

## Chapter 2

# LINEAR PROGRAMMING PROBLEMS

### 2.1 Introduction

Linear Programming is the branch of applied mathematics that deals with solving optimization problems of a particular functional form. A linear programming problem consists of a linear objective function (of decision variables) which is to be minimized or maximized, subject to a certain set of linear constraints on decision variables. The constraints are usually in the form of linear inequalities of decision variables used in the objective function. Linear programming is a relatively recent mathematical discipline. The Russian mathematician Leonid V. Kantorovich formulated the first problem in linear programming in 1939. This was what is now known as the transportation problem. The English economist George Stigler (1945) described a problem of determining an optimal diet as a linear programming problem. The simplex algorithm was developed by the American mathematician George B. Dantzig in 1947, and the theory of duality was developed by John von Neumann the same year i.e. in 1947. It was also Dantzig who coined the name "Linear Programming" to this class of problems. In 1972, Klee and Minty showed that the simplex method has exponential computational complex-

ity. In 1979, Khachiyan showed that linear programming problems can be solved by polynomial time algorithms. Karmarkar, in 1984, described his interior-point algorithm and claimed that it is a polynomial-time algorithm and works better than Khachiyan's algorithm.

What is common among all the algorithms mentioned above is that they are all iterative and therefore sequential. That is, every subsequent solution depends on a preceding solution. On one side, this introduces uniqueness in each of these algorithms. On the other hand, however, the same feature makes these algorithms rather restrictive. This chapter aims at developing a method that is neither sequential not restricted to the surface of the polytope representing the set of feasible solutions of the linear programming problem. This development is motivated by the need for a speedy algorithm, since each of the classical techniques often requires a vast amount of calculations [15].

## 2.2 Linear Programming Problems in Canonical Form

Linear programming problems can be expressed in the canonical form. The canonical form of a linear programming problem is

$$\text{Maximize } C'X \quad (2.1)$$

$$\text{subject to } AX \leq b, \quad (2.2)$$

$$\text{and } X \geq 0. \quad (2.3)$$

Where  $C$  and  $X \in \mathfrak{R}^n$ ,  $b \in \mathfrak{R}^m$  and  $A \in \mathfrak{R}^{m \times n}$ . Here (2.1) is called the objective function, (2.2) is a system of linear inequation constraints and (2.3) is a non-negativity constraint.

### 2.2.1 Features of linear programming problems

In this section we state some standard definitions and some of the important characteristics of a solution to a linear programming problem formulated in the canonical form.

- **Feasible solution set.** A set of values of the decision variables  $x_1, x_2, \dots, x_n$  which satisfy all the constraints and also the non-negativity condition is called the *feasible solution set* of the linear programming problem.
- **Feasible solution.** Each element of the feasible solution set is called a *feasible solution*. A feasible solution is a solution which satisfies all the constraints and also the non-negativity conditions of the linear programming problem.

- **Optimal solution.** An optimal solution  $X^*$  is a feasible solution subject to

$$C'X^* = \max \{C'X : AX \leq b, X \geq 0\}. \quad (2.4)$$

- A (LP) is **feasible** if there exists at least one feasible solution. Otherwise, it is said to be **infeasible**.
- **Convex set.** A set  $S$  is said to be *convex set* if, given any two points in the set, the line joining those two points is completely contained in the set. Mathematically,  $S$  is a convex set if for any two vectors  $X^{(1)}$  and  $X^{(2)}$  in  $S$ , the vector  $X = \lambda X^{(1)} + (1 - \lambda)X^{(2)}$  is also in  $S$  for any real number  $\lambda \in [0, 1]$  [55].

Figure 1 and 2 represent convex set, whereas Figure 3 is not a convex set.

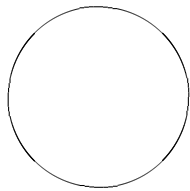


Figure 1



Figure 2

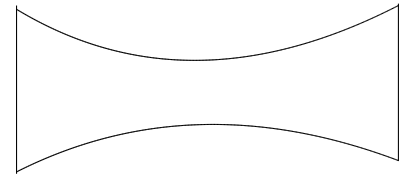


Figure 3

- **Extreme point.** An *extreme point* of the convex set is a feasible point that can not lie on a line segment joining any two distinct feasible points in the set. Actually, extreme points are the same as corner points [61].
- The set of all feasible solutions of *LPP* is a convex set.
- The optimal solution of *LPP* can always be associated with a feasible extreme point of feasible solution set.

- Linearity of the objective function results in convexity of the objective function.

Linear Programming Problems are solved by some methods like *the graphical method*, *the systematic trial and error method (enumeration method)* and *the simplex method*.

The graphical method is not applicable in more than two dimensional space. The enumerate method is a naive approach to solve a linear programming (which has an optimal solution) would be to generate all possible extreme points and determine which one of them gives the best objective function value. But, the simplex method, Which is a well-known and widely used method for solving linear programming problem, does this in a more efficient manner by examining only a fraction of the total number of extreme points of feasible solution set.

### **2.2.2 Simplex method**

The general steps of the simplex method are as follows.

1. Start with an initial extreme point of the feasible solution set.
2. Improve the initial solution if possible by finding another extreme point of feasible solution set with a better objective function value. At this step the simplex method implicitly eliminates from consideration all those extreme points of the feasible solution set whose objective function values are worse than the present one. This makes the procedure more efficient than the enumeration method.
3. Continue to find better extreme point of feasible solution set, improving the objective function value at every step.

4. When a particular extreme point of feasible solution set cannot be improved further, it becomes an optimal solution and the simplex method terminates.

### 2.2.3 Simplex method in tableau form

The various steps of the simplex method can be carried out in a more compact manner by using a tableau form to represent the constraints and the objective function. In addition by developing some simple formulas, the various calculations can be made mechanical [55]. In this section the simplex method is presented by illustrating the steps and using a tableau form for an example in canonical form.

