THE ELLIPSOID ALGORITHM FOR LINEAR PROGRAMMING

BY

LAISIN MARK

B.Sc. [Maths.] N.A.U., AWKA, 1993

A THESIS SUBMITTED TO THE DEPARTMENT OF MATHEMATICS, AHMADU BELLO UNIVERSITY, ZARIA IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF MASTER OF SCIENCE DEGREE IN MATHEMATICS.

AUGUST, 1998

DECLARATION

I hereby declare that this thesis is a result of my own research work, and that it is not a published or an unpublished work of another author. It has not been accepted or submitted for a degree of any kind in another University. Sources of information have been acknowledged both in the texts and references.

LAISIN MARK
AUGUST 18, 1998.

CERTIFICATION

This thesis entitled **"The Ellipsoid algorithm for Linear Programming"** by **LAISIN MARK** meets the regulations governing the award of the degree of Master of Science of Ahmadu Bello University, Zaria and is approved for its contribution to knowledge and literary presentation.

JNSingh	1-9-1998
DR. J.N. SINGH CHAIRMAN, SUPERVISORY COMMITTEE	DATE
DR. B. SANI	1-9-98
MEMBER, SUPERVISORY COMMITTEE	DATE
DR. M.Y. BELLO for JNSingh	1-9-1998
NAME AND SIGNATURE EXTERNAL EXAMINER	DATE
DR. U. M. MAKARFI	20/1/99
NAME AND SIGNATURE HEAD OF DEPARTMENT	DATE
PROF. J.Y. OLAYEMI	14/06/99
NAME AND SIGNATURE DEAN, POSTGRADUATE SCHOOL	DATE

DEDICATION

This work is dedicated to:

My Beloved FATHER and MOTHER MR & MRS LAISIN THOMAS for their endless love and care.

ACKNOWLEGMENT

To God be the glory, for His wonderful Love and with his GRACE I have been able to pass through this work successfully.

I deeply appreciate His love, the invaluable devotion and motivation of my project supervisor Dr. J.N. Singh, who carefully scrutinized the work. He is really a father when he gives me advice and a friend in the way he relates.

I appreciate the love, the invaluable devotion and motivation of my project supervisor Dr. Babangida Sani for his scrutinizing of the work to ensue that I stay on course is largely responsible for the success of this work.

My Head of Department, Dr. U.M. Makarfi and Prof. M.A. Rashid for their understanding, guidance, fatherly advice and useful discussions during the period of studies.

I appreciate Dr. G.U. Garba, Dr. U.C. Sharma, Dr. S.M.S. Imdadi, Mr. M.R. Quabili, Mr. B.O. Adejo and Mrs. (Prof.) M.A. Rashid, B.A. Madu, for their fatherly advice and useful discussions during the period of studies.

My heartily appreciation goes to my parents Mr. & Mrs. Laisin Thomas, my brothers Mr. Laisin Martin, Mr. Laisin Michael, Mr. Laisin Innocent, Mr. Laisin Kenneth, Mr. Laisin Alex-Gilbert. Miss Laisin Caroline, Terry, Solah, Lenyuy, Mokia, Siben, Laisin Jennet, for love and understanding given to me throughout the years.

I appreciate all my course mates for contributing to my success in one way or the other.

I appreciate Dr. Roland Kangkolo, Mr. John Billy, Mr. Peter, Mr. S.N. Agara ,Mr. Patrick, Rev. Father Samuel and Rev. Father James for useful advice during my studies.

I deeply appreciate the love, the invaluable devotion and motivation of Gabche Agnes for care and understanding throughout the years, to her I am indebted.

My thanks goes to Mr. Isah Abdulkarim for his patience in typing the work.

TABLE OF CONTENTS

ABSTRACT

This thesis deals with the application of the ellipsoid method to the solution of Linear Programming problems with integer data in polynomial time. The thesis is divided into four chapters.

In chapter one, we discuss the general Linear Programming problem, the simplex method and duality in linear programming.

In chapter two, we discuss the ellipsoid algorithm for the system of linear inequalities with rational or integer data and show how it can be used to determine the feasibility in polynomial time. In this chapter we also discuss some antecedents of the ellipsoid algorithms and modifications of the algorithm to achieve better rate of convergence.

Chapter three deals with the application of the ellipsoid algorithm to solve a linear programming problems. Herein, we have discussed Gacs and Lovasz approach, bisection method and the method of sliding objective function. Finally, in this chapter we discuss how exact solutions can be obtained from approximate solutions.

Chapter four deals with the comparison of the ellipsoid algorithm with the other linear programming algorithms. First we have compared the ellipsoid algorithm discussed in chapter two with the simplex algorithm discussed in chapter one. Before going for its comparison with Karmarkar's algorithm, we

have briefly discussed Karmarkar's algorithm and then we have critically examined its relation with the ellipsoid algorithm. Finally we have presented concluding remarks.

CHAPTER ONE

LINEAR PROGRAMMING AND THE SIMPLEX METHOD

1.1 INTRODUCTION

The problem of maximizing or minimizing a linear function subject to linear constraints and non-negativity restrictions on variables, is called a linear programming problem. These days linear programming happens to be one of the most frequently used decision making tools in the industry, administration and various other spheres of life. An extensive bibliography of Linear Programming Applications can be found in "Linear Programming Methods and Applications" by S.I. Gauss (pp. 469-529).

In the years following its inception in 1947, in connection with the planning activities of the military, linear programming has come into wide use.

In academic circles Mathematicians, Operations Researchers, Economists and Computer Scientists, have written hundred of books on the subject and of course, an unaccountable number of research papers.

Interestingly enough, in spite of its wide applicability to everyday problems, linear programming was unknown prior to 1947. It is true that two or three individuals may have been aware of its potentials; for example Fourier in 1823 and de la Vallee Poussin in 1911. But these were isolated cases. Their works were soon forgotten. Kantorvich in 1939 made an extensive proposal that was neglected by U.S.S.R. It was only after the major developments in Mathematical programming had taken place in the west (1959) that Kantorovich's works became known.

1.2 GENERAL LINEAR PROGRAMMING PROBLEM

A general linear programming problem can be stated as follows:

Find $x_1, x_2, \ldots, x_n$

So as to optimize (maximize or minimize) the linear function

$$Z = c_1x_1 + c_1x_2 + \ldots + c_nx_n$$

subject to the constraints:

$$\begin{aligned} &a_{11}x_1+\ldots+a_{1n}x_n (\le,=,\ge)\ b_1 \\ &a_{21}x_1+\ldots+a_{2n}x_n (\le,=,\ge)\ b_2 \\ &\vdots \\ &a_{m1}x_1+\ldots+a_{mn}x_n (\le,=,\ge)\ b_m \\ &x_1,\ x_2,\ \ldots,\ x_n \ge 0, \end{aligned} \qquad \ldots \quad (1)$$

where $a_{i1}, a_{i2}, \ldots, a_{in}; i=1,2,\ldots,m$ are called activities; $(c_1, c_2, \ldots, c_n)$ is a row vector called the price vector and $(b_1, b_2, \ldots, b_n)$ is called the requirement vector.

Without any loss of generality we can assume that all b_i are positive. If any b_i is negative, we can make it positive by multiplying that inequality by -1.

In matrix notion, the above linear programming problem can be stated as follows:

Optimize $Z = C^T X$

Subject to the constraints:

$AX \ (\leq, =, \geq) b$

$X \geq 0$

where $C, X \in \mathbb{R}^n$ and A is an $m \times n$ matrix and $b \in \mathbb{R}$. For the simplicity of description we consider a linear programming problem in maximization form as follows:

$$\text{Max } X = C^T X$$
$$\text{S.t.} \quad AX \leq b \qquad \ldots (2)$$
$$X \geq 0$$

1.3 IMPORTANT DEFINITIONS

The following are some basic definitions of LPP.

1.3.1 FEASIBLE SOLUTION (F.S.)

A feasible solution to a linear programming problem is the set of values of variables which satisfies the set of constraints and the non-negative restrictions of the problem (2).

1.3.2 BASIC SOLUTION (B.S.)

We consider the system AX=b of m-equations in n' variables, where n'>m and the rank of the augmented matrix (A:b)=m. Then non of these equations is redundant. If any m x m non-singular submatrix B is chosen from A and if all the n'-m variables not associated with the columns of this matrix are set equal to zero, then the solution of the resulting system of equation BX=b is called basic solution.

1.3.3 A BASIC FEASIBLE SOLUTION

A feasible solution to a linear programming problem (1) which is also basic is called a Basic feasible solution.

1.3.4 DEGENERATE/NON-DEGENERATE BASIC FEASIBLE SOLUTION

A basic feasible solution of a linear programming problem is said to be degenerate if at least one of the basic variable is zero otherwise it is called a non-degenerate.

1.3.5 OPTIMAL SOLUTION

A solution to a linear programming problem is called an optimal solution if it gives a maximum (for a maximizing problem) or a minimum (for minimizing problem) value of the objective function $Z=C^TX$.

1.3.6 UNBOUNDED SOLUTION

A linear programming problem is said to have an unbounded solution if its objective function increases of decreases arbitrarily i.e. there is no finite optimum value for the objective function.

1.3.7 SLACK AND SURPLUS VARIABLES

(a) SLACK VARIABLE

In a linear programming problem (1) if a constraint has a sign $\leq$, then in order to convert this inequality into equation we add a positive variable to the left hand side. This positive variable is called **Slack variable.**

(b) SURPLUS VARIABLES

In a linear programming problem (1) if a constraint has the sign $\geq$, then in order to convert this inequality into equation we subtract a positive variable from the left hand side. This positive is variable is called **Surplus Variable.**

1.4 SIMPLEX METHOD

The simplex method is the most powerful tool for solving linear programming problem. This method is an algebraic procedure which progressively approaches the optimal solution. This method was developed by George B. Dantzig in 1947 which was made available in 1951. The method start with an initial basic feasible solution and proceed systematically to other basic feasible solutions. Thus in a finite number of iterations it obtains the optimal basic feasible solution.

1.5 SOME IMPORTANT NOTATIONS

(a) Let B be an m x m non-singular square submatrix of the constraints matrix A whose column vectors are linearly independent i.e.,

$$B = (\beta_1, \beta_2, \ldots, \beta_m),$$

where $\beta_1, \ldots, \beta_m$ are linear independent. The matrix B is called the basic matrix.

(b) The variables corresponding to $\beta_1, \beta_2, \ldots, \beta_m$ are called basic variables and are denoted by $x_{\beta 1}, x_{\beta 2}, x_{\beta 3} \ldots, x_{\beta m}$ respectively.

The vector of m basic variables $x_{\beta 1}, x_{\beta 2}, x_{\beta 3} \ldots, x_{\beta m}$ is denoted by $X_B = (x_{\beta 1}, x_{\beta 2}, x_{\beta 3} \ldots, x_{\beta m})$ and is called a basic feasible solution to the linear programming problem (2).

(c) The coefficient of the basic variables $x_{\beta 1}, x_{\beta 2}, \ldots, x_{\beta m}$ in the objective function

$$Z = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$$

are denoted by $C_{\beta 1}, Cx_{\beta 2}, C_{\beta 3} \ldots, C_{\beta m})$ respectively and we put

$$C_B = (C_{\beta 1}, C_{\beta 2}, C_{\beta 3}, \ldots, C_{\beta m})$$

(d) we have

$$B = [\beta_1, \beta_2, \ldots, \beta_m]$$

$$= \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & & \\ b_{m1} & b_{m2} & \cdots & b_{mm} \end{bmatrix}$$

Since $\beta_1, \beta_2, \ldots, \beta_m$ are linearly independent. So they will from a basis for $\mathbb{R}^m$ i.e., each vector in R^m can be expressed as a linear combination of $\beta_1, \beta_2, \ldots, \beta_m$. If α_j denote jth column of matrix A, then $\alpha_j \in R^m$ and we can write

$\alpha_j = y_{1j}\beta_1 + y_{2j}\beta_2 + \ldots + y_{mj}\beta_m$, where y_{ij}, $i=1,\ldots,m$ are scalars.

$$= (\beta_1, \beta_2, \ldots, \beta_m) \begin{pmatrix} y_{1j} \\ \vdots \\ y_{mj} \end{pmatrix}$$

$$= B Y_j, \quad where \ Y_j = \begin{pmatrix} y_{1j} \\ \vdots \\ y_{mj} \end{pmatrix}$$

$$\Rightarrow Y_j = B^{-1}\alpha_j$$

(e) Finally we write

$$Z_j = C_B Y_j$$

1.6 SOME IMPORTANT THEOREMS

The following are the basic theorems of LPP, whose proof can be found in any standard book on LPP. They shall be stated without proofs.

