2 LINEAR PROGRAMMING AND ALGORITHMS

2.1 Introduction

2.1.1 Properties of solution to L.P. Problems

Linear Programming problems deal with the allocation of limited resources to meet desired objectives. The symbolic model of a LP problem consists of a linear objective function and a set of linear constraints.

Stated analytically, the general linear programming problem has the following mathematical structure. There are a number of constraints of the type

\[
\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad \text{for} \quad i = 1, 2, \ldots, m \quad (2.1.1)
\]

\[
x_j \geq 0 \quad \text{for} \quad j = 1, 2, \ldots, n \quad (2.1.2)
\]

and a systematic search is made for a nonnegative vector \( X \) which extremizes a linear objective function

\[
Z = \sum_{j=1}^{n} c_j x_j \quad (2.1.3)
\]

The \( b_i \)'s in the constraints (2.1.1) are the available resources and the parameter \( c_j \) is the contribution/cost coefficient associated with unit output of activity \( x_j \). The variables \( x_j \) which satisfy the constraints (2.1.1) are the levels of different activities that can be realised. The problem is to find that vector \( X \) which extremizes the objective function (2.1.3).
As such an optimal solution to the LP problem is a vector X which extremizes (2.1.3) and satisfies (2.1.1) and (2.1.2). A basic solution to the LP problem is a solution obtained by setting (n-m) variables to zero and solving for the remaining m variables, provided that the determinant of the coefficient matrix of these m variables is non-zero. These m variables are called the basic variables.

The solution to a LP problem is obtained by using a tool called the simplex method. An improved and efficient version for solving large scale LP problems on a digital computer is the revised simplex method. The revised simplex method used the same basic principles of the ordinary simplex method. But at each iteration, the entries in the entire tableau are not updated. The necessary information to move from one basic feasible solution to another is directly generated from the original set of matrix coefficients.

Later on, techniques such as multiple column selection, suboptimization procedure, crashing, advantageous starting basis (ab initio), multiplex method for primal and dual etc. were developed to reduce computations. Algorithms to solve LP problems in a finite number of steps have been developed in recent years [1,21]. Some heuristic approaches were tried by some researchers to solve LP problems [42,28].
2.2 Simplex Procedure

2.2.1 Algorithm

The given problem may be modelled as

\[ \text{Extremize } Z = \sum_{j=1}^{n} c_j x_j \]

subject to

\[ \sum_{j=1}^{n} a_{ij} x_j \leq b_i ; \quad i = 1 \text{ to } m \]
\[ x_j \geq 0 ; \quad j = 1 \text{ to } n \]

The problem is then converted into an equivalent problem consisting of linear equations by adding surplus/artificial variables. The simplex algorithm then addresses itself to extremizing \( Z \), satisfying the constraints.

The optimal solution to a linear program lies at an extreme point on the periphery of the constraint set. A basic solution corresponds to a vertex which is a meeting point of two or more number of constraints. The simplex algorithm moves from one basic feasible solution to another of improved objective function value. The problem of starting with an initial basic solution is resolved by starting with an identity matrix as basis. To get an identity matrix artificial vectors are added, if necessary.

The simplex algorithm solves the given equivalent problem in two distinct phases. In phase I, the algorithm minimizes an objective function.
where $R_i$ are the artificial variables. The artificial variables are then driven to zero value. If at the end of I, some artificial variables exist in the basis with a non-zero value, then the problem has no solution. If they exist in the basis at zero value, then there are possible redundancies in the problem. In phase II, the given objective function is minimised.

2.3 The revised simplex procedure

The revised simplex method saves both storage and computational time. Unlike the original simplex method, only the inverse of the current basis is maintained to generate the next inverse. All other quantities except $X_B$ are computed from their definitions as and when necessary. Even though, $X_B$ can be computed from its definition, it is more economical to transform it at each stage. When we have to determine the reduced cost coefficients, we simply determine

$$(z_j - c_j) = C_B S^{-1} p_j - c_j$$

If all $(z_j - c_j) \geq 0$, then optimal solution is attained.