We consider the following linear programming problem.

$$\begin{array}{llll} \text{maximize} & z & = & 6x_1 + 8x_2 \\ \text{subject to} & & & \\ & 5x_1 + 10x_2 & \leq & 60, \\ & 4x_1 + 4x_2 & \leq & 40, \\ & x_1, x_2 & \geq & 0. \end{array}$$

The standard form of the above  $LP$  problem is shown below.

$$\begin{aligned}
 &\text{maximize} && z &= & 6x_1 &+ & 8x_2 &+ & 0s_1 &+ & 0s_2 \\
 &\text{subject to} && && && && && && \\
 &&& 5x_1 &+ & 10x_2 &+ & s_1 && && && \leq & 60, \\
 &&& 4x_1 &+ & 4x_2 && &+ & s_2 && && \leq & 40, \\
 &&& x_1 & , & x_2 & , & s_1 & , & s_2 && && \geq & 0,
 \end{aligned}$$

where  $s_1$  and  $s_2$  are slack variables, which are introduced to balance the constraints.

- **Basic variable.** A variable is said to be *basic variable* if it has unit coefficient in one of the constraints and zero coefficient in the remaining constraints [53].

If all the constraints are “ $\leq$ ”, then the standard form is to be treated as the canonical form. The canonical form is generally used to prepare the initial simplex tableau [53]. The initial simplex table of the above problem is shown in Table 2.1.

**Table 2.1:** Initial Simplex tableau

	$c_j$	6	8	0	0		
$CB_i$	Basic Variable	$x_1$	$x_2$	$s_1$	$s_2$	Solution	Ratio
0	$s_1$	5	10	1	0	60	$60/10 = 6^{**}$
0	$s_2$	4	4	0	1	40	$40/4=10$
	$z_j$	0	0	0	0	0	
	$c_j - z_j$	6	8*	0	0		

\* key column

\*\* key row

Here,  $c_j$  is the coefficient of the  $j^{th}$  term of the objective function and  $CB_i$  is the coefficient of the  $i^{th}$  basic variable. The value at the intersection of the key column and the key row is called key element. The value of  $z_j$  is computed using the following formula.

$$z_j = \sum_{i=1}^2 (CB_i)(a_{ij})$$

Where  $a_{ij}$  is the coefficient for the  $i^{\text{th}}$  row and  $j^{\text{th}}$  column of the table.  $c_j - z_j$  is the relative contribution (profit). In this term,  $c_j$  is the objective function coefficient for the  $j^{\text{th}}$  variable. The value of  $z_j$  against the solution column is the value of the objective function and in this iteration, it is zero.

- **Optimality condition:** For maximization problem, if all  $c_j - z_j$  are less than or equal to zero, then optimality is reached; otherwise select the variable with the maximum  $c_j - z_j$  value as the entering variable (For minimization problem, if all  $c_j - z_j$  are greater than or equal to zero, the optimality is reached; otherwise select the value with the most negative value as the entering variable).

In Table 2.1, all the values for  $c_j - z_j$  are either equal to or greater than zero. Hence, the solution can be improved further.  $c_j - z_j$  is the maximum for the variable  $x_2$ . So,  $x_2$  enter the basis. this is known as *entering variable*, and the corresponding column is called *key column*.

- **Feasibility condition:** To maintain the feasibility of the solution in each iteration, the following steps need to be followed:
  1. In each row, find the ratio between the solution column value and the value in the key column.
  2. Then, select the variable from the present set of basic variables with respect to the minimum ratio (break tie randomly). Such variable is the *leaving variable* and the corresponding row is called the *key row*. The value at the intersection of the key row and the key column is called *key element* or *pivot element*.



In Table 2.1, the *leaving variable* is  $s_1$  and the row 1 is the key row. Key element is 10. The next iteration is shown in Table 2.2. In this table, the basic variable  $s_1$  of the previous table is replaced by  $x_2$ . The formula to compute the new values of table 2.2 is as shown below:

**Table 2.2:** Iteration 1

$CB_i$	$c_j$ Basic Variable	6 $x_1$	8 $x_2$	0 $s_1$	0 $s_2$	Solution	Ratio
8	$x_2$	1/2	1	1/10	0	6	6/(1/2)=12
0	$s_2$	<span style="border: 1px solid black; padding: 2px;">2</span>	0	-2/5	1	16	16/2=8**
	$z_j$	4	8	4/5	0	48	
	$c_j - z_j$	2*	0	-4/5	0		

Here

$$\text{New value} = \text{Old value} - \frac{\text{Key column value} \times \text{Key row value}}{\text{Key value}}$$

As a sample calculation, the computation of the new value of row and column  $x_1$  is shown below.

$$\text{New value} = 4 - \frac{4 \times 5}{10} = 4 - \frac{20}{10} = 4 - 2 = 2$$

Computation of the cell values of different tables using this formula is a boring process. So, a different procedure can be used as explained below.

1. *Key row:*

$$\text{New Key row value} = \text{Current Key row value} \div \text{Key element value}$$

2. *All other rows:*

$$\text{New row} = \text{Current row} - (\text{Its Key column value}) \times (\text{New Key row})$$

These computations are applied to the Table 2.1. in the following manner:

$$1. \text{ New Key row } (x_2 \text{ row}) = \frac{\text{Current Key row } (s_1 \text{ row})}{\text{Key element}(10)}$$

$$2. \text{ New } s_2 \text{ row} = \text{Current } s_2 \text{ row} - (4) \times \text{New key row } (x_2 \text{ row})$$

The results are shown in Table 2.2. The solution in the Table 2.2 is not optimal. The criterion row value for the variable  $x_1$  is the maximum positive value. Hence, the variable  $x_1$  is selected as entering variable and after computing the ratio,  $s_2$  is selected as leaving variable. The next iteration is shown in Table 2.3.

**Table 2.3:** Iteration 2

	$c_j$	6	8	0	0		
$CB_i$	Basic Variable	$x_1$	$x_2$	$s_1$	$s_2$	Solution	Ratio
8	$x_2$	0	1	1/5	-1/4	2	
6	$x_1$	1	0	-1/5	1/2	8	
	$z_j$	6	8	2/5	1	64	
	$c_j - z_j$	0	0	-2/5	-1		

In Table 2.3, all the values for  $c_j - z_j$  are either 0 (corresponding to basic variables) or negative (corresponding to nonbasic variables). Hence, the optimality is reached. The corresponding optimal solution is as follows:

$$x_1 = 8 \text{ and } x_2 = 2$$

and the optimal objective function value,  $z$  is 64.