Theorem (1.6.1) Min-Max Theorem

The maximization of a function $Z = C^T X$ where $C, X \in \mathbb{R}^n$ is equivalent to minimization of the negative of the same function and conversely. i.e.,

$$\max C^T X = \min (-C^T X)$$

In view of this theorem a linear programming problem can be considered in any of the form either maximization form or minimization form, as the two forms can be converted to one another.

Theorem 1.6.2

(Fundamental theorem of linear programming)

If a linear programming problem has been an optimal solution, then at least one of the basic feasible solutions must be optimal.

Theorem 1.6.3 (Determination of improved B.F.S.)

Let $X_B = B^{-1}b$ be a B.F.S to a linear programming problem with $Z=C^TX_B$ as the objective function. If for any column α_j in A but not in B the condition $C_j - Z_j > 0$ holds and at least one $y_{ij} > 0$; $i=1,2,\ldots,m$ then it is possible to obtain a new B.F.S., by replacing one of the columns in B by α_j and if the new value of the objective function is Z' then $Z' \geq Z$. If given B.F.S. is non-degenerate then $Z' > Z$.

Theorem 1.6.4 (Optimality conditions)

Let $X_B = B^{-1}b$ be the B.F.S. to the linear programming problem

$$\text{Max } Z = C^TX$$

$$\text{S.t. } AX = b$$

$$X \geq 0$$

and $Z^* = C_B X_B$ be the value of the objective function for this B.F.S.. If $C_j - Z_j \le 0$ for every column α_j in A but not in B then Z^* is the optimum (maximum) value of the objective function Z and this B.F.S. X_B is an optimal B.F.S.

1.7(a) TO DETERMINE STARTING B.F.S FOR CONSTRAINTS WITH INEQUALITIES SIGNS ≤.

We consider a linear programming problem

$$\text{Max } Z = C_1x_1 + \ldots + c_nx_n$$

$$\begin{aligned} \text{s.t.} \quad & a_{11}x_1 + \ldots + a_{1n}x_n \le b_1 \\ & a_{21}x_1 + \ldots + a_{2n}x_n \le b_2 \\ & \vdots \\ & a_{m1}x_1 + \ldots + a_{mn}x_n \le b_m \\ & x_1, \ldots, x_n \ge 0 \end{aligned}$$

After introducing slack variable $x_{n+1}, \ldots, x_{n+m}$ above linear programming problem can be stated as follows:

$$\text{Max } \quad Z = C_1x_1 + \ldots + C_nx_n + 0x_{n+1} + \ldots + 0x_{n+m}$$

$$\begin{aligned} \text{s,t.} \quad & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + x_{n+1} + 0 + \ldots + 0 = b_1 \\ & \vdots \qquad \vdots \\ & a_{m1}x_1 + c_{m2}x_2 + \ldots + a_{mn}x_n + x_{n+1} + 0 + 0 + \ldots + x_{n+m} = b_m \\ & x_1, x_2, \ldots, x_n, x_{n+1}, \ldots, x_{n+m} \ge 0 \end{aligned}$$

In matrix form the above linear programming problem can be written as:

$$\text{Max } \quad Z = C^T X$$

$$\text{s.t. } AX = b$$

$$X \ge 0$$

where $C = (c_1, c_2, \ldots, c_n, 0, \ldots, 0)$

$X = (x_1, x_2, \ldots, x_n, \ldots, x_{n+m})^T$

and

$$A = \begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} & 1 & 0 & 0 & \ldots & 0 \\ a_{21} & a_{22} & \ldots & a_{2n} & 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \ldots & & & & & & \\ a_{m1} & a_{m2} & \ldots & a_{mn} & 0 & 0 & 0 & \ldots & 1 \end{bmatrix}$$

Here we take $B = I_m$ (unit matrix of order m x m) as basic matrix and consequently we get the initial B.F.S. as $X_B=B^{-1}b=I_mb=b\geq 0$.

1.7(b) TO DETERMINE STARTING B.F.S. FOR CONSTRAINTS WITH INEQUALITIES SIGNS ≥.

Step I

In this case we convert inequalities into equations by subtracting positive variable called surplus variable from the left hand side of each inequality with sign ≥. Thus after introduction of the surplus variables $x_{n+1}, \ldots, x_{n+m}$ constraints take the form:

$$a_{11}x_1+\ldots+a_{1n}x_n-x_{n+1}+0+\ldots+0 = b_1$$

$$\vdots \qquad \vdots \qquad \vdots$$

$a_{m1}x_1+...+a_{mn}x_n+0+0+...+x_{n+m} = \quad b_m$

In matrix form, these equation can be written as:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & -1 & 0 & \cdots & 0 \\ a_{21} & a_{22} & \cdots & a_{2n} & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & & & \vdots & \vdots & & \vdots \\ a_{ml} & a_{m2} & \cdots & a_{mn} & 0 & 0 & \cdots & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n+1} \\ \vdots \\ x_{n+m} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Thus the initial basis matrix $B = -I_m$ and consequently we get $X_B = B^{-1}b = -I_m b = -b \le 0$

So the basic solution is not a basic feasible solution. In order to create an identity matrix we add one more variable called artificial variable to each constraints. Thus after adding surplus $x_{n+1}, \ldots, x_{n+m}$ and artificial variables $x_{n+m+1}, \ldots, x_{n+m+m}$, the constraints of a given linear programming problem can be written as follows:

$$
\begin{array}{llllll}
a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n - x_{n+1} + x_{n+m+1} & & & = b_1 \\
a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n & -x_{n+2} & +x_{n+m+2} & = b_2 \\
\vdots \quad \vdots \quad \vdots & & & \\
a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n & -x_{n+m} & + x_{n+m+m} & = b_m
\end{array}
$$

In matrix form, these equations can be written as:

$$
\begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} & -1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \\
a_{21} & a_{22} & \cdots & a_{2n} & 0 & -1 & \cdots & 0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & & & & & & & & & & \\
a_{m1} & a_{m2} & \cdots & a_{mn} & 0 & 0 & \cdots & -1 & 0 & 0 & \cdots & 1
\end{bmatrix}
\begin{bmatrix} x_1 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{n+m} \\ x_{n+m+1} \\ \vdots \\ x_{n+m+m} \end{bmatrix}
=
\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}
$$

Then Basis matrix $B = I_m$

$\therefore X_B = B^{-1}b = I_m b = b \geq 0$

which is a B.F.S.

Thus the B.F.S. can be obtained by writing all the non-basic variables $x_1, \ldots, x_n, x_{n+1}, \ldots, x_{n+m}$ equal to zero and solving the equations for the artificial variables $x_{n+m+1}, x_{n+m+2}, \ldots, x_{n+m+m}$.

1.7(c) THE SIMPLEX ALGORITHM

Step 1

If linear programming problem is a minimization one, we convert it into a maximization problem using min-max theorem 1.6.1.

Step 2

Make all b_i positive, if some b_i is negative multiply both sides of the corresponding inequality equation by -1.

Step 3

Convert inequalities into equation using slack, surplus and artificial variables (where applicable).

Step 4

Find the initial starting basic feasible solution.

Step 5

Construct a simplex table like table 1 in the appendix.

Step 6

Use theorem 1.6.4 to test the starting B.F.S. for Optimality i.e.

(I) If $\Delta_j = C_j - Z_j \leq 0 \; \forall_j$ the solution under the test is optimal.

(a) If some of Δ_j are zero then the other optimal solution will exist with some value of Z.

(b) If $\Delta_j < 0, \; \forall j$ then unique solution will exits.

(ii) If $\Delta_j > 0$ for any j then the solution under the test is not optimal and in view of theorem (1.6.3) it is possible to get an improve B.F.S., so, go to step 7. To get an improved B.F.S.

Step 7

To find entering and out going vector. The incoming vector will be taken as α_k if Δ_j = Max Δ_j.

The out going vector β_r is taken correct to that value of r for which

$$\frac{X_{\beta r}}{y_{rk}} = \min \left\{ \frac{X_{Bi}}{y_{ik}}, y_{ik} > 0 \right\}$$

Remark

If the minimum is not unique solution will be degenerated.

Step 8

If α_k be entering vector and β_r be the outgoing vector then the element

y_{rk} (= a_{rk}) is called key element or pivot element.

We will continue the process of improving B.F.S. until we get an optimal solution.

1.8 DEGENERACY IN LINEAR PROGRAMMING

As we have discussed in section 1.3.4 that a B.F.S. of a linear programming problem is said to be degenerate B.F.S. if at least one of the basic variables is zero. So far we have considered linear programming problems in which by the minimum ratio rule we get only one vector to be deleted from the basis. But there are linear programming problems with more than one vector to be deleted from the basis. Thus if

$\min_{i} \left\{ \frac{x_{Bi}}{y_{ik}}, y_{ik} > 0 \right\}$ (α_k is incoming vector) occurs at $i = i_1, i_2, \ldots, i_s$ ie

minimum occur for more than one value of I. Then the problem is to select the vector to be deleted from the basis. In such cases if we choose one vector say β_i where i is one of $i_1, i_2, \ldots, i_s$ and delete it from the basis then the next solution may be a degenerate B.F.S. i.e the objective function value is not improved. Such problem in linear programming is called the problem of degeneracy. In some cases due to the presence of degeneracy the same sequence of simplex table are repeated forever without ever reaching the optimal solution. This problem is called cycling. The procedure which prevent cycling within the simple routine and an optimal solution is obtained in a finite number of steps is called the resolution of degeneracy.

Degeneracy in linear programming can be resolved by either Charne's Perturbation method or by Generalized simplex method.

1.9 DUALITY IN LINEAR PROGRAMMING

INTRODUCTION

In linear programming, each linear programming problem is associated with another linear programming problem called the dual of the primal problem. Both the primal and the dual problem having the same problem data. Optimal solution of either problem reveals information concerning the optimal solution of the other i.e., if the optimal solution of its dual is known then the optimal solution of the primal is also available.

1.9 (a) SYMMETRIC DUAL PROBLEMS

We consider a linear programming problem

$$\begin{aligned} \text{Max} \quad & Z_p = C^T X \\ \text{s.t.} \quad & AX \leq b \qquad \ldots (3) \\ & X \geq 0 \end{aligned}$$

Then with the same problem data we construct another linear programming problem

$$\begin{aligned} \text{Min} \quad & Z_D = b^T Y \\ \text{s.t.} \quad & A^T y \geq C \qquad \ldots (4) \\ & Y \geq 0 \end{aligned}$$

Thus problem (4) is dual of the primal problem (3).

We consider minimization if the new objective function $Z_D=b_1y_1+...+b_my_m$.

1.9(b) UNSYMMETRIC DUAL PROBLEM

Consider the linear programming problem: Find $X \in \mathbb{R}^n$ so as to

$$\text{Max} \quad Z_p = C^T X$$
$$\text{s.t. } AX=b$$
$$X \geq 0$$

Then the dual of this problem can be written as follows: Find $y \in \mathbb{R}^m$ so as to

$$\text{Min} \quad Z_D = b^T Y$$
$$\text{s.t. } A^t y \geq C$$

1.9(c) DUALITY THEOREMS

Theorem (I):

The dual of the dual of a given primal problem is the primal itself.

Theorem (II):

If X be a feasible solution to the primal problem

$$\text{Max} \quad Z_p = C^T X$$
$$\text{s.t. } AX \leq b$$
$$X \geq 0$$

and Y is a feasible solution of the dual problem

$$\text{Min} \quad Z_D = b^T Y$$

$$\text{s.t. } A^T y \geq C$$

$$Y \geq 0$$

Then $C^T X \leq b^T W$.

Theorem (III):

If X is a feasible solution to the primal problem:

$$\text{Max} \quad Z_p = C^T X$$

$$\text{s.t. } AX \leq b$$

$$X \geq 0$$

and $\hat{Y}$ is a feasible solution to the dual problem:

$$\text{Min} \quad Z_D = b^T Y$$

$$\text{s.t. } A^T y \geq C$$

$$Y \geq 0$$

such that $C^T\hat{X} = b^T\hat{Y}$, then $\hat{X}$ and $\hat{Y}$ are optimal solution [Proof of theorem (I),(II) and (III) can be found in Hadley [13].