2.3.1 Detailed Algorithm

Again taking the standard LP model

Extremize $Z = CX$
subject to $AX \leq P_0$

$X \geq 0$  \hspace{1cm} (2.3.1)

where

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

$$P_0 = \begin{bmatrix}
b_1 \\
b_2 \\
\vdots \\
b_m
\end{bmatrix}$$

Let the columns corresponding to the matrix $A$ be denoted by $P_1, P_2, P_3, \ldots, P_n$ where

$$P_1 = \begin{bmatrix}
a_{11} \\
a_{21} \\
\vdots \\
a_{m1}
\end{bmatrix}$$

and

$$P_n = \begin{bmatrix}
a_{1n} \\
a_{2n} \\
\vdots \\
a_{mn}
\end{bmatrix}$$

Let the vector $X$ be partitioned as

$$X(n \times 1) = \begin{bmatrix}
x_B \\
x_N
\end{bmatrix}$$

where $X_B$ corresponds to the basic variables and $X_N$ to the non-basic variables.
The revised simplex procedure solves repeatedly a set of linear algebraic equations of the form

\[ B \mathbf{X}_B = \mathbf{P}_0 \]  

(2.3.2)

and finds the value of the objective function

\[ Z = C_B \mathbf{X}_B \]  

(2.3.3)

The set of equations (2.3.2) and (2.3.3) may be put in matrix notation as

\[
\begin{bmatrix}
1 & -C_B \\
0 & B
\end{bmatrix}
\begin{bmatrix}
Z \\
\mathbf{X}_B
\end{bmatrix}
= 
\begin{bmatrix}
0 \\
\mathbf{P}_0
\end{bmatrix}
\]  

(2.3.4)

Let

\[ M = \begin{bmatrix} 1 & -C_B \\ 0 & B \end{bmatrix} \]

Rewriting the equation (2.3.4) as

\[ M \mathbf{X} = \begin{bmatrix} Z \\ \mathbf{X}_B \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{P}_0 \end{bmatrix} \]

The solution vector in terms of the matrix \( M \) is

\[
\begin{bmatrix}
Z \\
\mathbf{X}_B
\end{bmatrix} = M^{-1} \begin{bmatrix}
0 \\
\mathbf{P}_0
\end{bmatrix}
\]  

(2.3.5)

\( M^{-1} \) exists if and only if the basis matrix \( B \) is nonsingular. Hence

\[ M^{-1} = \begin{bmatrix} 1 & C_B B^{-1} \\ 0 & B^{-1} \end{bmatrix} \]  

(2.3.6)

The solution vector is given by
\begin{align*}
\begin{bmatrix}
Z \\
X_B
\end{bmatrix} &= \begin{bmatrix}
1 & C_B B^{-1} \\
0 & B^{-1}
\end{bmatrix} \begin{bmatrix}
0 \\
P_0
\end{bmatrix} \\
&= \begin{bmatrix}
C_B B^{-1} & P_0 \\
B^{-1} & P_0
\end{bmatrix}
\tag{2.3.7}
\end{align*}

The solution is based mainly in identifying $B^{-1}$ for the current iteration.

The basis matrix $B$ is different from the previous or succeeding basis matrix by only one column and so is the $M$ matrix. Let the matrices $M_c$ and $M_n$ correspond to the current and next iterations of the revised simplex procedure. The $M_n^{-1}$ may be obtained from $M_c^{-1}$ using linear algebra and this eliminates the computation for direct inversion.

The procedure for determining $M_n^{-1}$ from $M_c^{-1}$ is summarised below.

Let the identity matrix $I_m$ be represented as

\begin{equation}
I_{m+1} = (e_1, e_2, e_i, \ldots, e_{m+1})
\end{equation}

where $e_i$ is a unit vector with a unit element at the $i$th place and the rest zero. Let $x_j$ and $x_r$ be the entering and leaving variables at the start of any iteration. The $M_n^{-1}$ can then be computed using the relationship

\begin{equation}
M_n^{-1} = E_c M_c^{-1}
\end{equation}

(2.3.8)
where the transformation matrix $E_0$ has $m$ unit vectors and only one non-unit vector corresponding to the entering variable.

\[
\begin{bmatrix}
0 \\
0 \\
\vdots \\
1 \\
0 \\
\vdots \\
0
\end{bmatrix}
\begin{bmatrix}
a_{o0} \\
a_{1j} \\
a_{r+1,j} \\
\vdots \\
a_{mj}
\end{bmatrix}
\begin{bmatrix}
-l/a_{rj} \\
-l/a_{rj} \\
1/a_{rj} \\
\vdots \\
-l/a_{rj}
\end{bmatrix}
\]

leaving vector  entering vector  multiplier

where $a_{o0} = z_j - c_j$

\[
a_{ij} = -\frac{1}{a_{rj}} p_j; \quad i = 1 \ldots m
\]

The entering vector is multiplied by the multiplier to generate the non-unit column ($\mu$) of the $E_0$ matrix:

\[
\eta = \begin{bmatrix}
-a_{o0} \\
-a_{rj} \\
-a_{1j} \\
-a_{rj} \\
+ l/a_{rj} \\
\vdots \\
-a_{mj} \\
-a_{rj}
\end{bmatrix}
\]
The $E_0$ matrix is now formed as

$$E = [e_1, e_2, \ldots, e_r, \eta, e_{r+2}, \ldots, e_{m+1}]$$

where $e_1, e_2, \ldots, e_r, e_{r+1}, \ldots, e_{m+1}$ are all unit vectors and $\eta$ alone is the non-unit vector. Thus the $M_n^{-1}$ is constructed using $M_c^{-1}$ and the transformation matrix $E_0$.