### 2.2.4 Revised simplex method

The simplex method discussed in section 2.2 performs calculation on the entire tableau during each iteration. However, updating all the elements in tableau during a basis change is not really necessary for using the simplex method. The only information needed in moving from one tableau (basic feasible solution) to another tableau is as follows:

1. The relative profit coefficients ( $c_j - z_j$  row).
2. The column corresponding to the entering variable (Key column).
3. The current basic variables and their values (Solution column).

The information contained in other columns of the tableau plays no role in the simplex process. Hence the solution of large linear programming problems on a digital computer will become very inefficient and costly if the simplex method were to be used in its full tableau form. However, *revised simplex* method is an improvement over simplex method. The *revised simplex* method is computationally more efficient and accurate. Its benefit is clearly comprehended in case of large *LP* problems. The revised simplex method uses exactly the same steps as those in simplex method but at each iteration the entire tableau is never calculated. The relevant information it needs to move from one basic feasible solution to another is directly generated from the original equations. The steps of the revised simplex method (using matrix manipulations) can be summarized as follows:

**Step 1.** Obtain the initial feasible basis  $B$ . Determine the corresponding feasible solution  $x_B = B^{-1}b$ .

**Step 2.** Obtain the corresponding simplex multipliers  $\pi = c_B B^{-1}$ . Check the optimality of the current basic feasible solution (BFS). If the current basis is optimal, then STOP.

**Step 3.** If the current BFS is not optimal, identify the entering variable  $x_j$  (that is,  $c_j - z_j = c_j - \sum_{i=1}^m \pi_j a_{ij} = c_j - \pi P_j < 0$ ).

**Step 4.** Obtain the column  $\bar{P}_j = B^{-1}P_j$  and perform the minimum ratio test to determine the leaving variable  $x_l$ .

**Step 5.** Update the basis  $B$  (or  $B^{-1}$ ) and go to Step 2.

For more information and details, refer to [55].

## 2.3 Duality Theory and Its Applications

From both the theoretical and practical point of view, the theory of duality is one of the most important and interesting concepts in linear programming. The basic idea behind the duality theory is that every linear programming problem has an associated linear programming called its *dual* such that a solution to the original linear program also gives a solution to its dual. Thus, whenever a linear program is solved by the simplex method, we are actually getting solutions for two linear programming problems. The Dual can be considered as the “inverse” of the Primal in every respect. The column coefficients in the Primal constraints become the row co-efficients in the Dual constraints. The coefficients in the Primal objective function become the right hand side constraints in the Dual constraints. The column of constants on the right hand side of the Primal constraints becomes the row of coefficients of the dual objective function. The direction of the inequalities are reversed. If the primal objective function is a “Maximization” function then the dual objective function is a “Minimization” function and vice-versa. The concept of duality is very much in useful to obtain additional information about the variation in the optimal solution when certain changes are effected in the constraint co-efficient, resource availabilities and objective function co-efficients. This is termed as “post optimality” or “sensitivity analysis”.

The concept of a dual is introduced by the following programme.

**Example 2.1**

$$\begin{array}{l}
 \text{Maximize } Z = x_1 + 2x_2 - 3x_3 + 4x_4 \\
 \text{subject to} \\
 x_1 + 2x_2 + 2x_3 - 3x_4 \leq 25, \\
 2x_1 + x_2 - 3x_3 + 2x_4 \leq 15, \\
 x_1, x_2, x_3, x_4 \geq 0.
 \end{array}$$

The above linear programme has two constraints and four variables. The dual of this problem is written as

$$\begin{array}{l}
 \text{Minimize } W = 25y_1 + 15y_2 \\
 \text{subject to} \\
 y_1 + 2y_2 \geq 1, \\
 2y_1 + y_2 \geq 2, \\
 2y_1 - 3y_2 \geq -3, \\
 -3y_1 + 2y_2 \geq 4, \\
 y_1, y_2 \geq 0.
 \end{array}$$

$y_1$  and  $y_2$  are called the dual variables. The original problem is called the *primal* problem. Comparing the primal and the dual problems, we observe the following relationships:

1. The objective function coefficients of the primal objective function have become the right-hand-side constants of dual. Similarly, the right-hand-side constants of primal problem have become the objective function coefficients of the dual.
2. The inequalities have been reversed in the constraints.
3. The objective is changed from maximization in primal to minimization in dual.

4. Each column in the primal corresponds to a constraint (row) in the dual. Thus, the number of dual constraints is equal to the number of primal variables.
5. Each constraint(row) in the primal corresponds to a column in the dual. Hence, there is one dual variable for every primal constraint.
6. The dual of the dual is the primal problem.

In both of the primal and the dual problems, the variables are nonnegative and the constraints are inequalities. Such problems are called *symmetric dual linear programmes*.

**Definition.** A linear programme is said to be in *symmetric form*, if all the variables are restricted to be nonnegative, and all the constraints are inequalities (in a maximization problem the inequalities must be in “less than or equal to” form and in a minimization problem they must be in “greater than or equal to”). In matrix notation the symmetric dual linear programmes are:

Primal:

$$\begin{array}{ll}
 \textit{Maximize} & Z = C'X \\
 \textit{subject to} & \\
 & AX \leq b, \\
 & X \geq 0.
 \end{array}$$

Dual:

$$\begin{array}{ll}
 \textit{Minimize} & W = b'Y \\
 \textit{subject to} & \\
 & A'Y \geq C, \\
 & Y \geq 0.
 \end{array}$$

Where  $A$  is an  $m \times n$  matrix,  $b$  is an  $m \times 1$  column vector,  $C$  is a  $n \times 1$  column vector,  $X$  is an  $n \times 1$  column vector and  $Y$  is an  $m \times 1$  column vector.

The following table describes more general relations between the primal and dual that can be easily derived from the symmetric form definition. They relate the sense of constraint  $i$  in the primal with the sign restriction for  $y_i$  in the dual, and sign restriction of  $x_j$  in the primal with the sense of constraint  $j$  in the dual. Note that when these alternative definitions are allowed there are many ways to write the primal and dual problems; however, they are all equivalent.