Theorem (IV) (Fundamental duality theorem):

If a finite optimal feasible solution exist for the primal, then there exist a finite optimal feasible solution for the dual and conversely. The value of the two objective functions are equal.

Proof:

We consider a pair of primal and dual problems as follows:

Primal

$$\text{Max } Z_p = CX$$
$$\text{s.t. } AX \le b \qquad \ldots \quad (1.9\text{-}1)$$
$$X \ge 0$$

Dual

$$\text{Min } Z_D = b^T W1$$
$$\text{s.t. } A^T W \ge C^T \qquad \ldots \quad (1.9\text{-}2)$$
$$W \ge 0$$

Now introducing slack vectors $X_s = (x_{1s}, x_{2s}, \ldots, x_{ms})$. Then the primal problem (1.9-1) can be restated as follows:

$$\text{Max } Z_p = CX$$
$$\text{s.t. } AX + IX_s = b \qquad \ldots \quad (1.9\text{-}3)$$
$$\text{and } X \ge 0,\ X_s \ge 0$$

Let X_B be the finite optimal feasible solution to the primal problem (1.9-1). Let B be the basic matrix corresponding to the feasible solution X_B and $C_B = (C_{B1}, C_{B2}, \ldots, C_{Bm})$ be the vector containing the prices of the basic variables. Since X_B is the optimal solution to the primal problem.

So, $\Delta_j \le 0$ for all j

i.e., $-Z_j + C_j \le 0$

i.e., $-C_B B^{-1}\alpha_j + C_j \le 0$

for all α_j including surplus variables we have,

$C_B B^{-1}\alpha_j + C_j \geq C_j$ ... (1.9-4)

i.e., $C_B B^{-1}(\alpha_1, \alpha_2, \ldots, \alpha_n) \geq (c_1, c_2, \ldots, c_n)$

for all vectors of $A = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ ... (1.9-5)

or $C_B B^{-1} A \geq C$

If we write $(\hat{W})^T = (w_1, w_2, \ldots, w_m)$

$= [w_1, w_2, \ldots, w_m]^T$

∴ From (1.9-5) we have

$(\hat{W})^T A \geq C$

or

$[(\hat{W})^T A]^T \geq C^T$

By taking the transpose of both sides

or $A^T [(\hat{W})^T]^T \geq C^T$

$A^T \hat{W} \geq C^T$

which proves that $\hat{W}$ satisfies the constraints of the dual problem (1.9-2).

Now we consider the equation (1.9-4) with α_j corresponding to slack variables. Therefore in this case $C_j = 0$.

∴ From (4) we have

$C_B B^{-1}(e_j) \geq 0$

for all $j = 1, 2, \ldots, m$

Or $(w_1, w_2, \ldots, w_m)\ e_j \geq 0$

$w_j \geq 0$

ie $w = [w_1, w_2, \ldots, w_m]$ is the feasible solution of the dual

(1.9-2).

Now we proceed to prove that w is an optimal solution to the dual (1.9-2)

$$\text{Now } Z_D = b^T\hat{W}$$

$$= [(\hat{W})^T b]^T$$

$$= (\hat{W})^T b$$

$$= (C_B B^{-1})$$

$$= (C_B (B^{-1} b)$$

$$= C_B X_B = \text{Max } Z_p.$$

Thus $\hat{W}$ and X_B are feasible solutions of the dual (1.9-2) and the primal (1.9-1) respectively such that

$$Z_p = Z_D.$$

Hence by Duality Theorem III, $\hat{W}$ is the optimal solution of the dual problem (1.9-2) which proves the theorem.

Theorem (V) (Complementary Slackness Theorem):

For the optimal feasible solutions of the primal and dual system.

(a) Whatever inequality occurs in the ith relations of either system, if the corresponding slack or surplus variable is positive, then the ith variable of its dual is zero.

(b) If the jth variable is positive in either system the jth relations of its dual holds as a strict equality (i.e. the corresponding slack or surplus variable W_{2+1} vanishes).

Proof

The primal and the dual problems in explicit form after introducing the non-neggative slack variables

$$X_{n+1},\ X_{n+2},\ \ldots,\ X_{n+m}$$

in the primal constraints and introducing the non-negative surplus variables

$$W_{m+1},\ W_{m+2},\ \ldots,\ W_{M+n}$$

in the dual constraints, can be written as

Primal problem

$$\text{max. } Zp = C_1x_1 + C_2x_2 + \ldots + C_n x_n \quad \ldots(1.9\text{-}6)$$

$$\begin{aligned} \text{s.t. } & a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n + x_{n+1} = b_1 \\ & a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n + x_{n+2} = b_2 \quad \ldots (1.9.7) \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n + x_{n+1} = b_m \\ \text{and } & x_1, x_2, \ldots x_n, x_{n+1}, \ldots x_{n+m} \geq 0 \end{aligned}$$

Dual Problem

$$\text{min. } Z_D = b_1w_1 + b_2w_2 + \ldots + b_nw_m \quad \ldots(1.9\text{-}8)$$

$$\begin{aligned} \text{s.t } & a_{11}w_1 + a_2w_2 + \ldots + a_{m1}w_m - w_{m+1} + 0 + \ldots 0 = c_1 \\ & a_{12}w_1 + a_{22}w_2 + \ldots + a_wx_m - w_{m+2} + \ldots + 0 = c_2 \quad \ldots (1.9.9) \\ & \vdots \\ & a_{2n}w_1 + a_{2n}w_2 + \ldots a_{mn}w_n + 0 + \ldots - w_{m+n} = c_n \\ \text{and } & w_1, w_2, \ldots w_m, w_{m+1}, \ldots w_{n+m} \geq 0 \end{aligned}$$

Now, multiplying the equations of (1.9.7) by

$$w_1, w_2, \ldots w_m$$

respectively and then adding, we have

$$x_1 \sum_{i=1}^{m} a_{i1} w_i + \ldots + x_n \sum_{i=1}^{m} a_{in} w_i + x_{n+1} w_1 + x_{n+2} w_2 + \ldots + x_{n+m} w_m = \sum_{i=1}^{m} b_i w_i \quad \ldots (1.9\text{-}10)$$

subtracting (1.9-10) from (1.9-6), we have $\left(c_1 - \sum_{i=1}^{m} a_{i1} w_i\right) x_1$

$$+ \ldots + \left(c_n - \sum_{i=1}^{m} a_{in} w_i\right) x_n - w_1 x_{n+1} - w_2 x_{n+2} - \ldots - w_m x_{n+m} = \left(Z_p - \sum_{i=1}^{m} b_i W_i\right)$$

$$= Z_p - Z_D \quad \ldots (1.9\text{-}11)$$

from (1.9-9), we have

$$-W_{m+j} = c_j - (a_{1j} w_j + a_{2j} w_2 + \ldots + a_{mj} w_m$$

$$j = 1, 2, \ldots, n.$$

$$-W_{m+j} = c_j - \sum_{i=1}^{m} a_{ij} w_i \quad \ldots (1.9\text{-}12)$$

for all $j = 1, 2, \ldots, n$.

Using (1.9-12) in (1.9-11), we have

$$-(W_{m+1} x_1 + W_{m+2} x_2 + \ldots + W_{m+n} x_n) -$$

$$(W_1 x_{n+1} + W_2 x_{n+2} + \ldots + W_n x_{n+m}) = Z_p - Z_p \quad \ldots (1.9\text{-}13)$$

Now if $x^* = (x^*_1, x^*_2, \ldots, x^*_n)$ and

$$W^* = (w^*_1, w^*_2, \ldots, w^*_n)$$

be the optimal solutions of the primal and the dual problems respectively, then by Duality theorem IV

$$Z^*_p = Z^*_D$$

Thus for the optimal solution x^* and w^* of primal and the dual problems the corresponding slack and surplus variables

$x^*_{n+i} \geq 0,\ i = 1, 2, \ldots, m$

$w^*_{n+j} \geq 0,\ j = 1, 2, \ldots, n$

From equation (1.9-13), we have

$(W^*_1 X^*_{n+1} + W^*_2 X^*_{n+2} + \ldots + W^*_n X^*_{n+m}) +$

$(W^*_1 X^*_{n+1} + W^*_2 X^*_{n+2} + \ldots + W^*_n X^*_{n+m}) = 0 \quad \ldots (1.9\text{-}14)$

Since all the variables in (1.9-14) are non-negative and so their product terms in (1.9-14) are also non-negative. The sum of these positive products in (1.9-14) will be zero if each term is individually equal to zero.

i.e. $w^*_{m+j} x^*_j = 0 \qquad \forall j = 1, 2, \ldots, n$........... (1.9-15)

and

$w^*_i X^*_{n+i} = 0 \qquad \forall j = 1, 2, \ldots, m$ ....(1.9-15)

(a) From (1.9-16), if $x^*_{m+i} > 0$, them we must have $^*w = 0$, i.e. if the slack variable in the ith relation of the primal is positive then ith variable of the dual is zero.

Again from (1.9-15), if $w^*_{m+j} > 0$, them we must have $x^*_j = 0$, i.e if the surplus variable in jth variable of the primal (dual of the dual) is zero.

This proves the part (a) of the theorem.

(b) Also from (1.9-15), if $x^*_j = 0$, then we must have $w^*_{n+j} = 0$ i.e if the jth variable in the primal is zero then the jth relation in the dual is strictly an equality (as $w^*_{n+j} = 0$).

Again from (1.9-16), if $w^{*}>0$ then $x^{*}_{m+i} = 0$, i.e. if the ith variable in the dual is positive then the ith relation in the primal (dual of the dual) is strictly an equality (as $x^{*}_{n+i} = 0$).

ALTERNATIVE STATEMENT OF THE COMPLEMENTARY SLACKNESS THEOREM

A necessary and sufficient condition for any pair of feasible solutions to the primal and dual to be optimal is that

$w_i x_{n+i} = 0, \ i = 1, 2, \ldots, m$

where x_{n+1} is the sack variable in the primal and

$x_j w_{m+j} = 0, \ j = 1, 2, \ldots, n$

where w_{m+j} is the surplus variable for the dual.

CHAPTER TWO

THE ELLIPSOID METHOD

2.0 INTRODUCTION

In this chapter we describe the ellipsoid algorithm for a system of linear inequalities and linear programming problems with integer data. Also we outline the modifications introduced by Khachiyan and the arguments used by him to prove that the feasibility or infeasibility of such a system can be determined in a polynomial time using this algorithm.

In section 2.1 we state some important definitions and in section 2.2 we describe the Ellipsoid Algorithm in $\mathbb{R}^n$.

In section 2.3 we describe the development of the ellipsoid method and give some perspective on its place in mathematical programming. We shall see that it is closely related to (I) the Relaxation method for linear inequalities (II) the subgradient method and (III) the method of central section for convex minimization.

In section 2.4 we describe several simple modifications to the basic algorithm to improve it rate of convergence. The deep cut will be described in details.

2.1 DEFINITIONS

Def. 2.1.1 The Ellipsoid in $\mathbb{R}^n$

An ellipsoid with centre at x_o in $\mathbb{R}^n$ is represented by the set

$$E_o = \{X \in \mathbb{R}^n : (x-x_o)^T D_o^{-1} (x-x_o) \leq 1\}$$

where D_o is a positive definite symmetric matrix.

Def. 2.1.2 Polynomial Algorithm

An algorithm A is called a polynomial algorithm for a problem P, if the number of steps to solve P on applying A is bounded by a polynomial function $\phi(m,n,L)$ of the dimension and input length of the problem.

Def. 2.1.3 INPUT LENGTH

We consider a system of m linear inequalities in n unknowns.

$x_1, \ldots, x_j, \ldots, x_n$ as follows:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq b_1$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq b_2$$
$$\vdots$$
$$a_{m1}x_1 + c_{m2}x_2 + \ldots + a_{mn}x_n \leq b_m$$

where a_{ij} and b_j are integer coefficient and $(X_1, \ldots, x_n) \in \mathbb{R}^n$. Then the input length L of the system of linear inequalities can be defined as follows:

$$L = \lceil \sum_{i=1}^{m} \sum_{j=1}^{m} \log_2(|a_{ij}|+1) + \sum_{i=1}^{m} (|b_i|+1) + \log_2 mn \rceil + 1$$

where $\lceil x \rceil$ denotes the least integer greater than or equal to x.

