4.1 Introduction

An inherent advantage in the conventional simplex procedure is that it always preserves the property of feasibility and consequently finds at the end of each iteration only such solutions which are basic feasible. The multiplex algorithm [40] did not preserve this property in the course of computation but only removed infeasibility, if any in the basis when optimality condition for solution is satisfied. The multiplex algorithm assumed that as long as the convergence to solution is faster than the simplex this property need not be preserved. It was shown in the above work that acceptance of infeasible solutions enroute to optimal solution accelerated the convergence and the nonpreservation of the feasibility property did not affect in any way the end result. The heuristics of the multiplex algorithm is that the solution at the end of the first pass may be in the neighbourhood of the optimal solution compared to the one obtained at the end of the first iteration of the regular simplex procedure and infeasibility if any, may either disappear automatically within a pass or in subsequent passes or could be removed wholesale at the very end of the search for optimal solution. Infeasibility may also be removed at the end of each pass and the next pass is started with a feasible solution.
Since this is possible in the multiplex algorithm the objective function is first improved disregarding the feasibility criterion until optimality condition is satisfied. The solution is then checked for feasibility. Infeasibility, if any, in the solution is removed by invoking the dual multiplex algorithm.

A revision in the multiplex algorithm is made incorporating a check for feasibility and accept only feasible solutions with a view to improving the computational efficiency. With the addition of this provision the multiplex algorithm is expected to possess both the virtues of simplex procedure and may become an effective computational procedure.

4.2 Inclusion of feasibility criterion

4.2.1 Motivation

The multiplex algorithm has deliberately ignored the preservation of the property of feasibility. The reason for ignoring this property was based on the assumption that the preservation of this property may cause the multiplex algorithm to become sluggish and reduce the rate of convergence. Neglecting the preservation of this property, the multiplex algorithm was found to be three to four times faster than the simplex for medium size problems [40] and fifteen to twenty times faster than the simplex for large scale problems.

Now a question has arisen about the validity of this conjecture and therefore led to investigate whether the
preservation of the feasibility property could have hastened the convergence further.

4.3 **Graphical representation of the range of solution**

Consider a linear programming problem for which optimal solution is searched by the conventional simplex procedure. The search for optimal solution starts from point 'A' of the following figure 4.1 which shows the various states that may arise in the course of computation.

All the basic feasible and nonoptimal solutions are to the left of the point 'O' and the simplex procedure finds some of them in the course of search for the optimal solution such that $Z < Z_{\text{opt}}$. If the point 'O' represents the optimal solution, then there cannot be any feasible solutions to the right of the point 'O'. If there are any, then the point 'O' cannot be optimal. Therefore, the simplex procedure always restricts its search for the optimal solution only to the segment 'AO' of the solution range.

On the other hand, when optimal solution is searched by the multiplex algorithm, the solution may be anywhere on the segment 'AD' at the end of the first pass. If the solution is to the left of the point 'O', it may be feasible or infeasible. But if the solution is to the right of the point 'O', it is bound to be infeasible. It is quite possible that multiplex algorithm may attain optimal
Fig. 4.1 Range of Basic Solutions
Fig. 4.2 Different possibilities for convergence.
solution at the end of first pass and this has happened in a few cases.

However, if pass 1 converges to a point to the left of the point 'O', then the solution may be feasible or infeasible, but certainly non-optimal.

Multiplex algorithm may find at the end of a pass a basic solution to the right of the point 'O' also. When such a situation arises the solution is bound to be infeasible, but it may be optimal or non-optimal and the question of restoration of feasibility arises.

When infeasibility is removed by invoking the dual multiplex method on the current solution, the next solution may be optimal or may end up on the one of the basic feasible solutions to the left of the point 'O', in which case it is non-optimal. It may be observed from the above that the multiplex algorithm introduces infeasibility as well as overshoot and undershoot in the process of searching for optimal solution.

Both are deficiencies of the multiplex algorithm since the simplex procedure does not give rise to any such contingency. Even with these two contingencies, the saving in computational effort is quite appreciable.