Modifications in the primal-dual formulations

Primal model	Dual model
<i>Maximization</i>	<i>Minimization</i>
Constraint $i$ is $\leq$	$y_i \geq 0$
Constraint $i$ is $=$	$y_i$ is unrestricted
Constraint $i$ is $\geq$	$y_i \leq 0$
$x_j \geq 0$	Constraint $j$ is $\geq$
$x_j$ is unrestricted	Constraint $j$ is $=$
$x_j \leq 0$	Constraint $j$ is $\leq$

### 2.3.1 Economic interpretation of the dual problem

**Primal:**

Consider a manufacturer who makes  $n$  products out of  $m$  resources. To make one unit of product  $j$  it takes  $a_{ij}$  units of resource  $i$ . The manufacturer has obtained  $b_i$  units of resources  $i$  in hand, and the unit price of product  $j$  is  $c_j$ . Therefore, the primal problem leads the manufacturer to find an optimal production plan that maximizes the sales with available resources.



**Dual:**

Lets assume the manufacturer gets the resources from a supplier. The manufacturer wants to negotiate the unit purchasing price  $y_i$  for resource  $i$  with the supplier. Therefore, the manufacturers objective is to minimize the total purchasing price  $b'Y$  in obtaining the resources  $b_i$ . Since the market price  $c_j$  and the “product-resource” conversion ratio  $a_{ij}$  are open information on market, the manufacturer knows that, at least, a “smart” supplier would like to charge him as much as possible, so that

$$a_{1j}y_1 + a_{2j}y_2 + \cdots + a_{mj}y_m \geq c_j.$$

In this way, the dual linear program leads the manufacturer to come up with a least-cost plan in which the purchasing prices are acceptable to the “smart” supplier.

An example is giving to get more understanding on above economic interpretation. Let  $n = 3$  (products are Desk, Table and Chair) and  $m = 3$  (resources are Lumber, Finishing hours(needed time) and Carpentry hours). The amount of each resource which is needed to make one unit of certain type of furniture is as follows:

Resource \ Product	Desk	Table	Chair
Lumber	4 lft	6 lft	1 lft
Finishing hours	4 hrs	2 hrs	1.5 hrs
Carpentry hours	2 hrs	1.5 hrs	0.5 hrs

48 lft of lumber, 20 hours of finishing hours and 4 carpentry hours are available. A desk sells for \$60, a table for \$30 and a chair for \$20. How many of each type furniture should manufacturer produce?

Let  $x_1$ ,  $x_2$ , and  $x_3$  respectively denote the number of desks, tables and chairs

produced. This problem is solved with the following LP:

$$\begin{array}{rcll}
 \text{Maximize} & Z & = & 60x_1 + 30x_2 + 20x_3 \\
 \text{subject to} & & & \\
 & 8x_1 + 6x_2 + x_3 & \leq & 48, \\
 & 4x_1 + 2x_2 + 1.5x_3 & \leq & 20, \\
 & 2x_1 + 1.5x_2 + 0.5x_3 & \leq & 8, \\
 & x_1, x_2, x_3 & \geq & 0.
 \end{array}$$

Suppose that an investor wants to buy the resources. What are the fair prices,  $y_1, y_2, y_3$ , that the investor should pay for a ft of lumber, one hour of finishing and one hour of carpentry?

The investor wants to minimize buying cost. Then, his objective can be written as

$$\min W = 48y_1 + 20y_2 + 8y_3$$

In exchange for the resources that could make one desk, the investor is offering  $(8y_1 + 4y_2 + 3y_3)$  dollars. This amount should be larger than what manufacturer could make out of manufacturing one desk (\$60). Therefore,

$$8y_1 + 4y_2 + 2y_3 \geq 60$$

Similarly, by considering the “fair” amounts that the investor should pay for the combination of resources that are required to make one table and one chair, we conclude

$$6y_1 + 2y_2 + 1.5y_3 \geq 30$$

$$y_1 + 1.5y_2 + 0.5y_3 \geq 20$$

Consequently, the investor should pay the prices  $y_1, y_2, y_3$ , solution to the following LP:

$$\begin{array}{rcl}
 \text{Minimize } W & = & 48y_1 + 20y_2 + 8y_3 \\
 \text{subject to} & & \\
 & & 8y_1 + 4y_2 + 2y_3 \geq 60, \\
 & & 6y_1 + 2y_2 + 1.5y_3 \geq 30, \\
 & & y_1 + 1.5y_2 + 0.5y_3 \geq 20, \\
 & & y_1, y_2, y_3 \geq 0.
 \end{array}$$

The above LP is the dual to the manufacturers LP. The dual variables are often referred to as *shadow prices*, the fair market prices, or the dual prices.

Now, we shall turn our attention to some of the duality theorems that gives important relationships between the primal and the dual solution.

**Theorem 2.1: Weak Duality Theorem**

Consider the symmetric primal-dual linear programmes,  $Max Z = C'X, s. t. AX \leq b, X \geq 0$ , and  $Min W = b'Y, s. t. A'X \geq C, Y \geq 0$ . The value of the objective function of the minimum problem (dual) for any feasible solution is always greater than or equal to that of the maximum problem (primal).

From the weak duality theorem we can infer the following important results:

**Corollary 2.1** The value of the objective function of the maximum (primal) problem for any (primal) feasible solution is a lower bound to the minimum value of the dual objective.

**Corollary 2.2** Similarly the objective function value of the minimum problem (dual) for any (dual) feasible solution is an upper bound to the maximum value of the primal objective.

**Corollary 2.3** If the primal problem is feasible and its objective is unbounded (i.e.,  $\max Z \rightarrow +\infty$ ), then the dual problem is infeasible.

**Corollary 2.4** Similarly, if the dual problem is feasible, and is unbounded (i.e.,  $\min W \rightarrow -\infty$ ), then the primal problem is infeasible.

**Theorem 2.2: Optimality Criterion Theorem**

If there exist feasible solutions  $X^0$  and  $Y^0$  for the symmetric dual linear programmes such that the corresponding values of their objective functions are equal, then these feasible solution are in fact optimal solutions to their respective problems.

