. 360 Fortran Linear Programming System, 360 D-15.2.006, pp. 5-9.
- 47 Kantorovich, L.V., "Mathematical Methods in the Organization and Planning of Production," Publication House of the Leningrad State University, 1939. 68 pp. Translated in Management Sci., Vol. 6, 1960, pp. 366-422.
- 48 -----, "On the Translocation of Masses," Compt. Rend. Acad. Sci., U.R.S.S., 37, 1942, pp. 199-201.
- 49 Kantorovich, L.V., and M.K. Gavurin, "The Application of Mathematical Methods to Problems of Freight Flow Analysis," (translation), Akademii Nauk SSSR, 1949.
- 50 Klein, B., "Direct Use of External Principles in Solving Certain Optimising Problems Involving Inequalities," Operations Research, Vol. 3, 1955, pp. 168-175.
- 51 Koopmans, T.C., "Optimum Utilization of the Transportation System," Proceedings of the International Statistical Conferences, 1947, Washington, D.C. (Vol. 5 reprinted as Supplement to Econometrica, Vol. 17, 1949).
- 52 -----, (ed.) Activity Analysis of Production and Allocation, John Wiley & Sons, Inc., New York, 1951, 404 pp.
- 53 Kuhn, H.W., "The Hungarian Method for the Assignment Problem," Naval Res. Logist. Quart., Col. 2, 1955, pp. 83-97.
- 54 Kuhn, H.W., and A.W. Tucker (eds.), Linear Inequalities and Related Systems, Annals of Mathematics Study No. 38, Princeton University Press, Princeton, New Jersey, 1956, 322 pp.
- 55 Land, A.H., and A.G. Doig, "An Automatic Method of Solving Discrete Programming Problems," Econometrica, Vol. 28, pp. 497-520.
- 56 Lemke, C.E., "The Dual Method of Solving the Linear Programming Problem," Naval Res. Logist. Quart., Vol. 1, 1954, pp. 36-47.

- 57 Markowitz, Harry M., "The Optimization of a Quadratic Function Subject to Linear Constraints," The RAND Corporation, Research Memorandum RM-1438, February 21, 1955; also The RAND Corporation, Paper P-637, June 27, 1955. Published in Naval Res. Logist. Quart., Vol. 3, 1956.
- 57(a) Mehndiratta, S.L., "General Symmetric Dual Programs," Operations Research, Vol. 14, No. 1, 1966, pp. 164-172.
- 58 Motzkin, T.S., and I.J. Schoenberg, "The Relaxation Method for Linear Inequalities," Canad. J. Math., Vol. 6, 1954, pp. 393-404.
- 59 Orchard-Hays, W., A Composite Simplex Algorithm - II, The RAND Corporation, Research Memorandum RM-1275, May, 1954.
- 60 Orden, Alex, and L. Goldstein (eds.), Symposium on Linear Inequalities and Programming, Project SCOOP, No. 10, Planning Research Division, Director of Management Analysis Service, Comptroller, USAF, Washington, D.C., April 1952.
- 61 Poussin, M. Ch. J. de la Vallee, "Sur la methode de l'approximation minimum," Ann. Soc. Sci. de Bruxelles, 35, 1911, pp. 1-16.
- 62 Perlis, S., Theory of Matrices, Addison-Wesley Press, Inc., Cambridge, Mass., 1952, 237 pp.
- 63 Rall, L.B., (ed.), Error in Digital Computation, Vol. II, John Wiley & Sons, New York, 1965, pp. 271-284.
- 63(a) Rosen, J.B., "Primal Partition Programming for Block Diagonal Matrices," Computer Science Division School of Humanities and Sciences, Stanford University, Stanford, California, Technical Report No. 32, November, 1963.
- 64 Saaty, T.L., "The Number of Vertices of a Polyhedron," Amer. Math. Monthly, Vol. 65, No. 5, May, 1955, pp. 327-331.
- 65 Simmonard, M., Linear Programming (translated by W.S. Jewell), Prentice-Hall, Inc., New Jersey, 1966, 430 pp.
- 66 Slater, M., "Lagrange Multipliers Revisited," Cowles Commission Discussion Papers, Math. 403, November, 1950; The RAND Corporation, Research Memorandum RM-676, August, 1951.
- 67 Stigler, George J., "The Cost of Subsistence," J. Farm Econ., Vol. 27, No. 2, May, 1945, pp. 303-314.
- 68 Smythe, William R. (Jr.), and Lynwood A. Johnson, Introduction to Linear Programming, With Applications, Prentice-Hall, Inc., New Jersey, 1966, 221 pp.
- 69 Saksena, C.P., "Multiplex Method of Linear Programming," Memorandum, Institute of Economics, University of Oslo, Norway, May 1959.