2.1.4 LINEAR CONVERGENCE

The sequence $<x_k>$ generated by an algorithm A converges to x_o linearly if for some norm $||.||$ there is an $\alpha \in (0,1)$ and $k_o \geq 0$ such that $\|x_{k+1}-x_o\| \leq \alpha\|x_k - x_o\|$ for all $k \geq k_o$

i.e., $$\frac{\|x_{k+1}-x_o\|}{\|x_k-x_o\|} \leq \alpha \ (<1)$$

2.1.5 SUPERLINEAR CONVERGENCE

The sequence of points $<x_n>$ generated by an algorithm A converges to x_o superlinearly if

$$\lim_{k\to\infty} \frac{\|x_{k+1}-x_o\|}{\|x_k-x_o\|} = 1.$$

Let the sequence $<f_k>$ of the functional values of the points generated by an algorithm converges to f_o then if

$$\overline{\lim_{k\to\infty}} \frac{|f_{k+1}-f_o|}{|f_k-f_o|^p} = \beta < \infty \ ,$$

then the suprimum of non-negative numbers p is called the speed of convergence. The number β used above is called convergence ratio. If $\beta < 1$, $p \leq 1$, then we say that the algorithm admits linear convergence. If $p=1, \beta=0$ then we say that algorithm admits superlinear convergence. Further details about the rate of convergence can be found in Bazaraa and Selty [2], Dennis and More [5] and Sing and Singh [38,39].

2.2 THE ELLIPSOID ALGORITHM IN $\mathbb{R}^n$

In this section we discuss how to use the ellipsoid algorithm to determine the feasibility of the system of linear inequalities with integer (rational) data in polynomial time.

Suppose we wish to find an n-vector x satisfying:

$$a_1x = a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leq \beta_1$$
$$a_2x = a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leq \beta_2$$
$$\vdots$$
$$a_mx = a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leq \beta_m$$

which can also be written as;

$$a_i^Tx \leq \beta_i \quad , \qquad i=1,2,\ldots,m$$

In matrix notations the above inequalities can be restated as

$$A^T x \leq b \qquad \ldots \quad (2.2\text{-}1)$$

Where $A^T \in Z^{m\times n}$, $x \in \mathbb{R}^n$ and $b \in Z^m$. And we assume that n>1 for n=1 is the trivial case.

THE BASIC ITERATION

The ellipsoid method constructs a sequence of ellipsoids:

$$E_j = \{x \in \mathbb{R}^n : (x-x_j)^T D_j^{-1} (x-x_j) \leq 1\}$$

for all j=0,1,2,...,k, and each of the E_j contain a point satisfying (2.1). At any (k+1)the iteration, we check, if the centre x_k of the current ellipsoid E satisfies all inequalities, if so we stop and x_k will be the solution of the system of linear inequalities. If not, take some constraint violated by x_k, say

$$a_i^T x > \beta_i \text{ for some } i,\ 1 \leq i \leq m \qquad \ldots \quad (2.2\text{-}2)$$

is chosen and the ellipsoid of minimum volume that contains half ellipsoid

$$\{x \in E_k : a_i^T x \leq a^T x_k\} \qquad \ldots \quad (2.2\text{-}3)$$

is constructed using the following formulae.

$$x_{k+1} = x_k - \frac{\tau D_k a}{\sqrt{a^T D_k a}} \qquad \ldots \quad (2.2\text{-}4)$$

$$D_{k+1} = \delta\left(D_k - \frac{\sigma(D_k a)(D_k a)^T}{a^T D_k a} \right) \qquad \ldots \quad (2.2\text{-}5)$$

where τ is called the step parameter

δ is called the dilation parameter

σ is called the expansion parameter.

where

$$\tau = \frac{1}{(n+1)}, \quad \sigma = \frac{2}{n+1} \quad and \ \delta = \frac{n^2}{n^2-1} \qquad \ldots \quad (2.2\text{-}6)$$

If no feasible solution is obtained after performing $6n^2l$ steps. Then we stop and conclude that the system of linear inequalities is infeasible.

2.3 THE ANTECEDENT OF THE ELLIPSOID ALGORITHM

In this section we describe some earlier methods which are closely related to the ellipsoid method. These methods are the Relaxation method for linear inequalities [1,9,29,52], the subgradient method [8, 37, 43, 56] and the method of the central section [17, 25, 28, 30, 48] for convex programming problems.

2.3.1 THE RELAXATION METHOD

Agmon [1] and Motzkin and Schoenberg [29] simultaneously introduced the relaxation method to solve the system of linear inequalities.

They considered the problem (2.2-1) and produce a sequence of iterates $\{x_k\}$. At any k^{th} iteration, if $_k x$ is feasible solution then the algorithm, stops. If x_k is not feasible then a violated constraint say (2.2-2) is chosen and we set,

$$x_{k+1} = x_k - \frac{\lambda_k a(a^T x_k - \beta)}{(a^T a)} \qquad \ldots \quad (2.3\text{-}1)$$

where λ_k is called the relaxation parameter. Motzkin and Schoenberg's [29] consider $\lambda_k = 2$ whereas Agmon [1] takes λ_k between 0 and 2. The choice $\lambda_k=1$ corresponds to the projection of x_k onto the hyperplane $\{x \in \mathbb{R}^n : a^T x = \beta\}$.

It also follows that if λ_k is between 0 and 2, then it correspond to the ball method , see Bland, Goldfarb and Todd [11, p. 1046].

2.3.2 THE SUBGRADIENT METHOD

Shor [43] introduced the subgradient method for minimizing a convex, (not necessarily differentiable) function

$$f:\mathbb{R}^n \rightarrow \mathbb{R}.$$

It has a general form

$$x_{k+1} = x_k - \frac{\mu_k g_k}{\|g_k\|} \qquad \ldots \quad (2.3\text{-}2)$$

where g_k is a subgradient of the function f at x_k.

Note: If we wish to solve (2.2-1) we can minimize

$$F(x) = \max\left\{\max_i\left\{a_i^T X - \beta\right\}, 0\right\} \ldots\ldots\ldots (2.3\text{-}3)$$

Then $a=a_i$ is the subgradient of f(x) at x_k; if $a_i^T x \leq \beta_i$ is the most violated constraint from (2.2-1). Thus (2.3-2) includes as a special case of (2.3-1) in which a constraint of most violation is chosen.

Ermolev [6] and Polyak [37] give choices for μ_k that ensure global convergence for this method.

2.3.3 THE METHOD OF CENTRAL SECTIONS

The third method on which the ellipsoid algorithm is based, is developed independently by Levin [28] and Newman [30] who addressed the problem of minimizing a convex function f(x) over a bounded polyhedron $P_o \subseteq \mathbf{R}^n$.

The method produces a sequence of iterates $\{x_k\}$ and polytopes $\{P_k\}$ by choosing x_k as the centre of gravity of P_k and

$$P_{k+1} = \{x \in P_k : g_k^T x \leq g_k^T x_k\}$$

where g_k is the subgradient of f(x) at x_k since f(x) is convex, P_{k+1} contains all points of P_k whose objective function value is not greater than, that of x_k. In this case the volume of P_{k+1} is at most $(1-e^{-1})$ times that of P_k. However calculating the centres of gravity of polytopes with many facets in high dimensional spaces is an almost cumbersome task. Levin [28] proposed some simplification for n=2.

2.4 REFINEMENT OF ALGORITHM

In this section we describe several simple modifications to the algorithm discussed in this section to improve its rate of convergence. The most obvious way to do this is to use deep cuts or at each iteration to generate smaller ellipsoids. We also discuss the surrogate cuts and the parallel cuts to achieve better rate of convergence.

(a) DEEP CUTS

We consider the system (2.2-1) and suppose (2.2-3) is the violated constraints for some x_k. Then, it follows that the ellipsoid E_{k+1} determined by formulae (2.2-4) to (2.2-6) contains the half ellipsoid

$$\{x \in E_k : a^T x > \beta\}.$$

Thus, it is clear that an ellipsoid of smaller volume is obtained by constructing a minimum ellipsoid that contains the hyperplane

$$a^T x = \beta$$

instead of the cut $\quad a^T x \le a^T x_k$

which passes through the centre of E_k as shown in the figure 1A and 1B.

ELLIPSOID WITHOUT DEEP CUT

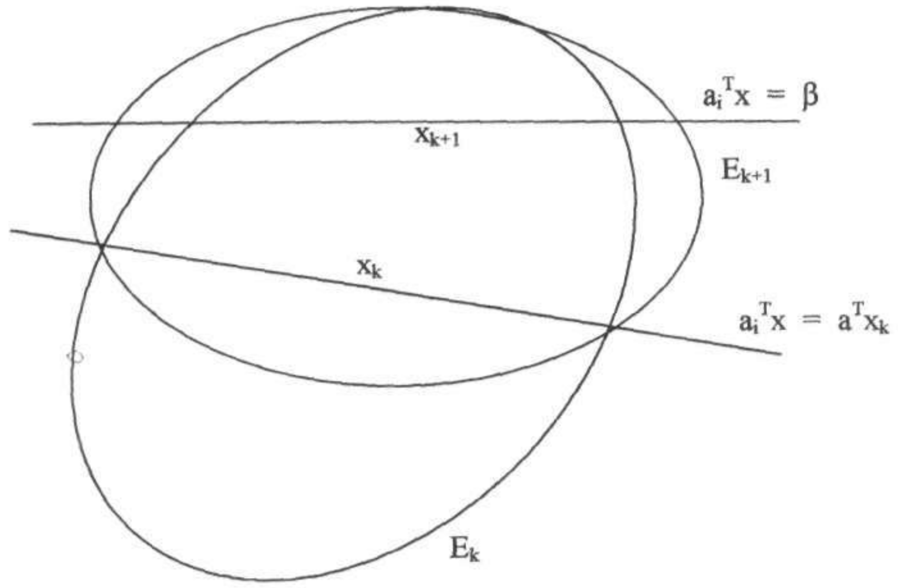


Figure 1a

ELLIPSOID WITH DEEP CUT

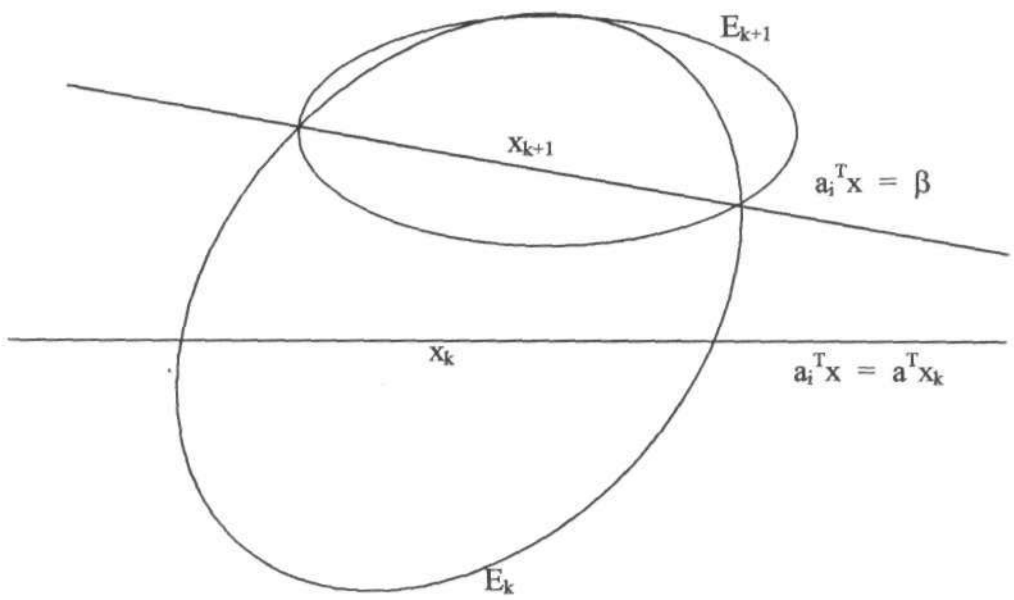


Figure 1b

Such smallest ellipsoid is given by x_{k+1} and D_{k+1} as in (2.2-4) and (2.2-5) with parameters τ, σ and δ chosen as:

$$\tau = \frac{1+n\alpha}{n+1}$$

$$\sigma = \frac{2(1+n\alpha)}{(n+1)(1-\alpha)} \qquad \ldots (2.4\text{-}1)$$

$$\delta = \frac{n^2(1-\alpha^2)}{n^2-1}$$

where $$\alpha = \frac{a^T x_k - \beta}{\sqrt{a^T D_k a}} \qquad \ldots (2.4\text{-}2)$$

o

The formulae (2.2.-4) and (2.2-5) with the choice of parameters τ, σ, δ and α given by equations (2.4-1) and (2.4-2) are only valid for determining E_{k+1} for $-1/n \le \alpha \le 1$.

