A Computer Program for Solving LP Problems by 2-Basic Variables Replacement at Each Simplex Iteration

M Babul Hasan and S M Asaduzzaman

Department of Mathematics, University of Dhaka, Dhaka-1000, Bangladesh.

Received on 20.07.2011. Accepted for Published on 22.02.2012

Abstract

In this paper, we develop a computer technique to implement the existing 2-basic variable replacement method of Paranjape for solving linear programming (LP) problems. To our knowledge there is no such computer oriented program which implemented Paranjape’s method. Our computer oriented program is a faster method for solving linear programs. A number of numerical examples are illustrated to demonstrate our algorithm.

I. Introduction

The development of Linear Programming (LP) was to seek the determination of the best (optimum) course of action of a decision problem under the restriction of limited resources.

We first consider a standard LP problem as follows.

\[(LP) \quad \sum_{j=1}^{n} a_j x_j = b \quad \text{s.t.} \quad x_j \geq 0\]

where 
\[
 a_j = (y_{1j}, y_{2j}, \ldots, y_{mj})^T, \quad j = 1, 2, \ldots, n, \\
 b \in \mathbb{R}^m, \quad m \leq n \quad \text{and} \quad X, C \in \mathbb{R}^n.
\]

Let us suppose the system \(B = \{a_{s_1}, a_{s_2}, \ldots, a_{r_1}, \ldots, a_{r_2}, \ldots, a_{s_m}\}\) is a basis, \(J_B\) the set of indices corresponding to vectors \(a_j\) of basis \(B\) i.e. \(J_B = \{s_1, s_2, \ldots, r_1, \ldots, r_2, \ldots, s_m\}\). If \(J = \{1, 2, \ldots, n\}\) then \(J_N = J \setminus J_B\) denote the set of indices of non basic vectors.

In the mid 20th century, numerous methods were developed for solving LP problems (Marcus[5], Kambo[6], Winston[9]). Among those methods, Dantzig’s [3] one basic variable replacement was widely accepted. Hadley [2] (P-105, problem 3-9) first suggested to replace two basic variables at each simplex iteration. Paranjape [8] finally presented a method in which he replaced two basic variables at each iteration. That method is faster than the usual simplex method of Dantzig [3].

II. Dantzig’s One Variable Replacement Method

In this section, we briefly discuss Dantzig’s [3] one variable replacement method. Let \(\tilde{x}_B\) be another feasible solution where \(\tilde{B} = \{a_{s_1}, a_{s_2}, \ldots, a_{u_1}, \ldots, a_{s_m}\}\), in which \(a_{u_1}\) is replaced by \(a_{u_1} \in J_N\). Since vectors \(a_{s_1}, a_{s_2}, \ldots, a_{r_1}, \ldots, a_{r_2}, \ldots, a_{s_m}\) of basis \(B\) are linearly independent, we can write \(a_{u_1}\) as a linear combination:

\[
a_{u_1} = \sum_{i=1}^{m} y_{i u_1} a_{s_i} = y_{r_1 u_1} a_{r_1} + \sum_{i \neq r_1} y_{i u_1} a_{s_i} \quad (2.1)
\]

\[
\Rightarrow a_{r_1} = \frac{1}{y_{r_1 u_1}}(a_{u_1} - \sum_{i \neq r_1} y_{i u_1} a_{s_i}) \quad (2.2)
\]

Now, the new value of right hand side constant ‘\(b\)’ becomes...
\[ b = \sum_{i=1}^{m} a_{s_i} x_{B_i} = \sum_{i \neq r_1}^{m} a_{s_i} x_{B_i} + a_{r_1} x_{B_{r_1}} \]

Substituting the values of \( a_{r_1} \) and performing necessary calculation (details can be found in Dantzig [3]), the value of \( 'b' \) can be represented by the following equation:

\[ b = \sum_{i \neq r_1}^{m} a_{s_i} \hat{x}_{B_i} + a_{u_1} \hat{x}_{B_{r_1}} \]  

(2.3)

Where \( \hat{x}_{B_{r_1}} = \frac{x_{B_{r_1}}}{y_{r_1 u_1}} \) and \( \hat{x}_{B_i} = x_{B_i} - y_{i u_1} \hat{x}_{B_{r_1}} \)  

(2.4)

The new optimal value of the objective function (details in Dantzig [3])

\[ \hat{Z} = Z + (c_{u_1} - Z_{u_1}) \hat{x}_{B_{r_1}} \]  

(2.5)

**Optimality Condition**

The objective function will improve if

\[ \hat{Z} > Z \Rightarrow (c_{u_1} - Z_{u_1}) \hat{x}_{B_{r_1}} > 0 \]

\[ \Rightarrow c_{u_1} - Z_{u_1} > 0 \]

i.e the general condition of improving objective function value is \( c_j - Z_j > 0 \).