- 70 Saksena, C.P., "Frisch's Multiplex Method," Lent Term Seminar, London School of Economics, University of London, 1960.
- 71 -----, "Frisch's Multiplex Method in Linear Programming and Electronic Computers," in J.N. Kapur, L.S. Srinath, and Holt Ashley (eds.), Proceedings of Symposium on Operations Research, Kanpur Institute of Technology, 1965, pp. 124-135.
- 72 -----, "Multiplex Method of Linear Programming," Report (unpublished) from London Computing Service, English Electric Company, July, 1960, 157 pp.
- 73 Saksena, C.P., and S. Nordbotten, "Detailed Flow Chart of the Multiplex Method of Linear Programming," Memorandum, Institute of Economics, University of Oslo, Norway, April 1959.
- 74 Saksena, C.P., and A.J. Cole, "The Bounding Hyperplane Method of Linear Programming," Operations Research; (accepted for publication, revised version submitted in May, 1970).
- 75 Tompkins, C.B., "Projection Methods in Calculation," in H.A. Antosiewicz (ed.), Proceedings of the Second Symposium in Linear Programming, Vol. 2, National Bureau of Standards and Directorate of Management Analysis, DCS/Comptroller, USAF, Washington, D.C., 1955, pp. 425-448.
- 76 Tucker, A.W., "Linear Inequalities and Convex Polyhedral Sets," in H.A. Antosiewicz (ed.), Proceedings of the Second Symposium in Linear Programming, Vol. 2, National Bureau of Standards and Directorate of Management Analysis, DCS/ Comptroller, USAF, Washington, D.C., 1955, pp. 569-602.
- 77 Vajda, S., Mathematical Programming, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1961. 310 pp.
- 78 van de Panne, C., and Andrew Whinston, "The Simplex and Dual Method for Quadratic Programming," International Center for Management Science, Rotterdam, Report 6314 (ICMS No. 28), April, 1963.
- 79 Wade, Thomas E., The Algebra of Vectors and Matrices, Addison-Wesley, Inc., Cambridge, Mass., 1951, 189 pp.
- 80 Waugh, F.V., and G.L. Burrows, "A Short Cut to Linear Programming," Econometrica, Vol. 23. January, 1955, pp. 8-29.
- 81 Wolfe, Philip, "The Simplex Method for Quadratic Programming," Econometrica, Vol. 27, No. 3, July, 1959.
- 82 Wolfe, P. and L. Cutler, "Experiments in Linear Programming," in R. Graves and P. Wolfe (eds.), Recent Advances in Mathematical Programming, McGraw-Hill Book Company, Inc., New York, 1963.

ADDENDUM

- 83 Balas, Egon, "Une methode de decomposition quasi-primale-
duale pour des programmes lineaires," Comptes Rendus
de l'Academie des Sciences (Paris), Vol. 261, 1965,
pp. 2572-2574.
- 84 -----, "Deuxieme variante de la methode de decomposition
quasi-primale-duale pour des programmes lineaires,"
Comptes Rendus de l'Academie des Sciences (Paris),
Vol. 261, 1965, pp. 2809-2811.
- 85 -----, "An Infeasibility-Pricing Decomposition Method for
Linear Programs," Vol. 14, No. 1, 1966, pp. 847-873.
- 86 Dahl, Ole-John, "A Comparison Between the Simplex and
Multiplex Methods," Teknisk notat F-22, ref. 170,
June, 1959, Forsvarets Forskningsin Stikitt, Lillestrom,
Norway.
- 87 Davie, A.J.T., "KLOCK" - a subroutine for timing calculations
on I.B.M. 360/44 computer.
- 88 Dorn, W.S., "A Duality Theorem for Convex Programs," I.B.M.
J. Res. Develop, Vol. 4, 1960, pp. 607-413.
- 89 Hotelling, H., "Some New Methods in Matrix Calculation,"
Ann. Math. Stat., Vol. 4, 1943, pp. 1-33.
- 90 Kuhn, H.W., and A.W. Tucker, "Nonlinear Programming,"
Proceedings of the Second Berkeley Symposium of
Mathematical Statistics and Probability, J. Newman,
University of California Press, Berkeley and Los
Angeles, 1951 pp. 481-492.