If $\alpha < -1/n$, the smallest ellipsoid containing $E_k \cap H$ is E_k itself, where H is a half space given by

$$H = \{x \in \mathbb{R}^n : a^T x < \beta\}.$$

Now if $-1/n \le \alpha \le 1$, then

$$\frac{Volume\ (E_{k+1})}{Volume\ (E_k)} = \left[\frac{n^2(1-\alpha^2)}{n^2-1}\right]^{\frac{n-1}{2}} \times \left(\frac{n(1-\alpha)}{n+1}\right) \qquad \ldots(2.4\text{-}3)$$

where $\alpha = -1/n$ then this volume ratio decreases monotonically from one to zero. When $\alpha=1$, E_{k+1} degenerates to a point. Now by computing α for each inequality in (2.2-1), we can select the deepest cut possible, i.e., the cut corresponding to the largest possible value of α. If we get some value of α which is greater than one, then the system of linear inequalities (2.2-1) is infeasible. We expect, that the use of deep cut can increase the speed of convergence. But it has been observed in Bland, Goldfarb and Todd [11, p.1085] that the improvement achieved by the use of deep cut is not much encouraging.

(b) SURROGATE CUTS

One can sometimes obtain cuts that are deeper than any cut generated by a single constraint in (2.2-1), by combining inequalities in it. Any cut of the form

$$u^T A^T x \leq u^T b$$

(i.e., $a^T x \leq \beta$ with $a=Au$ and $\beta=u^T b$) is valid as long as $u \geq 0$ for then no points that satisfy (2.2-1) are cut off by inequality.

Goldfarb and Todd [11] introduce the term "Surrogate" cuts, while Krol and Mirman [26] proposed that this idea of using surrogate cuts with ellipsoid method.

Thus the "best" or deepest surrogate cut is one which can be obtained by solving

$$\underset{u \geq 0}{Max} \; \frac{u^T (A^T x_k - b)}{\sqrt{u^T A^T D A u}}$$

which is equivalent to solving a quadratic programming problem. Let $\bar{A}^T X \leq \bar{b}$ (2.4-4)

be any subset of constraints (2.2-1), where the columns of $\bar{A}$ are linearly independent and at least one of the constraints in (2.4-4) is violated by x_k. It has been observed in Goldfarb and Todd [11] that if

$$\bar{u} = (\bar{A}^T D_k \bar{A})^{-1}(\bar{A}x_k - \bar{b}) \quad \ldots \quad (2.4.5)$$

is non-negative, then the surrogate cut

$$\bar{u}^T \bar{A}^T x \leq \bar{u}^T \bar{b}$$

is deepest with respect to that subset. It is shown in Goldfarb and Todd [11] and Todd [52], that if $\bar{A}^T D_k \bar{A}$ has non-positive off-diagonal entries i.e. the constraint normals in $\bar{A}$ are mutually obtuse in the metric given by D_k and if x_k violates all constraints in (2.4-4), then the $\bar{u}$ given by (2.4-5) is non-negative.

The price to obtain the deepest or nearly deepest surrogate cut is too high a price to pay in solving quadratic programming problem or computing $\bar{u}$ by (2.4-5). Consequently in Goldfarb and Todd [11] it is recommended that only surrogate cuts which can be generated from two constraints be considered.

Krol and Mirman [26] given necessary and sufficient conditions for forming surrogate cut using a deepest cut together with another less violated possibly even satisfied constraint.

These condition indicate whether or not the $\bar{u}$ in (2.4-5) for the two constraints case is non-negative. Thus the surrogate cut is deeper than the cut produced by either the two-constraints. The process can be repeated iteratively using the newly formed surrogate cut and a regular cut. If a valid surrogate cut cannot be formed, then either the point on the current deepest (surrogate) cut closest to the centre of E_k in the metric D_k is a solution to the system of linear inequalities (2.2-1) or that system is infeasible.

This iterative procedure described in Krol and Miman [26] which is based on these observations, can be seen as a relaxation method for the determining feasibility of the system of linear inequalities (2.2-1) independent of the ellipsoid algorithm.

(c) PARALLEL CUTS

We consider the system of linear inequalities (2.2-1). If (2.2-1) contains a parallel pair of constraints say $a^T x \leq \beta$ and $-a^T x \leq -\hat{\beta}$ then we can generate a new ellipsoid

E_{k+1} by using both constraints simultaneously.

$$\text{Let} \quad \alpha = \frac{a^T x_k - \beta}{\sqrt{(a^T D_k a)}}$$

$$\hat{\alpha} = \frac{\hat{\beta} - a^T x_k}{\sqrt{(a^T D_k a)}}$$

and suppose that $\alpha \hat{\alpha} = 1/n$ and $\alpha \leq \hat{\alpha} \leq 1$.

Then, formulae (2.2-4) to (2.2-5) with

$$\sigma = \frac{1}{n+1}\left[n + \frac{2}{(\alpha - \hat{\alpha})^2}\left(1 - \alpha\hat{\alpha} - \frac{\rho}{2}\right)\right]$$

$$\tau = \left(\frac{\alpha - \hat{\alpha}}{2}\right)\sigma$$

$$\delta = \frac{n^2}{n^2-1}\left[1-\frac{1}{2}\left(\alpha^2+\hat{\alpha}^2-\frac{\rho}{n}\right)\right]$$

and

$$\rho = \sqrt{4(1-\alpha^2)(1-\overline{\alpha}^2)+n^2(\overline{\alpha}^2-\alpha^2)}$$

generate an ellipsoid that contains the slice

$$\{x \in E_k: \ \hat{\beta} \le a^T x \le \beta\} \text{ of } E_k$$

when $\hat{\beta} = \beta$ ie $a^T x = \beta$ for all feasible x, $\hat{\alpha} = -\alpha$ and we get

$$\tau = \left(\frac{\alpha-\hat{\alpha}}{2}\right)\sigma = \left(\frac{\alpha+\alpha}{2}\right)1 = \alpha$$

$$\sigma = \frac{1}{n+1}\left[n+\frac{2}{(\alpha-\hat{\alpha})^2}\left(1-\alpha\hat{\alpha}-\frac{\rho}{2}\right)\right]$$

$$= \frac{1}{n+1}\left[n+\frac{1}{(2\alpha^2)}\left(1+\alpha^2-\frac{2(1-\alpha^2)}{2}\right)\right]$$

$$= \frac{1}{n+1}\left[n+\frac{1}{(2\alpha^2)}\left(\frac{2\alpha^2}{1}\right)\right] = 1$$

$$\delta = \frac{n^2}{n^2-1}\left[1-\frac{1}{2}\left(\alpha^2+\hat{\alpha}^2-\frac{P}{n}\right)\right]$$

$$= \frac{n^2}{n^2-1}\left[1-\frac{1}{2}\left((\alpha^2+\alpha^2-\frac{2(1-\alpha^2)}{n}\right)\right]$$

$$= \frac{n^2}{n^2-1}\left(\frac{n(1-\alpha^2)+(1-\alpha^2)}{n}\right)$$

$$= \frac{n(1-\alpha^2)}{n-1}$$

That is, rank (D_{k+1}) = rank $(D_k)-1$ and E_{k+1} becomes flat in the direction of a. As in the case of deep cuts Shor and Gershovich [48] were the first to suggest the use of parallel cuts and provide formulae for implementing them. They also provide formulae for circumscribing an ellipsoid (of close to minimum volume) about the region of the unit ball bounded by two or more hyperplanes which intersect in the interior of this ball and whose normals are mutually obtuse.

The formulae for parallel cuts were also derived independently by König and Pallaschke [24] show how to generate a parallel constraint.

$$-a^T x \le -\hat{\beta}$$

which does not cut off any point in E_k that satisfy (2.2-1) and which yields a slice

$$\{x \in E_k: \hat{\beta} \le a^T x \le \beta\}$$

of E_k that has minimum volume.

CHAPTER THREE

THE ELLIPSOID ALGORITHM FOR LINEAR PROGRAMMING

3.0 INTRODUCTION

So far, we have addressed the problem of detecting feasibility of the system of linear inequalities using the ellipsoid algorithm.

In this section we show how the ellipsoid algorithm can be used to solve a linear programming problems. In section 3.1 we discuss the primal and the dual solution in $\mathbb{R}^{m+n}$. In section 3.2 we discuss the Bisection method and in section 3.3 we discuss the sliding objective function method and lastly in the section 3.4 we discuss how to obtain an exact solution for linear programming from approximate solution.

3.1 PRIMAL AND DUAL SOLUTION IN $\mathbb{R}^{m+n}$

We consider a linear programming problem as follows:

Maximize $Z_p = C^T x$

subject to the constraint: ... (3.0-1)

$$A^T x \leq b$$

$$X \geq 0$$

where $C \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and A^T is an m x n matrix. The linear programming problem (3.0-1) is called the primal problem. Then, the dual of the primal problem (3.0-1) is:

$$\text{Minimize } Z_D = b^T y$$

Subject to the constraints: ... (3.0-2)

$$Ay \geq C$$
$$y \geq 0$$

Exploiting Duality theoem (II) of section 1.9(c), the problem (3.0-1) and (3.0-2) can be combined in the form of inequalities as follows:

$$\begin{aligned} A^T x &\leq b \\ -x &\leq 0 \\ -Ay &\leq -C \\ -y &\leq 0 \\ C^T x - b^T y &\leq 0 \end{aligned} \qquad \ldots \quad (3.0\text{-}3)$$

If $(\bar{X},\bar{Y})$ be the solution of system of linear inequalities (3.0-3) then in view of the fundamental duality theorem (theorem (IV) of section 1.9(c)) $\bar{X}$ is the solution of the primal problem (3.0-1) and $\bar{Y}$ is the solution of the dual problem (3.0-2).

Gács and Lovas [5] pointed out that the ellipsoid method can be applied to (3.0-3) to solve (3.0-1) and

(3.0-2). Clearly, a polynomial algorithm for linear inequalities yields in this way a polynomial algorithm for linear programming problems.

3.1.1 DISADVANTAGES TO THIS APPROACH

The ellipsoid algorithm is applied to a system of linear inequalities in $\mathbb{R}^{m+n}$ thus the high dimensionality slows convergence.

In many practical problems the feasible region of (3.0-1) is bounded and explicit bounds are known. Thus the method working only in the primal space can be initialized with an ellipsoid of large but not very large volume speeding convergence. Bounds on the dual variables may be difficult to obtain and hence the initial ellipsoid for (3.0-3) lie in the hyperplane.

$$C^T x = b^T y$$

Moreover even if (3.0-3) is feasible the volume of the feasible set of the perturbed problem will be very small and in such a circumstance the number of iteration is expected to be reasonably very large.

3.1.2 HOW TO OVERCOME THE PROBLEMS OF THIS APPROACH

To overcome these difficulties to some extent one can use a strategy of choosing constraints of (3.0-3) to generate cuts for the ellipsoid algorithm. Since the system (3.0-3) is separated into:

(I) feasibility problem for the primal;

(ii) feasibility problem for the dual and

(iii) Optimality problem for the primal and dual.

Thus, if no cut is base on this last constraint, the matrix D_k defining the ellipsoid E_k remains block diagonal with two blocks corresponding to the x and y variables. When such a matrix D_k is updated, as long as the final constraint is not used as a cut only one of the blocks is updated and only one of x_k and y_k changes. In this case the high dimensionality is not too drastic a problem until feasibility is reached. It seems reasonable to base cuts only on the primal constraints until a primal feasible x is generated; then only the dual constraints until a dual feasible y is generated. Then a cut based on the coupling constraint is used.

Suppose that before perturbation of (3.0-3), L bits are necessary to define the system. Then one can show that using the strategy above if no primal feasible solution is generated in 6n(m+n+1)L steps, then (3.0-1) is infeasible. If a primal feasible solution is generated in k steps but then no dual feasible solution is generated after a further minimum {k+6m(m+n+1)L, 6((m+n)(m+n+1)L-k} steps; then (3.0-

1) is unbounded.