**Entering and out going Variable**

The choice of entering variable into the basic is made by selecting \( u_1 \) th column of \( A \) for which \( c_{u_1} - Z_{u_1} \) is the greatest positive (for maximization problem) of \( c_j - Z_j, j = 1,2,\ldots n \). The selection of out going variable is based on choosing the basic variable corresponding to minimum value by minimum ratio test i.e choosing \( r_1 \) th column of \( B \) for which

\[ \frac{x_{B_{r_1}}}{y_{r_1 u_1}} = \min_i \left( \frac{x_{B_i}}{y_{i u_1}}, y_{i u_1} > 0 \right) \]

**Paranjape’s Two Basic Variables Replacement Method**

In this section, we briefly discuss Paranjape’s [8] two variables replacement method. Let \( \hat{x}_B \) be another feasible solution where

\[ \hat{B} = \{ a_{s_1}, a_{s_2} \ldots a_{u_1}, \ldots a_{u_2} \ldots a_{s_m} \} \].

in which \( a_{r_1} \) and \( a_{r_2} \) is replaced by \( a_{u_1}, a_{u_2} \in J_N \) respectively.

Since vectors \( a_{s_1}, a_{s_2} \ldots a_{r_1} \ldots a_{r_2} \ldots a_{s_m} \) of basis \( B \) are linearly independent, we can write \( a_{u_1} \) and \( a_{u_2} \) as linear combinations:

\[ a_{u_1} = \sum_{i=1}^{m} y_{i u_1} a_{s_i} = y_{r_1 u_1} a_{r_1} + y_{r_2 u_1} a_{r_2} + \sum_{i \neq r_1, r_2} y_{i u_1} a_{s_i} \]  

\( \Rightarrow y_{r_1 u_1} a_{r_1} + y_{r_2 u_1} a_{r_2} = a_{u_1} - \sum_{i \neq r_1, r_2} y_{i u_1} a_{s_i} \)  

(2.6)

Similarly, \( y_{r_1 u_2} a_{r_1} + y_{r_2 u_2} a_{r_2} = a_{u_2} - \sum_{i \neq r_1, r_2} y_{i u_2} a_{s_i} \)  

(2.7)

Solving these two equations for \( a_{r_1} \) and \( a_{r_2} \) (solution procedure and other calculations Hasan [4]), we have

\[ a_{r_1} = \frac{1}{K} (a_{u_1} y_{r_1 u_2} - a_{u_2} y_{r_1 u_1}) + \frac{1}{K} \sum_{j \neq r_1, r_2} y_{i u_1} (y_{r_1 y_{r_1}} - y_{i u_1} y_{r_2}) \]  

(2.8)

\[ a_{r_2} = \frac{1}{K} (a_{u_2} y_{r_1 u_2} - a_{u_1} y_{r_2 u_1}) + \frac{1}{K} \sum_{j \neq r_1, r_2} y_{i u_2} (y_{r_2 y_{r_2}} - y_{i u_2} y_{r_2}) \]  

where \( K = \begin{vmatrix} y_{r_1 u_1} & y_{r_1 u_2} \\ y_{r_2 u_1} & y_{r_2 u_2} \end{vmatrix} \)

The new value of \( 'b' \) becomes

\[ b = \sum_{i=1}^{m} a_{s_i} x_{B_i} = \sum_{i \neq r_1, r_2} a_{s_i} x_{B_i} + a_{r_1} x_{B_{r_1}} + a_{r_2} x_{B_{r_2}} \]

Omitting the detail calculation (see Hasan [4]), we can write the new value of \( 'b' \) with the help of \( a_{r_1} \) and \( a_{r_2} \) by the following equations:

\[ b = \sum_{i \neq r_1, r_2} a_{s_i} \hat{x}_{B_i} + a_{u_1} \hat{x}_{B_{r_1}} + a_{u_2} \hat{x}_{B_{r_2}} \]  

(2.9)
A Computer Program for Solving LP Problems by 2-Basic Variables Replacement at Each Simplex Iteration

where

\[
\hat{x}_{r1} = \frac{1}{K} \begin{bmatrix} x_{B1} & y_{r1}u_1 \\ x_{B2} & y_{r2}u_1 \end{bmatrix}
\]

and

\[
\hat{x}_{r2} = \frac{1}{K} \begin{bmatrix} y_{r1}u_1 & x_{B1} \\ y_{r2}u_1 & x_{B2} \end{bmatrix}
\]  \hspace{1cm} (2.10)

The optimum value becomes (nice explanation in Hasan [4])

\[
\hat{Z} = Z + (c_{u1} - Z_{u1}) \hat{x}_{r1} + (c_{u2} - Z_{u2}) \hat{x}_{r2}
\]

Optimality Condition

The value of the objective function will improve if \( \hat{Z} > Z \)

\[
\Rightarrow Z + (c_{u1} - Z_{u1}) \hat{x}_{r1} + (c_{u2} - Z_{u2}) \hat{x}_{r2} > Z
\]

which implies

(i) \( c_{u1} - Z_{u1} > 0 \) and

(ii) \( c_{u2} - Z_{u2} > 0 \).