This is illustrated in a sample example:

Refer to the linear programming problem coined by Beale to illustrate the vicious circle of cycling.
Max \( Z \) = \( 0.75x_1 - 20x_2 + 0.5x_3 - 6x_4 \)

subject to

\[
\begin{align*}
0.25x_1 - 8x_2 - x_3 + 9x_4 & \leq 0 \\
0.5x_1 - 12x_2 - 0.5x_3 + 3x_4 & \leq 0 \\
x_3 & \leq 1
\end{align*}
\]

\( x_1, x_2, x_3, x_4 \geq 0 \).

The simplex algorithm after seven basis changes returns to the starting basis if a proper choice of the variable is not made in between iterations.

The multiplex algorithm selects either the vector \([x_1, s_2, x_3]^T\) or \([s_1, x_1, x_3]^T\). The choice \([s_1, x_1, x_3]^T\) as the basis leads to optimal solution in a single pass. However, if \([x_1, s_2, x_3]^T\) is chosen as the basis, optimal solution is obtained at the end of fourth pass. The basic solutions obtained by using the multiplex algorithm in the spectrum between \(Z=0\) and \(Z \gg Z_{opt}\) are all indicated below.

<table>
<thead>
<tr>
<th>(Z=0)</th>
<th>(Z_{opt})</th>
<th>(Z \gg Z_{opt})</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>(Z = 1.25)</td>
<td>(Z = 1.33)</td>
<td>(Z = 2)</td>
</tr>
<tr>
<td>(s_1 = 0.75)</td>
<td>(s_1 = 2/3)</td>
<td>(x_1 = -8)</td>
</tr>
<tr>
<td>(x_1 = 1)</td>
<td>(x_2 = -1/24)</td>
<td>(x_2 = -3/8)</td>
</tr>
<tr>
<td>(x_3 = 1)</td>
<td>(x_3 = 1)</td>
<td>(x_3 = 1)</td>
</tr>
</tbody>
</table>
It is seen, while applying the multiplex algorithm that there was a tie among the leaving variables $s_1$, $s_2$ when $x_1$ enters the basis. If the starting basis is chosen as $[s_1, x_1, x_3]$, (i.e) the variable $s_2$ and $s_3$ are sent out, convergence to optimal solution is obtained in a single pass. On the other hand, if $s_1$ and $s_3$ are sent out, then the starting basis is $[x_1, s_2, x_3]^T$ which leads to an infeasible basic solution and converges to the optimal solution after three dual iterations. This example illustrates that all points to the right of 'O' are infeasible and the objective function value gradually reduces to optimum value.

4.4 Effect of feasibility on the rate of convergence

The revised multiplex algorithm was tested with a number of problems to check whether there is any improvement in computational efficiency. But quite contrary to the expectation, it turned out that preservation of feasibility property introduces a drag on the rate of convergence. Had not the simplex procedure preserved the property of feasibility, it would have accelerated to the optimal solution fast. Simplex procedure converges to optimal solution similar to an overdamped system whereas multiplex introduces at times damped oscillations similar to an underdamped system permitting overshoots and consequently quick response. From the table (4.1), it may be observed that in no case the revised multiplex has converged faster
than the multiplex and in one case the revised multiplex is worse than the simplex. Therefore, the inference is that as long as a linearly independent set of vectors can be identified, feasibility can be forsaken to accelerate convergence. This has been tested beyond doubt that with a number of examples and therefore feasibility property need not be preserved from iteration to iteration although the ultimate solution should be feasible and optimal.

4.5 Algorithm

Consecutive basic solutions in simplex procedure are related by the following:

The two solution vectors $[X_B]_c$ and $[X_B]_n$ are linked by the relationship

$$E[X_B]_n = [X_B]_c$$ \hspace{1cm} ... 4.5.1

where $E$ is a transformation matrix defined in Chapter 3. The transformation matrix is nonsingular as long as the pivot element is non-zero and therefore $E^{-1}$ exists. $[X_B]_n$ is computable in terms of $[X_B]_c$ as follows:

$$E^{-1}[X_B]_n = E^{-1}[X_B]_c$$ \hspace{1cm} ... 4.5.2

Therefore

$$[X_B]_n = E^{-1}[X_B]$$ \hspace{1cm} ... 4.5.3

Since $[X_B]_c$ is non-negative vector, feasibility of $[X_B]_n$ may be checked using equation (4.5.3).
If the entering variable combines feasibly with the remaining variables, then it is brought into the basis, otherwise it is rejected and the next promising variable is tried.