Jones and Marwil [19] present a variant of this approach of simultaneously solving the primal and dual using the complementary slackness conditions as discussed in theorem V of the section (1.9(c)) for (3.0-1) and (3.0-2) to reduce the dimensionality of the problem as iterations are performed.

If α given by (2.4-2) is less than -1 for one of the constraints of (3.0-1) or (3.0-2) then the ellipsoid E_k is contained in the interior of the half space associated with that constraint hence one can conclude that the complementary constraint must be binding at any solution to (3.0-3).

Consequently when such a situation occurs, Jones and Marwil [19] project the current iterate x_k onto this binding constraint and collapse the ellipsoid into its intersection with the constraint. All constraints with $\alpha<-1$ can be temporarily eliminated. When a solution to (3.0-3) excluding these constraints is found. If they are satisfied by that solution then we stop. Otherwise it is necessary to continue iteration after reintroducing any violated constraints into the problem.

3.2 BISECTION METHOD

We consider the system of linear inequalities of the form

$$A^T x \le b$$
$$-x \le 0 \qquad \ldots \quad (3.0\text{-}4)$$
$$-C^T x \le -\xi$$

for various values of ξ.

The method initially applies the ellipsoid algorithm to constraints of (3.0-1) to obtain a feasible solution x. if it exists, if there is none we stop. Thus, we take

$$\xi_* = C^T x_*$$

as a lower bound on the optimal value of (3.0-1).

To obtain the upper bound ξ^* on this value first we check whether the feasible region of (3.0-1) is bounded and contained in the current ellipsoid E_k given by (2.4) then;

$$\xi^* = C^T x_k + \sqrt{C^T D_k C}$$

is such an upper bound.

Otherwise we may apply the ellipsoid algorithm to the constraints of (3.0-3) to obtain a dual feasible solution y^* if one exists, and set $\xi^* = b^T y^*$, if (3.0-2) is infeasible we stop.

From now on each major iteration starts with an interval $[\xi_*, \xi^*]$ of the real line $\mathbb{R}$, that contains the optimal value, where

$$\xi_* = C^T x_*$$

for some known feasible solution x_* and applies the ellipsoid algorithm to (3.0-4) with

$$\xi = \frac{\xi^* + \xi_*}{2}$$

If a feasible solution x_k is generated we set,

$$x_* \leftarrow x_k$$

$$\xi_* \leftarrow C^T x_k$$

and proceed to the next major iteration.

If it is determined that (3.0-4) is infeasible we set,

$$\xi^* \leftarrow \xi$$

and proceed to the next iteration.

The process stop when

$$\xi^* - \xi_*$$

is sufficiently small.

3.2.1 ADVANTAGE OF THE APPROACH

This bisection method has the advantage that it is operating in $\mathbb{R}^n$ (except for possibly one application in $\mathbb{R}^m$). Polynomial time algorithm combining the bisection with ellipsoid method are given by Padberg and Rao [33] for linear programming.

3.2.2 DISADVANTAGE OF THE APPROACH

The system (3.0-4) will be infeasible if ξ is too large and in such a case we used the deep cuts of section 2.4 and the resulting tests for infeasibility to allow early termination in such cases.

It is important to note that:

(1) If a feasible solution x_* has just been generated and if we can start a major iteration with a new ξ greater than the old one then the final ellipsoid of the previous major iteration with centre x_* can be taken as the initial ellipsoid of the new major iteration.

(2) If on the other hand ξ has smaller value than the older one then we can initialize with the last recorded ellipsoid with feasible centre x_*, then the algorithm backtracks. Avoiding such backtracking leads to the another method called sliding objective function method which we shall consider in the next section 3.3.

3.3 SLIDING OBJECTIVE FUNCTION METHOD

We consider the system (3.0-1) then determine the feasible solution x_* to (3.0-1), then we also consider (3.0-4) with

$$\xi \leftarrow \xi_* = C^T x_*$$

Although x_* is feasible in the problem, we may proceed with the ellipsoid algorithm using the cut

$$C^T x \geq C^T x_* = \xi_*$$

Since the next ellipsoid can be defined even when the current iterate lies on the chosen cut. Hence in this method we are always considering feasible system. It follows that whenever a feasible iterate x_* satisfies

$$C^T x_k > C^T x_* = \xi_*$$

then we set $x \leftarrow x_*$ and $\xi \leftarrow \xi_* \leftarrow C^T x_*$ and proceed as above.

This method is probably the most efficient for practical implementation. It always consider feasible systems and never backtracks. All computation takes place in $\mathbb{R}^n$. If the feasible region of (3.0-1) is bounded and known to lie in the initial ellipsoid "E_0" then the upper bound ξ_k^* can be found as

$$\xi_k^* = \text{minimum} \quad \{\xi_{k+1}^*, \; C^T x_k + (C^T D_k C)^{1/2}\}.$$

The computation can stop when ξ_* and ξ^* are sufficiently close.

To improve the performance of the method we set

$$x \leftarrow x_k + \theta S$$

whenever x_k is feasible, here S is a ascent direction (e.g. $S=C$ or $S=D_k C$) and θ is as large as possible so that x is feasible.

The modifications required to obtain a polynomial algorithm using this method are:

First we use the ellipsoid algorithm to determine whether (3.0-1) is feasible.

If so, we next determine whether it is unbounded by applying the ellipsoid algorithm either to the constraints of (3.0-2) or to the system

$$A^T x \le b$$

$$-x \le 0$$

$$-C^T x \le -1$$

any solution to these inequalities gives an unbounded ray for (3.0-1).

Suppose we determine that (3.0-1) was a finite optimal solution.

Let x^{**} be the unknown solution with a value ξ^{**}.

Let assume that the feasible region of (3.0-1) contains a ball

$$B(a_o, r).$$

Otherwise, we can assume this by making suitable perturbations. Then the feasible region also contains the "cone" C with vertex x^{**} and this ball as it base.

For any $\xi < \xi^{**}$

we can easily obtain a large enough lower bound on the volume of

$$C^* = C \cap \{x \in \mathbb{R}^n : C^T x \ge \xi\},$$

o

So that the sliding objective function method for (3.0-1) obtains feasible solutions with objective function values with ϵ of ξ^{**} in number of steps polynomial in L, $|\log 1/r|$ and $\log 1/\epsilon$.

It is important to note that, C^* assumes the role played by ball $B(a_o, r)$ in the inequality case. From these solutions optimal solution can be obtained when ϵ is sufficiently small.

The sliding objective function method was first proposed by Iudin and Nemirovskii [17] and Shor [42]. Grotschel *et al* [12] use it as a tool for demonstrating polynomial equivalence of certain combinatorial optimization problems. Goldfarb and Todd [11] added the refinement of stepping as far as possible in a steepest ascent or Newton like direction, while still remaining feasible, before effecting an objective function cut. Pickel [34] proposes stepping to a vertex by a simplex like approach before effecting an objective function cut.

3.4 TO OBTAIN EXACT SOLUTIONS FROM APPROXIMATE SOLUTIONS

In this section we discuss how exact solutions can be obtained from approximate solution.

ϵ-approximate solution of a LP problem: For $\epsilon > 0$, x is defined to be an ϵ-approximate solution of (3.0-1) if there exist a feasible solution y and an optimal solution $\hat{x}$ of (3.0-1) such that $\|y-x\|\le\epsilon$ and $C^T\hat{x} - C^T x \le \epsilon$.

Suppose that ϕ is a positive integer and $\hat{x}$ is a rational vector of the form

$$\left(\frac{P_1}{q_1},\frac{P_2}{q_2},\ldots,\frac{P_n}{q_n}\right)^T \qquad \ldots \quad (3.0\text{-}5)$$

where P_i q_i are integers and $|q_i|\le\phi$, $i=1,2,\ldots,n$.

Given $x\in\mathbb{R}^n$ such that $\hat{x}$ is in the interior of the ball $B(x, \frac{1}{2\phi^2})$, then $\hat{x}$ is the unique rational vector of the form (3.0-5) in this ball. For

$$y\in B(x, \frac{1}{2\phi^2})$$

$$\Rightarrow \|y - \hat{x}\| \le \frac{1}{2\phi^2}$$

but if y is also of the form (3.0-5) and for some j, $y_j \neq \hat{x}$

then

$$|y_i - \hat{x}_j| \ge \frac{1}{\phi^2}$$

implying that $\|y_j - \hat{x}\| \ge \frac{1}{\phi^2}$.

Therefore if x and $\hat{x}$ are as above, x is known and $\hat{x}$ is not. We can obtain $\hat{x}$ by rounding each component of x to the nearest rational p/q having $|q| \le \phi$ by the method of continued fractions, Niven and Zuckerman [32].

Consider a situation where $\hat{x}$ is an optimal extreme point of the linear programming problem (3.0-1) and x is obtained from the ellipsoid method, Grotschel *et al* [12] point out that one can replace the objective function vector C of (3.0-1) by a perturbed vector.

$$d = \gamma_n + (\gamma_o, \ldots, \gamma_{n-1})^T$$

such that the problem

Minimize $Z = d^T x$

subject to the constraint ...(3.0-6)

$$A^T x \leq b$$

$$X \geq 0$$

Thus (3.0-6) has a unique optimal solution at an extreme point $\hat{x}$ and $\hat{x}$ also solves (3.0-1), for example we can get

$$\gamma = 2^{(L_{n+1}+L+1)}$$

it is important to note that $\log\gamma$ is a polynomial in L and n so that the size of (3.0-6) is a polynomial function of the size of (3.0-1). By Cramer's Rule $\hat{x}$ is of the form (3.0-5)

for ϕ greater than or equal to the absolute value of the largest determinant of any n x n submatrix of the constraint matrix of (3.0-1) in particular we can take $\phi=2^L$.

For sufficiently small $\epsilon > 0$

$$||\hat{x} - x|| < \frac{1}{2\phi^2} \qquad \ldots \quad (3.0\text{-}7)$$

for every ϵ-approximation solution x of (3.0-6). If in addition ϵ can be chosen so that

$$\log\left(\frac{1}{\epsilon}\right) \qquad \ldots (3.0\text{-}8)$$

is a polynomial in n and L then an ϵ-approximation solution x of (3.0-6) can be computed by ellipsoid method in time polynomial in n and L. It follows from the derivation in Grötschel *et al* [10] that one can specify ϵ as a function of n and L and $\|c\|$ so that (3.0-7) and (3.0-8) are satisfied with

$$\phi = 2^L \quad \text{and} \quad \gamma = 2^{L_n + L + 1}$$

one can set

$$\frac{1}{\epsilon} = n^{3/2} 2^{(n^2+2n+2)L+2n+5} + ||C|| n^{1/2} 2^{2nL+3L+5}$$

Since $x_j < 2^L$ for each component x_j of x, the rounding of x by continued fractions requires at most 0[n(P+L)] arithmetic operations each involving number with the most P+L binary digits, where P is the number of binary digits of precision maintained in ellipsoid method.

Continued fractions are also used in Kozlov *et al.* [25] to round an approximate optimal objective function value to the exact optimal value in polynomial time.

CHAPTER FOUR

4.0 COMPARISON WITH OTHER LINEAR PROGRAMMING ALGORITHMS AND CONCLUDING REMARKS

In this chapter we present a comparison of the ellipsoid method with other algorithms for linear programming.

In section 4.1 we compare the ellipsoid method with the simplex algorithm. In order to facilitate a comparison between the ellipsoid method and Karmarkar's algorithm, we first present Karmarkar's algorithm in section 4.2. After a brief presentation of the Karmarkar's algorithm in section 4.2, we compare it with the ellipsoid method in section 4.3. Finally in section 4.4 we present concluding remarks.

4.1 RELATIONS TO THE SIMPLEX METHOD

Like the ellipsoid method one can of course view simplex pivoting in a similar way. If one uses it to find a feasible point satisfying the constraints $Ax=b$, $x\geq 0$, given in standard form, then each simplex pivot (on an infeasible row) can be thought of as an affine transformation from one space of nonbasic variables to another obtained by exchanging one nonbasic and basic variables . Partitioning A and x into basic and nonbasic parts $A=[B|N]$ and $x^T=[x^T_B,$

x^T_N], it is clear that simplex steps are performed until $x_N=0$ is contained in the transformed polytope

$P=\{x_N | x_N \geq 0,\ B^{-1}x_N \leq B^{-1}b\}$ or infeasibility is detected.

