1. INTRODUCTION

1.1 An Overview

Linear Programming abbreviated as LP is related to allocation of resources, game theory, input/output analysis and decision theory. As an offspring of World War II, it was first applied as a planning tool by the United States Air Force in a project known as SCOOP (Scientific Computation of Optimum Programs), a study that sought the most efficient means of allocating human and material resources of the entire country in the event of a wholesale war. The need for such an informed allocation undoubtedly dogged every nation that has ever ventured to go on war. However, the problem was so complex that headway in the application of these techniques was made only with the advent of the electronic computers.

This high speed computing equipment enabled George Dantzig and his team of the Rand Corporation to develop the mathematical techniques to solve linear programming problems. Since 1947, when simplex algorithm to solve linear programming problems was first published, this technique and its extensions have been used for, among other purposes:

- Oil well exploration and production
- Petroleum refining and distribution
- Gasoline blending
- Ship, rail, truck and aircraft scheduling
- Product mix and Production planning
The primary reasons for its wide applications are [16]

1. any resource allocation problems in general can be identified or approximated as linear programming models.

2. efficient computational procedures for solving linear programming problems have been developed and

3. the ease with which the effect of parameter variation (sensitivity analysis) on the optimal solution to a LP problem can be handled.

The simplex algorithm is essentially an algebraic iterative procedure which in a finite number of steps determines the optimal solution, if there exists one or at least indicates the non-availability of an optimal solution due to unboundedness or infeasibility.

The criterion employed to improve the current situation to the next, causes the value of a linear objective function to change until the optimum solution is
reached. However, the number of iterations required for most of the large scale real life problems to converge to optimal solution may be large and consequently, the amount of computer time needed to solve linear programming problems is appreciable and considerably substantial. As such, quite an amount of research effort has been directed to the development of a variety of computational methods to hasten the rate of convergence.

1.2 State of the Art

The algorithm developed by George Dantzig and others at the RAND Corporation is an efficient computational method for solving linear programming problems. By changing only one vector in the basis at a time, a better basic feasible solution is obtained from the current.

The criteria which determine the entry and exit of vectors are precisely the same as those used in the Gauss elimination procedure. By adopting a technique to utilise the current iteration's computational results in the next iteration, an improvement in the solution can be made with a reduced computational effort [22].

A survey of the various algorithms which are expected to reduce the computational effort are outlined in the following paragraphs.
1.2.1 Starting bases:

The basis with which a problem is initiated to search for solution naturally has a great influence on the number of iterations required to solve it. In practice, one attempts to guess a starting basis which may be feasible and in the neighbourhood of the optimal solution. It is true that the total number of iterations will be reduced if a prior knowledge about the expected routine can make use of such a guess even if the basis is incomplete or infeasible.

Initially, a basis is chosen. Subsequently each column of the tableau is examined and pivoted into the basis if it had a non-zero entry corresponding to any unfilled position [20]. Once a feasible starting basis is found, the procedure is repeated until it extremises the objective function. Steps used to select the pivot column are as follows:

i. the conventional procedure selects that variable which has the most promising cost/contribution coefficient [3,34].

ii. the positive normalized procedures (PN1 and PN2) divide the cost/contribution coefficient used in choosing the pivot column, by some combination of the coefficients $a_{ij}$. The first of the two procedures proposed by [15] Dickson and Frederick uses the formula
where $a_{ij}^+$ is the positive part of $a_{ij}$. The column for which $d_j$ is maximal/minimal for $c_j < 0$. $c_j > 0$ is chosen as the pivot column.

The positive normalised procedure PN2 \[34\] employs the following simpler formula

$$d_j = \frac{c_j}{(c_j^2 + \Sigma a_{ij}^+)}$$


to determine the pivot column.

iii. The greatest change procedure was contemplated long ago but has been little used \[34\]. The column which gives the greatest change in the value of the objective function is chosen. It is that $j$ which maximises/minimises the expression

$$c_j = \min_{i} \left[ \frac{b_i}{a_{ij}} ; a_{ij} > 0 \right]$$

depending on the objective.