**Theorem 2.3: Main Duality Theorem**

If both the primal and dual problems are feasible, then they both have optimal solutions such their optimal values of the objective functions are equal.

We will use the properties of duality theory in proposed search algorithm, see section 5.

## 2.4 Computational Problems in Simplex Method

There are a number of computational problems that may arise during the actual application of the simplex method for solving a linear programming problem. In this section we discuss some complication that may occur.

### 1. Ties in the selection of the nonbasic variable:

In a maximization problem, the nonbasic variable with the largest positive value in the  $c_j - z_j$  row is chosen. In case there exists more than one variable with the same largest positive value in the  $c_j - z_j$  row, then we have a tie for selecting the nonbasic variable.

### 2. Ties in the minimum ratio rule and degeneracy:

While applying the minimum ratio rule it is possible for two or more constraints to give the same least ratio value. This result in a tie for selecting which basic variable should leave the basis. This complication causes degeneracy in basic feasible solution.

***Degenerate basic feasible solution:*** A basic feasible solution is said to be degenerate basic feasible solution if at least one basic variable in the solution equals zero.

### 3. Cycling:

At a degenerate basic solution, there is a risk of getting trapped in a cycle. When there exist degeneracy in feasible solution we can't say that simplex method will be terminated in a finite number of iterations, because this phenomenon causes that we obtain a new basic feasible solution in the next iteration which has no effect on the objective function value. It is then the-

oretically possible to select a sequence of admissible bases that is repeatedly selected without over satisfying the optimality criteria and, hence, never reaches a optimal solution. A number of anticycling procedure have been developed [17, 14, 66, 49, 10]. These anticycling method are not used in practice partly because they would slow the computer program down and partly because of the difficulty of recognizing a true zero numerically [7].

### 2.4.1 Complexity of the simplex method

The simplex method is not a good computational algorithm. It may lead to an exponential number of iteration as the example which is provided by Klee and Minty (1972). They showed that the simplex method as formulated by Dantzig visit all  $2^n$  vertices before arriving at the optimal vertex, this shows that the worst case complexity of the simplex algorithm is exponential time. Also, the number of arithmetic operations in the algorithm grows exponentially with the number of variables. The first polynomial time (interior point) algorithms, Ellipsoid method, for linear programming were given by Khachian (1979), then Karmarkar (1984) suggested the second polynomial time algorithm. Unlike the ellipsoid method Karmarkar's method solves the very larg linear programming problems faster than does the simplex method [58]. Both the ellipsoid method and the karmarkar's method are mathematically iterative and they are not better than simplex algorithm certainly for small linear programming problems and possibly for reasonably large linear programming problems [58].

According this restrictions of simplex algorithm and interior point methods provided by Khachian (1979) and Karmarkar (1984) and their variants, we need a new algorithm. We are going to give a stochastic search algorithm to obtain a **near-optimal** solution. Stochastic search algorithm are useful for applications

where stable and acceptable (i. e. near-optimal) answers are desired quickly [27]. In general, Stochastic search algorithms do not require knowledge of a derivative (as in gradient search methods) and perform best with highly nonlinear or highly combinatorial problems [12].

## 2.5 Proposed Stochastic Search Algorithm

Consider the linear programming problem in the canonical form:

$$\begin{array}{rcllcl}
 \text{P: max} & z & = & c_1x_1 & + & c_2x_2 & + & \dots & + & c_nx_n \\
 \text{subject to} & & & & & & & & & \\
 & a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & \leq & b_1, \\
 & a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & \leq & b_2, \\
 & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\
 & a_{m1}x_1 & + & a_{m2}x_2 & + & \dots & + & a_{mn}x_n & \leq & b_m, \\
 & x_1 & , & x_2 & , & \dots & , & x_n & \geq & 0.
 \end{array}$$

Consider the decision variable  $x_j$  for some  $j = 1, 2, \dots, n$ . If all other decision variables are set to zero, then  $x_j$  must satisfy  $a_{ij}x_j \leq b_i, i = 1, 2, \dots, m$ . To satisfy all of this condition for all  $i$ , we must have

$$0 \leq x_j \leq \min_{\{i: a_{ij}>0\}} \left\{ \frac{b_i}{a_{ij}} \right\}, 1 \leq j \leq n.$$

The bounding box  $B = \left\{ \underline{x} \mid 0 \leq x_j \leq \min_{\{i: a_{ij}>0\}} \left\{ \frac{b_i}{a_{ij}} \right\}, 1 \leq j \leq n \right\}$ ,  $B \subset E^n$ , contains the simplex set of feasible solutions of  $P$ . All feasible solutions of  $P$  are elements of  $B$ , though the converse may not hold. In this case all dual decision variables have zero lower bound, in fact, the positive quadrant contains the simplex representing the set of feasible solution of dual problem. We call it  $DB$ .

Now, generate a random element each from  $B$  and  $DB$  using a specified distribution,  $\mathbf{p}_r$ , on  $B$  and  $DB$ . If this element is primal feasible, then compute the primal objective function at this solution. Ignore all infeasible elements. Compute the largest value of the primal objective functions for these primal feasible solutions. The same process is applied to the dual problem by computing the smallest value of dual objective functions for the generated feasible dual solutions. In this process, a feasible primal solution is declared inadmissible if the value of the primal objective function does not exceed the highest value obtained till then. A feasible



dual solution is declared inadmissible if the value of the dual objective function is not smaller than the smallest value obtained till then. We keep track of the number of generated elements, the number of feasible solutions among these and the number of admissible points among the latter, in both problems. The procedure can be terminated after generating a specified number of

1. random elements
2. feasible solutions, or
3. admissible solutions.

The best solution at termination of the procedure is declared as the near-optimal solution. A second possible termination rule is obtained by considering the gap between near optimal primal and near optimal dual solutions. The procedure is terminated when this gap (or interval) is sufficiently small.

### 2.5.1 Multiple starts

The simulation approach obtains a near-optimal solution for every starting value. Near-optimality of the final solution does not depend on the initial value and hence the rate of convergence can not be determined. Therefore, it may be convenient to begin with more than one initial value, generating an independent sequence of solutions for every initial value. This will give us several end-points and the best of these will be better than the solution obtained from any one of them.