Like the simplex method, there are many ways to implement the ellipsoid method. These includes :

(I) the positive definite matrix D_k as described in chapter two, Gács and Lovász [7] Padberg and Rao [33] and Grotschel *et al* [12].

(ii) the matrix J_k which transforms an sphere into the ellipsoid E_k translated to the origin, Shor [43], Khachiyan [21] and Krol and Mirman [26].

(iii) the Cholesky factorization of D_k Jones and Marwill [18]. Goldfarb and Todd [11] and

(iv) the problem data under the transformation induced by J_k Halfin [14], Krol and Mirman [26].

One of the principal computational and practical drawbacks of the ellipsoid method is that it is not possible to implement it and takes advantage of any sparsity in the problem data other than block diagonal structure. To save work, it has been suggested that the ellipsoid and relaxation method be combined into a hybrid algorithm. (Goldfarb and Todd [11] and Telgen [51]). If α is large enough one can simply scale D_k; it, set $\tau=\alpha$, $\delta=1-\alpha^2$ and $\sigma=0$ in (2.2-4) and (2.2-5). If $\alpha \geq 1/n$ the volume ratio is less than

$$(1-\frac{1}{n^2})^{n/2} < e^{-1/2n} \quad ;$$

hence such an algorithm which combines the ellipsoid method with simplex method is proposed in Pickel [34].

4.2 KARMARKAR'S PROJECTIVE METHOD FOR LINEAR PROGRAMMING

As discussed in Karmarkar's [20], Singh [39] and Singh and Singh [41] a linear programming problem in standard form with (n-1) variables can be transformed to the following canonical form.

$$\begin{aligned} \text{Minimize} \quad & Z = C^T X \\ \text{Subject to} \quad & AX=0 \\ & e^T X = 1 \\ & X \geq 0 \end{aligned} \qquad \ldots \quad (4.0\text{-}1)$$

where (i) A is an m x n matrix of integers and rank A=m.

(ii) $C=(C_1,C_2,\ldots,C_n)\in Z^n$

$X=(x_1,x_2,\ldots,x_n)\in \mathbb{R}^n$

$e^T=(1,1,\ldots,1)$ is a vector of ones, and

(iii) the minimum value of the objective function is zero.

Let a>0 be a feasible solution to the problem (4.0-1) and $D=\text{diag}\{a_1,a_2,\ldots,a_n\}$ be a diagonal matrix whose diagonal elements are n components of vector a. We consider a projective transformation T of the simplex

$$S=\{X\in\mathbb{R}^n : X\geq 0,\ e^TX=1\},$$

onto itself defined by

$$X' = T(X) = \frac{D^{-1}}{e^TD^{-1}X} \qquad \ldots(4.0\text{-}2)$$

whose inverse map is given by

$$X = T^{-1}(X') = \frac{DX'}{e^TDX'} \qquad \ldots(4.0\text{-}3)$$

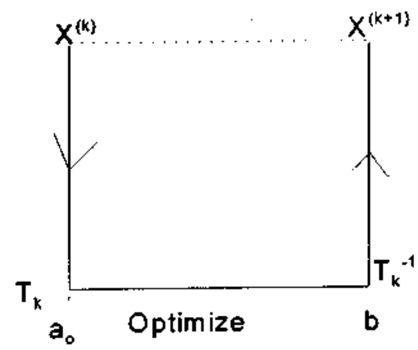
Karmarkar's projective algorithm generates a sequence $X^{(0)}, X^{(1)}, X^{(2)}, \ldots X^{(n)}$ of points in $\mathbb{R}^n$, starting with

$$X^{(0)} = a^{\circ} = \frac{e}{n}$$

At any k^{th} iteration, the point $X^{(k)}$ is mapped to the center a_o of the simplex S and a point b is obtained in the following substeps whose inverse image gives the next point of the sequence.

Step 1:

Let $D=\text{diag}\{X_1, X_2, \ldots, X_n\}$ be a diagonal matrix whose diagonal elements are n components of the current iterate $X^{(k)}$ then the projective transformation T_k is



given by

$$X' = T_k(X) = \frac{D^{-1}X^{(k)}}{e^{T}D^{-1}X^{(k)}}$$

which maps $X^{(k)}$ to the center $a_o = \left(\frac{1}{n}\right)e$ of the simplex s.

Substep 1.1

Let B = $\begin{pmatrix} AD \\ e^T \end{pmatrix}$

Substep 1.2

Project DC orthogonally into null space of B to obtain,

$C_p = [I - B^T(BB^T)^{-1}B]DC$

Substep 1.3

Compute unit vector $\hat{C}$ in the direction of C_p ie

$$\hat{C} = \frac{Cp}{||Cp||}$$

Substep 1.4

Take a step length αr in the direction $\hat{C}$ and obtain

a point $b = a_o - \alpha r \hat{C}$ where $r = \frac{1}{\sqrt{n(n-1)}}$ is the radius of

the largest sphere inscribed in the simplex and "α" is a parameter such that $0<\alpha<1$ Karmarkar [20] suggests to take $\alpha=-\frac{1}{4}$).

Substep 1.5

Apply inverse projective transformation T_k-1 to b, so

that $$X^{(k+1)} = \frac{Db}{e^{T}Db}$$

Step 2

Check for feasibility or unboundedness: since linear objective function C^TX is not invariant under the projective transformation, so Karmarkar [20] has exploited a potential function f(X) defined as

$$f(X) = \sum \ln \frac{C^{T}X}{X_j} \qquad \ldots \quad (4.0\text{-}3)$$

This potential function f(X) is used to measure the progress of the algorithm. We expect a certain decrease γ (which depends on the parameter α) in the potential function f(X) at each iteration of the algorithm. If we do not observe the expected improvement i.e. if

$f(X^{(k+1)}) > f(X^k - \gamma$ then we stop and conclude that the minimum $C^TX>0$. This situation corresponds to the case that the original problem (ie, the standard form linear programming problem from which the canonical form (4.0-1) has been obtained) does not have finite optimum ie, it is either infeasible or unbounded.

Step 3

The iterations stop when the current iterate $X^{(k)}$ meets the required convergence check $C^TX^{(k)} \le 2^{-q}C^TX^{(0)}$ where q is a positive integer.

4.3 COMPARISON BETWEEN ELLIPSOID AND KARMARKAR'S ALGORITHMS

Both for the ellipsoid method and Karmarkar's algorithm the number of iteration to solve a linear programming problem are bounded by some polynomial function of m,n, and L. It has been observed in Karmarkar [20] that the ellipsoid algorithm requires about $0(n^6L^2)$ bits operations whereas Karmarkar's algorithm requires $0(n^{3.5}L^2)$ bits operations. Thus, it has been claimed in Karmarkar [20] that his algorithm is better than the ellipsoid algorithm by a factor of $n^{2.5}$. But it has been reported in Khachiyan's latter work [22] whose algorithm requires only $0(n^4L^2)$ bits operations. Thus the real improvement on the complexity bounds due to Karmarkar [20] is only by $n^{0.5}$ and not $n^{2.5}$ as claimed in Karmarkar [20]. Thus there is only a little advantage to Karmarkar's algorithm from the stand point of theoretical bounds. In practical performance the difference becomes marked. The ellipsoid method appears to require a number of iterations closed to its worst-case bound of $O(n^4L^2)$. On the other hand, a number of studies have established that many variants of Karmarkar's algorithm only take a number of iterations between 20 and 50 to get a very accurate solution and this number appears to grow slowly with n.

In contract to the overwhelming computational superiority of the projective algorithm over the ellipsoid method its theoretical implications are far more limited. Grotschel *et al* [12] and others have use the ellipsoid method to show that certain combinatorial optimization problems are in P (ie have polynomial algorithm) and others are NP-hard (ie are unlikely to have polynomial algorithms). A key to its use is the fact that the ellipsoid method does not need to have all the constraints listed in advances, they can be generated as needed. Thus it is possible to handle problem with an exponential number of constraints. On the other hand, it appears that Karmarkar's algorithm requires all the constraints and variables to be explicitly present, and thus it cannot be used to solve such problems with our present knowledge.

Many researchers have noted that Karmarkar's algorithms and Ellipsoid algorithms use many similar concepts; - ideas from nonlinear programming, geometric motivation and infinite iterative scheme that can be truncated after a polynomial number of step when applied to rational data, with exact optimal solution then available by rounding. However, the details of the two algorithms seem very different. It has been observed in Singh [40] that the heart of each iteration of either algorithms is the solution of a weighted least-squares subproblem and that this subproblems are very closely related. This view point allows further insights into the two methods, in particular suggesting reasons for the very slow convergence of the ellipsoid algorithm compared to apparently very fast convergence of the Karmarkar's algorithm. The weighted least-squares subproblem has other important features. Both the ellipsoid and Karmarkar's algorithm appears at first sight not to provide solutions to the dual linear programming problem, but a closer examination shows that dual solutions are indeed generated during the course of the methods essentially from the least-squares subproblems, Singh [40].

Naturally, optimal dual solutions can be generated from optimal primal solution at termination. However approximate dual solution at intermediate stage are very useful in guaranteeing the quality of current solution or certifying infeasibility.

In singh [41] a similarity between the inequalities governing volume reduction in the ellipsoid method for linear inequalities and convergence in Karmarkar's projective algorithm for linear programming has been established. In fact the RHS of the volume reduction inequality

$$\frac{v(E_k)}{v(E_o)} < e^{-\frac{k}{2(n+1)}}$$

And the convergence inequality

$$\frac{C^T X^{(k)}}{C^T X^{(0)}} < e^{-\frac{k\gamma}{(n+1)}}$$

becomes identical when, we achieve $\gamma=\frac{1}{2}$.

4.4 CONCLUDING REMARKS

It becomes clear from the comparison of the ellipsoid method with the simplex and Karmarkar's algorithms that the ellipsoid algorithm is unable to take full advantage of the sparsity of the problem data, where the simplex and Karmarkar's algorithms fully exploit the sparsity of the problem data. Although relatively little computational experience with the ellipsoid algorithm is available, general consensus is that it is not a practical alternative to the simplex and Karmarkar's algorithms. A list of few papers reporting computational results can be found in Wolfe [56]. Dantzig [4] on the basis of computational results has estimated that the number of iterations to solve a linear programming problem lies between m and 3m, where m is the number of row in constraint matrix of the liner programming problem. This empirical upper bound on the number of iterations is quite small in comparison to the upper bound $O(n^4L^2)$ as established in the ellipsoid method [22]. This situation happen, with most of the linear programming problems of moderate sizes encountered in real life. Klee ad Minty [23] have identified a class of linear programming problems which compels the simple method to take exponential number of iterations. But the examples constructed by Klee and Minty [23], contrasting average behaviour of the simplex method, though constructed Mathematically inherit an essential artificiality which hardly can have correlate in real life. Thus it may be generally accepted that unless some further break through in its implementation aspects take places, it cannot plausibly be said that the ellipsoid algorithm can be a successful alternative of the simplex and Karmarkar's algorithms. However as discussed in the foregoing sections despite showing some unsurmountable difficulties in its practical applicability, the overall impact of the ellipsoid method especially on theoretical developments, combinatorial optimization and in handling problems with an exponential number of constraints cannot be denied. It is a strongly believed that the dilemma "theoretically significant but practically improverished" indicates some serious reconsideration of various complexity measures.

REFERENCES

[1] Agmon, S.," The Relaxation Method for Linear Inequalities". CAN J. Math. B. Pp. 282-392, 1954.

[2] Bazaraa, M.S. and Setty, C.M., "Non-linear programming theory and algorithm", John Wiley and Sons (1979).

[3] Bland, R.,Goldfarb, D.and Todd, M.J., "The Ellipsoid Method: A survey", Operations Research 26(16) (1981), pp. 1039-1091.

[4] Dantzig, G.B., "Linear Programming and Extensions", Princeton University Press, Princeton, N.J. (1963).

[5] Dennis, J.E. and More, J.J., "Quasi-Newton Methods, Motivation and theory", SIAM Review 19(1) (1977), pp. 46-89.

[6] Ermolev "Methods of Solution of nonlinear Extremal problems" Kobernetika 2(4), 1-17 (translated in Cybernetics 2(4), 1-14, 1966).

[7] Gács, P. And Lovász, L., (1981). "Khachiyan's Algorithm for Linear Programming", Math. Program Stud., 14(1981), pp. 61-68.