1.2.2 Crashing Techniques:

As an improvement over the previous method, crashing techniques are employed to crash through to a better basis in a relatively short time and many methods may be adopted to choose the order of the basis exchanges in the crashing technique \[4,47\].
In the process of crashing, a selected set of vectors is made basis in one of the variety of ways without regard to feasibility and a means for the vectors to enter and leave the crash algorithm at will was included in order to crash through to a new basis rapidly.

1.2.3 Multiple Column Selection

An improved version of the previous method is known as the multiple column selection. The process of selection consists of a number of passes at the beginning of each of which some $L$ number of nonbasic columns is selected as a set of candidates for pivoting. This is also referred to as block or mass pivoting [20]. During this pass the end of the induction of a set of variables into the basis is called a pass which may consist of one to many iterations in the simplex sense, no other nonbasic variable is considered. Simplex iterations are employed using the selected columns until the objective function has been extremised on that subset.

During a pass, any of the various means of selecting a pivot column explained previously might be used in extremising the objective. A basic column which becomes nonbasic during a pass is not further considered.

1.2.4 Suboptimisation

It is also possible to include the basic variables which become nonbasic during the above process, in the
set of candidates for pivoting. This particular process is familiarly known as suboptimisation [9], since one is effectively solving a subprogram containing a very limited number of columns (non basic variables) by the straight simplex method. But it is doubtful whether complete suboptimisation is worthwhile since a lot of time may be wasted with variables popping in and out of the basis.

As such the crashing techniques, multiple column selection procedure and suboptimisation procedure, all select a subset from the nonbasic part of the matrix which can be exchanged for a set of basis vectors. The order in which the vectors of the subset enter the basis and the choice of pivot rows, however, are not determined by a single set of criteria. It is this lack of flexibility which has given rise to a wide variety of algorithms.

1.2.5 Multiplex Method

Considering all the methods discussed so far, the improvement in the solution will be obtained by moving from one vertex to another vertex of the constrained set. In other words, a vertex in the feasible region is found out first, and from this vertex, a march is made in a direction which is influenced by the direction of the preference vector,

In the multiplex algorithm on the other hand [36, 37], the selection of vertices for the various passes
will be on the boundary of the constraint set which need necessarily be along the course of vertices taken by the simplex method. However, in finding the initial vertex an element of calculated guess is necessary. An improper guess leads to increased iterations and time.

1.2.6 Kachiyan's Polynomial Algorithm [21, 27, 34]

In simplex method, the number of steps increases as exponential based on the size of the problem. According to Kachiyan's polynomial algorithm, the number of steps increases as a polynomial based on the size of the problem.

This algorithm is a procedure for deciding whether the system is consistent or not. In addition, if the system is consistent, it finds the co-ordinates of a point satisfying all of the inequalities or at least finds them within a small margin of error.

Additionally, this method gives at start a maximum number of steps (each step requiring a fixed number of mathematical operations) that will be required to solve the problem. This maximum number of steps increases as the number of variables in the problem increases. But, with Kachiyan's method, the maximum number of steps grows far more slowly than with any other known method [6].
1.2.7 Karmarkar's polynomial-time Algorithm [30]

The new mathematical method evolved by Dr. Narendra Karmarkar to find solutions with absolutely least computational effort to day-to-day problems facing the business, industry aviation and message switching in communication is said to have passed many tests carried out by fellow mathematicians of the Bell Laboratories. Some of them have admitted that the new 'linear programming algorithm' discovered by him does have far-reaching potential.

However, it is commented that more work may have to be done before the Karmarkar algorithm is firmly established as the new panacea for solving those problems of the industrial world that have defied solution by the simplex method, which holds the field till now. According to researchers in mathematical programming the simplex method is effective in solving about 80 to 90 percent of linear programming problems but it becomes ineffective for the remaining problems which are more complex. It is claimed that the Karmarkar algorithm can tackle the remainder of the problems as well.