The value of the objective function would not improve when \( \hat{x}_{r1} \) and \( \hat{x}_{r2} \) both are separately equal to zero. The general condition that the objective function value will improve if \( c_j - Z_j > 0 \), \( j = 1, 2 \ldots n \)

Entering Variable into the Basis

(i) Choose the \( u_1 \) th column of \( A \) whose index is in \( J_N \), but not in \( J_B \) for which

\( c_{u1} - Z_{u1} \) is the greatest positive value of

\( c_j - Z_j > 0, \ j = 1, 2 \ldots n \)

(ii) Choose the \( u_2 \) th column of \( A \) whose index is in \( J_N \), but not in \( J_B \) for which

\( c_{u2} - Z_{u2} \) is the greatest positive value of

\( c_j - Z_j > 0, \ j = 1, 2 \ldots n, \ j \neq u_1 \)

Out going Variable

Since the right hand side constant should always be non negative, thus the conditions on the choices of \( r_1 \) th and \( r_2 \) th columns of \( A \) whose index is in \( J_B \) are

(i) \( \hat{x}_{r1} \geq 0 \), (ii) \( \hat{x}_{r2} \geq 0 \), (iii) \( \hat{x}_{r1} \geq 0 \)

Satisfying the inequalities, we have the conditions for the selection of \( x_{B1} \) and \( x_{B2} \) respectively as

(i) choose \( x_{B1} \) for which

\[
\frac{x_{B1}}{y_{r1}u_1} = \min_i \left( \frac{x_B}{y_{i1}u_1}, y_{i1} > 0 \right)
\]

(ii) choose \( x_{B2} \) for which

\[
\frac{x_{B2}}{y_{r2}u_2} = \min_i \left( \frac{x_B}{y_{i2}u_2}, y_{i2} > 0 \right)
\]

Difference between the Two Methods

The main difference between Dantzig’s one basic and Paranjape’s two basic variable replacement method is in computing the formulae. The first one adopts the procedure of pivot operation whereas the second method establishes new formulae (other than pivot operation) for computing basic and non-basic variables. The computation formulae for basic and non basic variables have been presented through equations (2.10) and (2.11).

III. Our Computer Oriented Algorithm

In this section, we present our computer oriented algorithm for solving LP problems by replacing 2-basic variables at each simplex iteration.

Step 1: Define the types of the constraints and express the problem in its standard form.

Step 2: Start with an initial feasible solution in canonical form and set up initial table.

Step 3: Calculate the relative profit factors \( \pi_j \) as

\( \pi_j = c_j - Z_j = c_j - \langle c_B, \text{column corresponding to } x_j \rangle \) in the canonical system.

Step 4: If all \( \pi_j \leq 0 \) (maximization), the current basic feasible solution is optimal; stop here. If there is a single \( \pi_j > 0 \), one variable replacement; Go to Step 6. Otherwise go to Step 5.

Step 5:
Substep 1: Select the non basic variable with most positive and second most positive $c_j$ to enter the basis.

Substep 2: Choose two outgoing variables from the basis by minimum ratio test. If selected columns give more than one same minimum ratio, then choose distinct rows.

Substep 3: Perform two basic variable replacement operations to get simplex table.

Substep 4: Go to Step 4.

Step 6: Select the non basic variable to enter the basis.

Substep 1: Choose the outgoing variable from the basis by minimum ratio test.

Substep 2: Perform the pivot operation to get the table and basic feasible solution.

Substep 3: Go to Step 4.

Step 7: If any $c_j$ corresponding to non basic variable is zero, take this column as pivot column (for alternative solution) and go to Step 6.

Computer oriented program

In this section, we present our program in programming language Mathematica (Eugere[1], Wolfram[7]). This program is written in Mathematica 5.2 for Students version. In this program, we have used eight module functions, \texttt{maketable[t_] \_rnonoperation[t_], tmobasic[t_, onetsop[t_, alter[t_, tmobop[t_, main[tmobasic_], vinput[n, a_]. The function \texttt{vinput[n, a_]} has been used for taking inputs. This function will ask the user to input number of rows, number of columns, number of greater than type constraints, input row by row, right hand side constants, cost vector and type of each constraint e.g. ‘$l$’ for less than type, ‘$g$’ for greater than type and ‘$e$’ for equality type constraints respectively. Our program is case sensitive and minimizes the tedious work of input data by generating slack or artificial variables.