[8] Goffin, J.L., "Acceleration in the Relaxation Method for linear inequalities and subgradient optimization", Working paper 78-101(1978). Faculty of Management, Mc. Gill University, Montreal, Canada.(Also in the proceedings of a task force on non differentiable optimization) held at HASA, Laxenburg. Astria December, 1978.

[9] Goffin, J.L., "On the Non-polynomiality of the Relaxation Method for system of Inequalities", Faculty of Management Mc. Gill University, Montreal Quebec (November, 1979).

[10] Goffin, J.L., "Convergence of a Cyclic shsor-Khachiyan Method for systems of linear equalities", working paper No. 79-54, Faculty of Management Mc. Gill University Montreal, Quebec (December, 1979).

[11] Goldfarb, D.and Todd, M.J., "Modifications and Implementations of the shor-Khachiyan Algorithm for Linear Programming", Technical Report no 406,

Department of Computer Science, Cornell University, Ithaca, N.Y. (January, 1980).

[12] Grostschel,M., Lovász, L. and Schrijver, A., "The Ellipsoid Method and its consequences in combinatorial optimization", Combinatorica 1(1981) pp. 169-197.

[13] Hadley, G., "Linear Programming", Addison Wesley Reading, Mass, 1962.

[14] Halfin, S., "The sphere method for Khachiyan's algorithm", Bell Telephone Laboratories Holmdel, N.G. March, 1980.

[15] Hoffman, A.J. "On Approximate solutions of systems of linear inequalities", J. Res. Natl. Bur stand 49(1952), pp. 263-265.

[16] Iudin, D.B. and Nemirovskii, A.S., "Evaluation of the informational complexity of Mathematical programming problems", Matekona translation of Russian and East European Math. Economics 13(1976), 3025, Winter 76-77.

[17] Iudin, D.B. and Nemirovskii, A.S., "Informational complexity and Effective Methods of solution for convex Extremal problems", Ekonomikai Matematicheskie Metody. 12(1976) 357-369 (translated in Matekon, Translations of Russian and East European Math. Economics 13(1976) 25-45, Spring 77).

[18] Jones, P.C. and Marwil, E.S., "A Variant of Khachiyan's Algorithm for Linear Programming", E.G. and G. Idaho, Inc. P.O.B., 1625, Idaho Falls Idaho 83415 (November, 1979).

[19] Jones, P.C. and Marwil, E.S., "A Dimensional Reduction variant of Khachiyan's Algorithm for Linear Programming Problems", E.G. and G. Idaho Inc., P.O.B. 1625, Idaho Falls, Idaho 83414 (January, 1980).

[20] Karmarkar, N., "A new polynomial-time Algorithm for Linear Programming", Combinatorica 4(4) (1984), pp. 373-395.

[21] Khachiyan, L.G., "A Polynomial Algorithm for Linear Programming", Doklady Akad, 20(1979), pp. 191-194.

[22] ______________, "A polynomial Algorithm in Linear Programming", USSR Computational Mathematics and Mathematical Physics 20(1980) PP 53-72.

[23] Klee, V and Minty, G.L., "How good is the simplex algorithm?", in O. Shisha ed. Inequality III, Academic Press, New York (1972), pp. 159-175.

[24] König, H. And Pallaschke, D., "On Khachiyan's Algorithm and Minimal Ellipsoids", Numerische Mathematik 38, (1981), pp, 211-223.

[25] Kozlov, M.K., Tarasov, S.P. and Khachiyan, L.G., "Polynomial Solvability of convex Quadratic Programming", Doklady Akademiia Nuuk SSSR 243 (1979) (tranlated in Soviet Mathematics Doklady 20(1979), pp. 1108-1111).

[26] Krol, Y.and Mirman, B., "Some Practical Modifications of Ellipsoid Method for Linear Programming problems", (undated reviewed Jan. 1980). Arcon Inc. Boston.

[27] Lemarechal, C., "An Extension of Davidson Methods to Non-Differentiable problems", Math. Program Stud. 3(1975) pp. 95-109.

[28] Levin, A.N., "On an Algorithm for the Minimization of convex Functions", Doklady Akademiia Nauk. SSSR 160 (1965), pp. 1244-1247. (Translated in Soviet Mathematics Doklady 6(1965), pp. 286-290, 1965).

[29] Motzkin, T. and Schoenberg, I.J., "The Relaxation Method for Linear inequalities", Can. J. Math. 6(1954), pp. 393-404.

[30] Newmann, D.J., "Location of the Maximum on Unimodal Surfaces", J. Assoc. Comput. Mach. 12(1965), pp. 395-398.

[31] Nemirovskii, A.S. and Indin, D.B., "Optimization Methods Adapting to the "Significant" Dimension of the problem", Automation and Remote control 38(4)(1977), 513-524).

[32] Niven, I.and Zuckerman, H.S., "An Introduction to the theory of numbers", John Wiley and Sons New York, 1966.

[33] Padberg, M.W. and Rao, M.R., "The Russian Method for Linear inequalities", Gradnate School of Business Administration, New York University New York (Jan., 1980).

[34] Pickel, P.F., "Some improvements to Khachiyan's Algorithm in Linear Programming". Math. Dept. Polytechnic Institute for New York (Dec., 1979).

[35] Polyak, B.T., "A General Method for solving Extremum problems", Doklady Akademiia Nauk SSSR 174(1967), pp. 33-36 (translated in Soviet Mathematics Doklady 8, pp. 593-597, 1976).

[36] ____________, "Minimization of unsmooth Functionals", Zhurnal Vychisch tel'nol Matematiki I Matematicheskoi Fiziki 9(1969), pp. 509-521 (translated in USSR Computational Mathematics and Mathematical Physics 9, pp. 14-29, 1969).

[37] ____________, "Subgradient Methods", A survey of Soviet Research, In Nonsmooth optimization 11ASA Proceedings, Vo. 3c. (1978) Lemarechal an dR. Mifflin eds.). Pergamon Press Oxford.

[38] Singh, J.N., and Singh, D., "Some remarks on Karmarkar's algorithm for Linear programming", J. BMS, Vol. 11(1987/88) pp. 86-90.

[39] ____________, "Convergence in Karmarkar's algorithm a review", Int. J. Maths. Educ. Sci. Technol. 1994,Vol. 25 No. 3, pp. 333-341.

[40] Singh, J.N., "Weighted least squares problems in the ellipsoid Algorithm and projective methods", ACTA, CIENCIA INDICA Vol. 21(3), 1995, pp. 383-385.

[41] Singh, J.N., "A comparison of two recent polynomial algorithms for linear programming", Mathematics & Computer Education Vol. 31 No. 2, pp. 167-175, 1997.

[42] Shor, N.Z., "On the structure of Algorithms for the Numerical solution of optimal planning and Design", Problems dissertation, Cybernetics Institute, Academy of Science of the U.K. rainian SSR Kiev (1964).

[43] ___________, "The Rate of Convergence of the

Generalized Gradient Descent Method", Kibernetika 4(3) (1968),pp. 98-99 (translated in Cybernetics 4(3), pp. 79-80, 1968).

[44] ____________, "Utilization of the operation of spade Dialation in the minimization of convex functions", Kibernetika 6(1)(1970) pp. 6-12 (translated in Cybernetics 6(1) (1970), pp. 7-15).

[45] ____________, "Convergence Rate of the Gradient Descent Method with Diallation of the space", Kibernetika 6(2) (1970), pp. 80-85 (translated in Cybernetika 6(2) (1970), pp. 102-108).

[46] ____________, "Cut-of Method with space Extension in convex programming problems", Kibernetika 13(1) (1977), pp. 94-95 (translated in Cybernetics 13(1)(1977), pp. 94-96).

[47] ____________, "New Development Trends in Non-differentiable Optimization", Kibernetika 13(6)(1977), pp. 87-94 (translated in Cybernetics 13(6)(1977), pp. 881-886).

[48] Shor, N.Z. and Gershovich, V.I., "Family of Algorithms for solving convex programming problems", Kibernetika 15(4)(1979), pp. 62-67 (translated in Cybernetics 15(4), 1979, pp. 502-507).

[49] Shor, N.Z. and Znur Benko, N.G., "A Minimization Method utilizing the operation of space expansion in the Direction of the Difference of two successive Gradients", Kibernetika 7(3)(1971), pp. 51-59 (translated in Cybernetics 7(3), (1971)).

[50] Skokov, V.A., "Note on minimization Methods employing space stretching", Kibernetika 4(4)(1974), pp. 115-117 (translated as Cybernetics 4(4)(1974), pp. 689-692).

[51] Telgen, J., "On Relaxation Methods for systems of linear inequalities", Technical Report Management Science Program University of Tenn (Jan., 1980).

[52] Todd, M.J., "Some Remarks on the Relaxation Method for linear inequalities", Technical Report 419(1979), School of Operations Research and Industrial Engineering, Cornell University Ithaca, N.Y.

[53] ___________, "Minimum Volume Ellipsoid Containing part of a given Ellipsoid", Technical Report No. 468 (1980), School of Operations Research and Industrial Engineering, Cornell University, Ithaca, N.Y.

[54] ____________, "Polynomial Algorithm for Linear Programming" Advances in Optimization and Control (H.A. Eiselt and G. Pederzoli, eds.) Berline Springer Verlay (1988), pp. 49-66.

[55] Vavasis, S. and Ye, Y., "A Primal-dual interior point method whose running time depends only on the constraint matrix. Math. Programming 74, pp. 79-120, 1996.

[56] Wolfe, P. "A method of conjugate subgradients for minimizing Non-differentiable Functions", Math. Program. Stud. 3(1975), 145-173.

[57] ___________, " Bibliography for the Ellipsoid Algorithm", IBM Research Centre Report (July 7, (1980)).

[58] Yinue, Ye, "Karmarkar's Algorithm and the Ellipsoid Method", Operations Research Letters 6(4), pp. 177-182., 1987

APPENDEX II

B	C_B	X_B	$Y_1 (= \alpha_1)$	$Y_2 (= \alpha_2)$	...	$Y_k (= \alpha_k)$	...	$Y_n (= \alpha_n)$	$Y_{n+1} (\beta_1)$	$Y_{n+2} (\beta_2)$	...	$Y_{n+m} (\beta_m)$	Mini Ratio
y_{n+1}	$c_{B_1} = 0$	$x_{B_1} = b_1$	$y_{11} = a_{11}$	$y_{12} = a_{12}$	...	$y_{1k} = a_{1k}$	...	$y_{1n} = a_{1n}$	1	0	...	0	
y_{n+2}	$c_{B_2} = 0$	$x_{B_2} = b_2$	$y_{21} = a_{21}$	$y_{22} = a_{22}$	...	$y_{2k} = a_{2k}$	...	$y_{2n} = a_{2n}$	0	1	...	0	
y_{n+r}	$c_{B_r} = 0$	$x_{B_r} = b_r$	$y_{r1} = a_{r1}$	$y_{r2} = a_{r2}$	...	$y_{rk} = a_{rk}$							
y_{n+m}	$c_{B_m} = 0$	$x_{B_m} = b_m$	$y_{m1} = a_{m1}$	$y_{m2} = a_{m2}$	...	$y_{mk} = a_{mk}$	...	$y_{mn} = a_{mn}$	0	0	...	1	
		x_j	x_1 0	x_2 0	...	...	...	x_n 0	x_{n+1} b_1	x_{n+2} b_2	...	x_{n+m} b_m	
$= C_B X_B = 0$		c_j	c_1	c_2	...	...	...	c_n	0	0	...		
		Δ_j											

Starting Simplex Table

A[redacted]NDEX II

		c_j^*	c_0^*	c_1^*	c_2^* . . .	c_n^* . . .	c_{n+m}^*	
B_1	$C_{B_1}^*$	$X_B^{(1)}$	$Y_0^{(1)}$	$Y_1^{(1)}$	$Y_2^{(1)}$ . . .	$Y_n^{(1)}$ . . .	$Y_{n+m}^{(1)}$	θ Mini $\left\{\frac{X_B^{(1)}}{Y_k^{(1)}}\right\}$
$Y_{n+1}^{(1)}$	— 1	b_0						
$Y_{n+2}^{(1)}$	— 1							
⋮	⋮							
$Y_{n+m}^{(1)}$	— 1							
$Z^* = C_{B_1} X_B^{(1)}$		x_j	. . .	. . .	. . .	. . .	. . .	
		Δ_j^*	. . .	. . .	. . .	. . .	. . .	
$Z_w = \sum_{j=0}^{m} b_j w_j$		w_{sj}						

Starting Primal dual Table