The maximum of the primal objective function and the minimum of the dual objective function obtained through the above procedure are both **near-optimal**. This method is not iterative in the sense that consecutive solutions may not improve the objective function. Also, it is simple from the mathematical point of view and there are no complicated computations.

Algorithmically, the method is described as follows.

• **Initialization in primal problem (PP):**

Set  $i = 1$ ,  $z_{n-o} = 0$ ,  $\underline{x}_{n-o} = \underline{0}$ ,  $i_o = 100$  (number of feasible points).

1. **Generation in PP:**

**Step 1.1.** Generate  $\underline{x}_i \in B$  using uniform distribution on the bounding box  $B$ .

**Step 1.2.** Test feasibility of  $\underline{x}_i$ .

**Step 1.3.** If  $\underline{x}_i$  is feasible, go to step 1.4. Otherwise, go to step 1.1.

2. **Computation in PP:**

**Step 1.4.** Compute  $z_i = \underline{c}'\underline{x}_i$ ,  $i = 1, 2, \dots, i_o$ .

**Step 1.5.**  $z_{n-o} = \max_i(z_i)$ ,  $i = 1, 2, \dots, i_o$  and  $\underline{x}_{n-o} =$  associated  $\underline{x}_i$  with  $z_{n-o}$ .

3. **Termination in PP:**

**Step 1.6.** Keep the values of  $\underline{x}_{n-o}$ ,  $z_{n-o}$  and stop.

• **Initialization in dual problem (DP):**

Set  $j = 1$ ,  $g_{n-o}$  is infinity,  $\underline{y}_{n-o}$  is infinity,  $j_o = 100$  (number of feasible points).

1. **Generation in DP:**

**Step 2.1.** Generate  $\underline{y}_j \in DB$  using uniform distribution on the bounding box  $DB$ .

**Step 2.2.** Test feasibility of  $\underline{y}_j$ .

**Step 2.3.** If  $\underline{y}_j$  is feasible, go to step 2.4. Otherwise, go to step 2.1.

## 2. Computation in DP:

**Step 2.4.** Compute  $g_j = \underline{b}y_j$ ,  $j = 1, 2, \dots, j_o$ .

**Step 2.5.**  $g_{n-o} = \min_j(g_j)$ ,  $j = 1, 2, \dots, j_o$  and  $g_{n-o} =$  associated  $\underline{y}_j$  with  $g_{n-o}$ .

## 3. Termination in DP:

**Step 2.6.** Keep the values of  $\underline{y}_{n-o}$ ,  $g_{n-o}$  and stop.

- **Termination in restarting:** if  $(g_{n-o} - z_{n-o}) \leq \epsilon$ , small positive value, or the number of restarting get equal to its initial value, then output the values of  $z_{n-o}$ ,  $\underline{x}_{n-o}$ ,  $g_{n-o}$ ,  $\underline{y}_{n-o}$  and stop restart.

### 2.5.2 Choice of probability distribution ( $p_r$ )

For the probability distribution, we have two choices.

#### 1. Uniform distribution.

In this case, every element is generated from  $B$  and  $DB$  with equal probability of selection. A significant limitation of uniform distribution is the progressively increasing rejection rate (rejection can occur for one of the two reasons: infeasibility and inadmissibility). This happens because points are generated with equal probability and rejection depends on the position of the current solution.

#### 2. Non-Uniform distributions.

Due to the limitation of Uniform distribution, we propose to use a non-uniform distribution. As a pilot study, we use the normal distribution with mean  $\mu_j =$  upper bound of  $x_j$ ,  $j = 1, \dots, n$  in  $PP$  and  $\nu_i =$  lower bound of

$y_i, i = 1, \dots, m$  in  $DP$  and unit variances. The results of pilot study show that the normal distribution is more suitable than uniform distribution.

A major limitation of the normal distribution is that half of the generated points are expected to exceed the mean. This implies a rejection rate exceeding  $1/2$ . This can be overcome by modifying the mean as follows.

We take a fraction  $f \in (0, 1)$  and use the mean

$$f \cdot \mu_j, 1 \leq j \leq n \text{ or } f \cdot \nu_i, 1 \leq i \leq m.$$

We tried some fractions to identify a good value for the fraction.

Another feature of the normal distribution is its unbounded support. Since the set of feasible primal solutions is bounded, we considered a distribution on a bounded support. In particular, we tried the multivariate Beta distribution, with the marginal distribution of  $x_j$  on  $\left[0, \min_{\{i: a_{ij} > 0\}} \left\{ \frac{b_i}{a_{ij}} : 1 \leq j \leq n \right\} \right]$  with parameters  $\alpha, \beta > 0$  so that  $\alpha > \beta$  or  $\alpha = k \cdot \beta$  with  $k > 1$ .

The gamma distribution is reasonable for the dual problem because its feasible solution set is unbounded.

For a problem having integer optimal solution, a discrete probability distribution is found to be very efficient. This is more so if the coordinates of bounding points are small multiples of optimal  $\underline{x}_j$  or  $\underline{y}_i$ .

## 2.6 Results of Simulation Studies

We consider some examples and implement the proposed simulation algorithm under some non-uniform distributions, namely normal, beta, Weibull and gamma, with different parameters and two discrete distributions, namely geometric and Poisson.

The following symbols are used in the tables.

- $Z_{n-o}$  and  $g_{n-o}$  are the primal and dual near-optimal objective functions, respectively.
- gap is the length of interval between primal and dual near-optimal values (gap).
- CPU time/Sec. is the average of running time of stochastic algorithm in terms of second per restart.