The various steps involved in the revised simplex procedure are as follows:

**Step 1:** Determination of the entering vector $P_j$

$$(z_j - c_j) = C_B B^{-1} P_j - c_j = (1, C_B B^{-1}) \begin{bmatrix} -c_j \\ P_j \end{bmatrix} \quad (2.3.9)$$

The most promising vector enters the basis. Otherwise, if all $(z_j - c_j) \geq 0$, then the optimal solution is attained.

**Step 2:** Determination of the leaving vector $P_k$. When the entering vector is $P_j$ and the current basis matrix is $B_c$, the leaving vector must correspond to

$$\Theta = \min_k \frac{B_c^{-1} P_0}{\alpha_{kj}} \quad (2.3.10)$$

where $\alpha_{kj} = B_c^{-1} P_j$ and

$k = 1, 2, \ldots, n$

If all $\alpha_{kj} \leq 0$ then the problem has no bounded solution.
Step 3: Determination of the next basic solution. The \( E_0 \) matrix is constructed as explained above and the \( M_n^{-1} \) is computed by using (2.3.8) and the solution vector is defined by (2.3.5). Thus \( B_n^{-1} \) is expressed as a function of \( B_c^{-1} \) and the processing is returned to Step 1.

This procedure is repeated until the optimal solution is reached. The drawback of this procedure is that it does not allow mass pivoting at any stage during the computational process.

2.3.2 Revised simplex and linear independence

A nonsingular matrix \( B \) alone ensures a basic solution for an LP problem, in the \( m \) dimensional requirement space. A square matrix of dimension \((m \times m)\) is nonsingular only if it has a set of \( m \) linearly independent vectors. A set of vectors \( P_1, P_2, \ldots, P_m \) are said to be linearly independent, if for all real \( \lambda_j \)

\[
\sum_{j=1}^{m} \lambda_j P_j = 0 \tag{2.3.11}
\]

This is achieved only if all \( \lambda_j = 0 \), where \( \lambda_j \)'s are scalars if

\[
\sum_{j=1}^{m} \lambda_j P_j = 0
\]

for some \( \lambda_j \neq 0 \), then the vectors are said to be linearly dependent.
The conventional simplex method ensures the linear independence of the entering vector. This should be true because, the starting basis matrix differs from the initial basis matrix only in one column. As long as the pivot element of the entering vector is non-zero, the new basis matrix is bound to be non-singular. The justification for this is that $E_0$, the transformation matrix is none other than the inverse of the matrix with $m$ unit vectors and only one unit vector, whose pivot element is non-zero as otherwise it cannot enter the basis. This particular property is formalised in linear algebra as follows:

The transformation matrix $E_0$ is the inverse of a matrix

$$H = \begin{bmatrix}
1 & 0 & \cdots & (z_j-c_j) & \cdots & 0 \\
0 & 1 & \cdots & a_{ij} & \cdots & 0 \\
\vdots & & \ddots & \vdots & & \vdots \\
0 & \cdots & & a_{rj} & \cdots & 0 \\
0 & 0 & \cdots & a_{mj} & 1
\end{bmatrix}$$

The value of the determinant of $H$ is the pivot element $a_{rj}$ itself. Hence the inverse of $H$ is

$$E_0 = H^{-1} = \begin{bmatrix}
1 & 0 & \cdots & - (z_j-c_j) & \cdots & 0 \\
0 & 1 & \cdots & -a_{ij}/a_{rj} & \cdots & 0 \\
\vdots & & \ddots & \vdots & & \vdots \\
0 & 0 & \cdots & 1/a_{rj} & 0 & \cdots \\
0 & 0 & \cdots & -a_{mj}/a_{rj} & 1
\end{bmatrix}$$
Considering the following matrix

\[
W = \begin{bmatrix}
1 & 0 & \ldots & a_{ij} & \ldots & 0 & 0 \\
0 & 1 & \ldots & a_{2j} & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & a_{rj} & \ldots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & \ldots & a_{mj} & \ldots & 0 & 1
\end{bmatrix}
\]

If the columns of \( W \) are to be linearly independent then no set of \( m \) scalars whose one or more values are all other than zero could be found

\[
\lambda_1 + \ldots + \lambda_r a_{1j} = 0 \\
\lambda_2 + \ldots + \lambda_r a_{2j} = 0 \\
\vdots \\
+ \lambda_r a_{rj} = 0 \\
\vdots \\
+ \lambda_r a_{mj} + \ldots + \lambda_m = 0
\]

As evidenced from the above, \( \lambda_r = 0 \) if \( a_{rj} \neq 0 \).