Though the Karmarker method is said to be amazingly fast mathematicians have complained that they are unable to do much work on this procedure, because AT and T Bell Laboratories are keeping a tight lid on the details of the Indian Mathematician's findings.
1.3 **Perspective of the work and its relevance**

In a simplex iteration, most of the time is spent in determining the reduced cost/contribution coefficients (pricing) for the candidate columns and in determining the transformed vector $Y_j$ corresponding to selected column $a_j$.

For large problems with great many columns, pricing can be very time consuming. The above method is not an efficient one, because it is edge directed and examines adjacent basic feasible solutions only. It is felt that this method can move faster to an optimal solution if it examines nonadjacent solutions (i.e.) changing more than one variable at a time.

But many of the suggested variants to the simplex method have not produced any appreciable change in the total computational time. It is suggested by Hadley [22] that simplex algorithm is not a unique method to determine the entering vector. The column $a_k$ to enter the basis is chosen using

$$ z_k - c_k = \min_j [z_j - c_j], \quad z_j - c_j < 0 \quad (1.3.1) $$

It seems logical to select the vector to enter the basis which will give the greatest increase in the objective function value. According to this criterion, the entering vector is selected by means of

$$ x_k(z_k - c_k) = \max_j \left[ \frac{(P^{-1}P_0)_j}{(P^{-1}P_j)_j} \right] x (c_j - z_j), $$
11

\[ z_j - c_j < 0 \quad \text{and} \quad (B^{-1}p_j)_i > 0 \quad \text{for all } i \quad (1.3.2) \]

It is reported that when both criteria have been applied to actual problems, both lead to an optimal solution in about the same manner and number of steps. In other words, from the computational point of view, there seems to be little or no advantage in using the more complicated criterion (1.3.2) to determine the vector to enter the basis. No computation is needed to apply (1.3.1) and for this reason (1.3.1) is usually used in preference to (1.3.2). It is found in practice that the number of iterations of a standard linear program with \( m \) constraints and \( n \) variables varies between \( m \) and \( 3m \), the average being \( 2m \). An upper bound for the number of iterations is \( 2(m+n) \) and occasionally some problems have violated even this bound.

The ideal criterion to choose the entering vector would be one which would assure that once a vector had been inserted into the basis, it would never have to be removed again (i.e.) a minimum of basis changes would lead to the optimal basis, in fact, no more than \( m \) iterations would be needed. Unfortunately, no criterion has been developed which will guarantee that once a vector has been inserted in the basis, it will never be removed.

An algorithm which is computationally simple and at the same time brings more than one variable not only
at start but also in the subsequent computations, known as MULTIPLEX ALGORITHM [40] has been developed, tested and reported. The multiplex algorithm while preserving the property of linear independence when two or more variables are brought into the basis, deliberately ignored the preservation of the property of feasibility.

The motivation for this work is to

i. improve and extend the multiplex algorithm as modified version (neglecting the preservation of feasibility property) to solve several day-to-day large scale LP problems and to explore its full potential,

ii. to investigate whether preservation of the feasibility property could have hastened the convergence further or not,

iii. to extend and investigate the performance of dual multiplex algorithm,

iv. to extend modified multiplex algorithm to solve bounded variable problems and

v. to extend multiplex algorithm to solve quadratic programming problems.

1.4 Conclusions

In this chapter the use of linear programming problems and the various techniques generally employed to solve them are discussed. The motivation for this work is to improve and extend the applicability of multiplex algorithm to solve large scale and non-linear
programming problems reducible to approximate LP models, and to develop a dual multiplex algorithm to remove more than one infeasible variable from the basis at a time. The dual multiplex algorithm may be extended to the study of sensitivity of LP problem solutions.