### 1. Beale's Example

The following example is called Beale's example and falls into cycling when the simplex algorithm is used. The optimum value of objective function is **0.05**.

$$\max z = 0.75x_1 - 150x_2 + 0.02x_3 - 6x_4$$

*s.t.*

$$0.25x_1 - 60x_2 - 0.04x_3 + 9x_4 \leq 0,$$

$$0.5x_1 - 90x_2 - 0.02x_3 + 3x_4 \leq 0,$$

$$x_3 \leq 1,$$

$$x_1, x_2, x_3, x_4 \geq 0.$$

In this example based on algorithm steps the following steps are carried out in both problems, Primal and Dual:

- (a) The points are generated randomly from different distribution with different parameters.
- (b) Test the feasibility of generated points.
- (c) Evaluate the primal and dual objective function values,  $z_i$  and  $g_j$  for feasible points.
- (d) Compute  $z_{n-o} = \max_i(z_i)$  and  $g_{n-o} = \min_j(g_j)$
- (e) if  $(g_{n-o} - z_{n-o}) \leq \epsilon$ , small positive value, or the number of restarting get equal to its initial value, then the algorithm is stopped and outputs the values of  $z_{n-o}$ ,  $\underline{x}_{n-o}$ ,  $g_{n-o}$ ,  $\underline{y}_{n-o}$  and stop restart.

The summary of results are shown in the following table.

**Table 2.4:** Near-Optimal values of Primal and Dual objective functions in LPP by using stochastic search algorithm with different distributions in primal and dual problem.

Distribution	$z_{n-o}$	$g_{n-o}$	# infeasible points			gap	CPU time/ sec.
			Primal	Dual	Total		
<i>Uniform</i> (0, $\mu_j$ )	0.0486	0.0537	109	4722	4831	0.0051	2.9
<i>N</i> ( $\mu_j, \mu_j^2$ )	0.049	0.0502	1690	4271	5961	0.0012	0.3
<i>N</i> ( $1/2\mu_j, \mu_j^2$ )	0.0496	0.0505	1133	7835	8968	0.0009	0.4
<i>N</i> ( $0.75\mu_j, \mu_j^2$ )	0.0492	0.051	1414	5583	6997	0.0018	0.32
<i>Beta</i> (0.5, 0.75)	0.0493	0.0513	136	983	1119	0.002	0.85
<i>Beta</i> (0.75, 0.5)	0.05	0.0512	100	5270	5370	0.0012	3
<i>Beta</i> (0.05, 0.05)	0.05	0.05	101	503	604	0	0.84

- As we mentioned before, the performance of non-uniform statistical distributions is better than uniform distribution. This can be understood from running time of proposed algorithm and gap (distance between primal and dual near-optimal objective function values).
- When normal distribution with the mean at the half of upper (lower) bound for primal (dual) decision variable is considered the length of gap is smaller than other cases of mean, this means the primal and dual near-optimal value are nearer to each other than other cases of mean.
- The results which have obtained from Beta distribution with parameters  $0 < \alpha, \beta < 1$  are better than other cases of  $\alpha, \beta$ .

## 2. Kuhn's Example

This example faces to cycling by using simplex algorithm. The optimum value of objective function is **2**.

$$\max z = 2x_1 + 3x_2 - x_3 - 12x_4$$

*s.t.*

$$-2x_1 - 9x_2 + x_3 + 9x_4 \leq 0,$$

$$\frac{1}{3}x_1 + x_2 - \frac{1}{3}x_3 - 2x_4 \leq 0,$$

$$2x_1 + 3x_2 - x_3 - 12x_4 \leq 2,$$

$$x_1, x_2, x_3, x_4 \geq 0.$$

In this example based on algorithm steps the following steps are carried out in both problems, Primal and Dual:

- (a) The points are generated randomly from different distribution with different parameters.

- (b) Test the feasibility of generated points.
- (c) Evaluate the primal and dual objective function values,  $z_i$  and  $g_j$  for feasible points.
- (d) Compute  $z_{n-o} = \max_i(z_i)$  and  $g_{n-o} = \min_j(g_j)$
- (e) if  $(g_{n-o} - z_{n-o}) \leq \epsilon$ , small positive value, or the number of restarting get equal to its initial value, then the algorithm is stopped and outputs the values of  $z_{n-o}$ ,  $\underline{x}_{n-o}$ ,  $g_{n-o}$ ,  $\underline{y}_{n-o}$  and stop restart.

The summary of results are shown in the following table.

**Table 2.5:** Near-Optimal values of Primal and Dual objective functions in LPP by using stochastic search algorithm with different distributions in primal and dual problem.

Distribution	$z_{n-o}$	$g_{n-o}$	# infeasible points			gap	CPU time/ sec.
			Primal	Dual	Total		
<i>Uniform</i> (0, $\mu_j$ )	1.9993	2	290	1491	1781	0.000071	0.29
<i>N</i> ( $\mu_j$ , 1)	1.9988	2	413	1595	2008	0.0012	0.124
<i>N</i> ( $\mu_j$ , $\mu_j^2$ )	1.9989	2	709	1536	2245	0.0011	0.117
<i>N</i> ( $1/2\mu_j$ , $\mu_j^2$ )	1.9988	2	1620	1573	3193	0.0012	0.125
<i>Beta</i> (0.5, 0.75)	1.9984	2	354	1510	1861	0.0016	0.238
<i>Beta</i> (0.5, 0.5)	1.9954	2	269	1310	1519	0.0046	0.223
<i>Gamma</i> (1, 1)	1.9947	2	385	1794	2179	0.0053	0.161
<i>Gamma</i> (1, 2)	1.9995	2	426	1426	1852	0.00005	0.1605
<i>Geo</i> (0.5)	2	2	431	1676	2107	0	0.221

- For this example uniform and non-uniform distributions with considering a discrete distribution (Geometric( $p = 0.5$ )) for dual problem have good and almost same performance.
- The problem has an integer optimal solution then, a discrete distribution (Geometric( $p = 0.5$ )) without considering the upper bounds of decision



variables in primal problem was applied and the exact optimal solution for both problems obtained.

### 3. Klee and Minty's Example:

In this example, simplex method as formulated by Dantzig, visits all  $2^n$  vertices before arriving at the optimal vertex. The optimal objective function value is **625**.

$$\begin{aligned} \max \quad & z = 8x_1 + 4x_2 + 2x_3 + x_4 \\ \text{s.t.} \quad & \\ & 16x_1 + 8x_2 + 4x_3 + x_4 \leq 625, \\ & 8x_1 + 4x_2 + x_3 \leq 125, \\ & 4x_1 + x_2 \leq 25, \\ & x_1 \leq 5, \\ & x_1, x_2, x_3, x_4 \geq 0. \end{aligned}$$

In this example based on algorithm steps the following steps are carried out in both problems, Primal and Dual:

- (a) The points are generated randomly from different distribution with different parameters.
- (b) Test the feasibility of generated points.
- (c) Evaluate the primal and dual objective function values,  $z_i$  and  $g_j$  for feasible points.
- (d) Compute  $z_{n-o} = \max_i(z_i)$  and  $g_{n-o} = \min_j(g_j)$

- (e) if  $(g_{n-o} - z_{n-o}) \leq \epsilon$ , small positive value, or the number of restarting get equal to its initial value, then the algorithm is stopped and outputs the values of  $z_{n-o}$ ,  $\underline{x}_{n-o}$ ,  $g_{n-o}$ ,  $\underline{y}_{n-o}$  and stop restart.