This when substituted in the other equations, makes all the \( \lambda_i = 0, \; i = 1, 2, 3 \ldots r-1, r+1 \ldots m \). Hence \( W \) is a non-singular matrix. The transformation matrix \( E_0 \) used to get the next inverse of the basis matrix from the current inverse is none other than the inverse of the matrix \( W \) whose non-singularity has just been established. Whatever is applicable to the basis matrix \( B \) is equally applicable to the matrix \( M \). Hence, when the inverse of the basis matrix \( B \) exists, the inverse of \( M \) also exists.
2.4 **Revised simplex and the variant**

A variable which makes the rate of change of the objective function very rapid is selected as the entering variable. Alternatively, a variable which gives rise to the greatest change in the objective function is selected as the entering variable. This needs a modification in Step 1 of the previously explained revised simplex method.

**Step 1:** \( \lambda_j = \max/\min [\beta_j(z_j - c_j)] \)  

where  

\[ \beta_j = \min_i \left[ \frac{B^{-1}p_i}{\alpha_{ij}} \right] \quad \text{if } \alpha_{ij} > 0 \]  

where \( \alpha_{ij} = [B^{-1}p_i]_i \)

The rest of the steps are the same as per the revised simplex procedure.

2.5 **Multiplex Technique** [36,37]

The multiplex method selects more than one variable to enter a basis, at a time. This method cuts across the constraint set in search of a vertex unlike the conventional simplex procedure which is edge directed and searches only adjacent vertices for solution.

The steps of this algorithm are the following:
Step 1: Select a point in the interior of the feasible region, that is, such a combination of the values of the basis variables as will make all the variables effectively positive.

Step 2: From this initial point move by some method in a direction which is influenced by the direction of the preference vector. The logarithmic potential technique can be used to decide about the direction of movement.

Step 3: Determine the value of $\lambda$ for which it is largest and the solution is feasible. This will give the optimum solution. Otherwise make the new point as the starting point and go to Step 2.

At every point, selection of the number of variables which is independent is arbitrary and hence the number of the variables plays a crucial role in determining the optimal solution.

2.6 Polynomial Algorithms

2.6.1 Kachiyan's polynomial algorithm is a procedure for deciding whether the system of inequations given by

$$AX \leq b \tag{2.6.1}$$

is consistent. In addition, if the system is consistent, it finds the coordinates of point satisfying all the inequalities or it at least determines them within a small margin of errors. For a system given by the equation (2.6.1) let
\[ L = \left[ \sum_{i=1}^{m} \sum_{j=1}^{n} \log_2(|a_{ij}| + 1) + \sum_{i=1}^{m} \log_2(|b_i| + 1) + \log_2 mn \right] + 1 \]

where \([x]\) denotes the greatest integer less than \(x\). The execution of the algorithm involves \(N = 16L^2\) iterations.

The principle of the algorithm is like the traditional way of catching fish using a net. Casting the net over such a large region that some of what is wanted must be inside. Then gradually decreasing the volume of the net. When the volume is sufficiently reduced it becomes clear whether or not anything has been caught.

Similar to the above analogy a large ellipsoid is formed in the beginning. Depending upon the discrepancies in the solution vector, the size of the ellipsoid is gradually shrunk. When there is no discrepancy in the solution vector, the optimal solution is reached.

2.6.2 Quadratic Programming Problems [29]

Among the class of non-linear programming problems, an important subclass of problem is the quadratic programming problem.

Since the Khun-Tucker conditions are a set of linear inequalities for this class of problems, methods based on the simplex algorithm have been found to be quite efficient in solving them. The multiplex algorithm is several times faster than the simplex even for medium size LP problems. This algorithm may be extended to solve
quadratic and other NLP problems which employ simplex method repetitively.

2.7 Conclusions

The various algorithms available for solving linear programming problems were highlighted. The lack of fixity in these algorithms has led to the wide variety of procedures like double pricing etc. but such efforts have been constantly discouraged on the assumption that these methods which will bring more than one variable into the basis in each iteration may not lead to any saving in computational effort. In the next chapter an algorithm which will bring more than one variable in each pass is discussed.