The following table (4.1) gives the CPU time clocked for various types of problems.

4.5.1 Step by Step procedure

The various steps involved in the revised multiplex algorithm are given below

Step 1: i. The promising vectors are identified using the relationship

\[(z_j - c_j) = [1, C_B B^{-1}] [p_j]^c\]

ii. under optimal conditions the processing is terminated.

iii. under non-optimal conditions promising vectors are identified.

Step 2: As in the regular multiplex algorithm a column \( a \) defined as \( a_{ij} = (B^{-1} p_j)_i \) is found where \( j \in J \).

\[ J = \text{[set of all subscripts of the promising variables]} \] and

\[ L = \text{is the no. of elements in the set.} \]

Step 3: The set of \( k \) number of entering vectors is selected from among the \( L \) number of promising vectors as per the multiple column selection procedure and 'B' matrix where \( 1 \leq k \leq L \).
Table 4.1 Comparison of Simplex, Multiplex, Modified Multiplex and Revised Multiplex Algorithm

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Variables</th>
<th>Constraints</th>
<th>Objective</th>
<th>Revised Simplex</th>
<th>Multiplex</th>
<th>Modified multiplex</th>
<th>Revised multiplex</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>372</td>
<td>43</td>
<td>Max.</td>
<td>30.83</td>
<td>15.99</td>
<td>13.03</td>
<td>17.0</td>
</tr>
<tr>
<td>2</td>
<td>412</td>
<td>100</td>
<td>Max.</td>
<td>129.00</td>
<td>7.37</td>
<td>6.33</td>
<td>19.62</td>
</tr>
<tr>
<td>3</td>
<td>462</td>
<td>150</td>
<td>Max.</td>
<td>302.00</td>
<td>19.39</td>
<td>17.01</td>
<td>21.33</td>
</tr>
<tr>
<td>4</td>
<td>700</td>
<td>100</td>
<td>Max.</td>
<td>205.00</td>
<td>17.19</td>
<td>16.53</td>
<td>50.40</td>
</tr>
<tr>
<td>5</td>
<td>928</td>
<td>74</td>
<td>Min.</td>
<td>136.97</td>
<td>56.00</td>
<td>51.02</td>
<td>177.00</td>
</tr>
<tr>
<td>6</td>
<td>372</td>
<td>43</td>
<td>Min.</td>
<td>30.67</td>
<td>16.80</td>
<td>16.15</td>
<td>17.03</td>
</tr>
</tbody>
</table>

TIME IN SEC.
Step 4: The $M^{-1}$ matrix is computed as described for the multiplex algorithm using the promising variables selected in Step 3, except that the feasibility of the solution is checked before updating the $M^{-1}$ matrix. Since $M^{-1}$ is obtained by successive multiplication of $E_0$ matrices, the solution vector $(E^{-1}P_0)$ can also be found using the $E_0$ matrix and $P_0$. Therefore, before updating the $M^{-1}$ matrix after finding the $E$ vector, the solution vector is computed using $E$ vector and the previous solution vector. If the resulting solution vector is also feasible then the $M^{-1}$ is updated, otherwise that variable is left out and the next variable in the selected list is considered.

Step 5: Step 4 is repeated till all the selected variables are exhausted. Then, return to step 1.

4.6 Conclusions

The preservation of the feasibility property is incorporated as an adjunct to the multiplex algorithm hoping to get additional saving in computation effects.

The observations recorded for various large scale linear programming problems give the indication that, though the revised multiplex algorithm possesses both the virtues of the simplex procedure, it is not an effective computational procedure. It is true that original multiplex algorithm introduces infeasibility as well as damped oscillations in the process of searching for optimal solu-
tion. However, if we try to check for feasibility and thereby forcibly arrest oscillations, the system tends to take more number of iterations and computational time. Preservation of the feasibility property in no way contributes for quick convergence. Preservation of feasibility property introduces a high degree of damping and makes the response slow and sluggish. In this sense it behaves as an overdamped system. Permitting infeasible solution to occur now and then makes the system under damped and quickens the response. Therefore preserving feasibility property is detrimental to the rate of convergence.