The summary of results are shown in the following table.

**Table 2.6:** Near-Optimal values of Primal and Dual objective functions in LPP by using stochastic search algorithm with different distributions in primal and dual problem.

Distribution	$z_{n-o}$	$g_{n-o}$	# infeasible points			gap	time/ sec.
			Primal	Dual	Total		
<i>Uniform</i> (0, $\mu_j$ )	588.05	634.05	20254	0	20254	46	14.19
$N(\mu_j, \mu_j^2)$	na	na	na	na	na	na	na
$N(1/2\mu_j, \mu_j^2)$	570.3	625.1	64880	237	65117	54.8	1.35
<i>Beta</i> (0.5, 0.75)	614.03	627.69	1920	0	1920	13.66	1.64
<i>Beta</i> (0.5, 1)	618.08	626.22	1179	0	1179	8.14	1.04
<i>Beta</i> (0.25, 1)	623.31	625.01	171	0	171	1.7	0.47
<i>Beta</i> (0.25, 0.75)	624.35	625.03	322	0	322	0.68	0.55
<i>Beta</i> (0.05, 0.05)	625	625	1121	0	1121	0	1.33
<i>Geo</i> (0.5)	625	625	833	0	833	0	0.55

- Also in this example, the efficiency of most of non-uniform statistical distributions is better than uniform distribution. This can be understood from running time of proposed algorithm and gap (distance between primal and dual near-optimal objective function values).
- The normal distribution with the mean at upper (lower) bound for primal (dual) decision variable is not applicable. But normal distribution with mean at the half of the upper (lower) bound for primal (dual) decision variable has better performance than uniform distribution.
- The results which are obtained from Beta distribution with parameters  $0 < \alpha < 1$  and  $\beta = 1$  are better than other cases of  $\alpha, \beta$ .

- The coordinates of bounding points of this example are multiples of optimal solution  $\underline{x}_{opt}, \underline{y}_{opt}$  then a discrete distribution is very efficient and here the Geometric distribution with parameter  $p = 0.5$  reaches to the exact optimal solution which is integer.

#### 4. Further example

This example has integer optimal solution in both problems and the optimal objective function value is **2200**.

$$\max z = 45x_1 + 80x_2$$

*s.t.*

$$5x_1 + 20x_2 \leq 400,$$

$$10x_1 + 15x_2 \leq 450,$$

$$x_1, x_2 \geq 0.$$

In this example based on algorithm steps the following steps are carried out in both problems, Primal and Dual:

- The points are generated randomly from different distribution with different parameters.
- Test the feasibility of generated points.
- Evaluate the primal and dual objective function values,  $z_i$  and  $g_j$  for feasible points.
- Compute  $z_{n-o} = \max_i(z_i)$  and  $g_{n-o} = \min_j(g_j)$
- if  $(g_{n-o} - z_{n-o}) \leq \epsilon$ , small positive value, or the number of restarting get equal to its initial value, then the algorithm is stopped and outputs the values of  $z_{n-o}$ ,  $\underline{x}_{n-o}$ ,  $g_{n-o}$ ,  $\underline{y}_{n-o}$  and stop restart.

The summary of results are shown in the following table.

**Table 2.7:** Near-Optimal values of Primal and Dual objective functions in LPP by using stochastic search algorithm with different distributions in primal and dual problem.

Distribution	$z_{n-o}$	$g_{n-o}$	# infeasible points			gap	CPU time/ sec.
			Primal	Dual	Total		
$N(\mu_j, \mu_j^2)$	2189.8	2208.32	1335	350	1685	18.472	0.076
$N(1/2\mu_j, \mu_j^2)$	2191.6	2207.2	1082	346	1428	15.536	0.071
$Weibull(\mu_j, 3.5)$	2198.7	2206.9	1454	550	2004	8.25	0.136
$Poisson(\mu_j)$	2200	2200	50000	2070	52070	0	18.2
$Poisson(1/2\mu_j)$	2200	2200	17	1675	1692	0	0.191

The Uniform distribution has no efficiency for this example. Then, we applied Non-uniform distribution for both primal and dual problems. Normal and Weibull distributions for primal and Gamma distribution for dual problem. As it can be seen when a fraction of mean parameter of Normal distribution is considered the efficiency of algorithm is increased. It is clear that the Weibull distribution for scale parameters bigger than 3.4 leads to Normal distribution and we see the behavior of this distribution. Because of existence an integer solution for the problem, discrete distributions are best for this problem.

All this work has been published in the Pakistan Journal of Statistics, Vol. 27, No. 1, pp. 65-74.

## 2.7 Conclusions

1. Since stochastic search technique uses pseudo-random numbers, there is no learning in the simulation based algorithm. As a result, more work does not necessarily imply improvement in the solution. On the other hand, the result is not necessarily bad if we have not worked much. Hence, an early solution can be a near-optimal solution and may not significantly improve with more work.
2. Generating feasible solutions is the major work in this algorithm. The bounding box is either a hyper-cube or cuboid and then there is a high rejection rate, or the complete Euclidean space with the same consequence. Attempts are being made to improve the performance of random solution generator.
3. Stochastic search algorithm is not an iterative procedure.
4. Stochastic search algorithm is simple from mathematical point of view and there are no complicated computations.
5. Stopping rule can be based on number feasible points in generation, number restarts the algorithm, length of interval of optimal objective function value (the length of gap between primal near-optimal and dual near-optimal values).
6. Discrete distributions like Geometric and Poisson are very efficient in certain special cases.