The function \texttt{maketable[t_]} is for making tables and the function \texttt{rnonoperation[t_]} performs all necessary calculations for single variable replacement. The module function \texttt{tmobasic[t_]} has been used for two basic variable replacements in a single iteration. If the case arises that a simplex table ends with only one positive $c_j$, then to incorporate the problem with single variable replacement we have introduced the function \texttt{onetsop[t_]} and this function controls all necessary operations for single variable replacement. The function \texttt{alter[t_]} identifies alternative (if any) solutions in either single basic or two basic variable replacements. The module function \texttt{tmobop[t_]} does the primary works for using the function \texttt{tmobasic[t_]}. Finally the function \texttt{main[tmobasic]} calls all the functions discussed above and controls the program. The combined program can be found by contacting the author.

Numerical Examples and Comparison

In this section, we compare the results obtained by our program with that of Dantzig’s [3] and Paranjape’s [8] methods. We also show the differences between these methods with illustrative numerical examples.

Example 1: This example is taken from Paranjape [8].

\[\begin{align*}
\text{Max} \quad Z &= -15x_1 + 25x_2 + 15x_3 - 30x_4 + 10x_5 - 40x_7 - 10x_9 \\
\text{subject to} \quad & -\frac{1}{2}x_1 + \frac{1}{2}x_2 + \frac{1}{2}x_3 \leq 0, -\frac{1}{4}x_1 + \frac{3}{4}x_2 - \frac{1}{4}x_3 \leq 0, -\frac{1}{4}x_4 + \frac{1}{4}x_5 + \frac{1}{4}x_6 \leq 0 \\
& \quad -\frac{1}{2}x_4 + \frac{1}{2}x_5 - \frac{1}{2}x_6 \leq 0, x_1 + x_4 + x_7 \leq 100, x_2 + x_5 + x_8 \leq 100, x_3 + x_6 + x_9 \leq 60 \\
& \quad x_i \geq 0
\end{align*}\]

Solution obtained from usual simplex method

Optimal table of one basic variable replacement method is presented in Table 5. Note that the table number refers to the number of iterations. For example, here Table 5 indicates that the optimal solution is obtained after 5 iterations. The optimal value is $Z = 191/6$ and the optimal solution is $x_1 = 0, x_2 = 7/2, x_3 = 16/3$ which is identical to that obtained form usual simplex method.

Example 1 takes five iterations in Dantzig’s method whereas it takes only three (including initial table) iterations in 2-basic variables replacement method. Using our program, we have to input 7, 9, 0 respectively to indicate the LP has 7 constraints with 9 variables and no ‘greater than type’ constraints. If there exists any ‘greater than type’ constraints then input the number of those constraints.
A Computer Program for Solving LP Problems by 2-Basic Variables Replacement at Each Simplex Iteration

Table 1: Caption: no. of iterations=5

| -- | C_1 | -15 | 25 | 15 | -30 | 10 | 0 | -40 | 0 | -10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PHS |
|----|-----|-----|----|----|-----|----|---|-----|---|----|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C_B Basis | X_1 | X_2 | X_3 | X_4 | X_5 | X_6 | X_7 | X_8 | X_9 | S_1 | S_2 | S_3 | S_4 | S_5 | S_6 | S_7 | -- |
| 25 | X_1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50 |
| -15 | X_2 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 100 |
| 10 | X_3 | 0 | 0 | 0 | -3 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 0 |
| 0 | S_4 | 0 | 0 | 0 | 1 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | -2 | 1 | 0 | 0 | 0 | 0 |
| 15 | X_4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50 |
| 0 | S_5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50 |
| 0 | S_7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50 |

Optimal table of our method

Optimal table of our method is presented in table 2.

Table 1: Caption: no. of iterations=3

| -- | C_1 | -15 | 25 | 15 | -30 | 10 | 0 | -40 | 0 | -10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PHS |
|----|-----|-----|----|----|-----|----|---|-----|---|----|---|---|---|---|---|---|---|---|---|---|---|---|---|
| C_B Basis | X_1 | X_2 | X_3 | X_4 | X_5 | X_6 | X_7 | X_8 | X_9 | S_1 | S_2 | S_3 | S_4 | S_5 | S_6 | S_7 | -- |
| 15 | X_1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50 |
| 25 | X_4 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50 |
| 10 | X_5 | 0 | 0 | 0 | -3 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | S_4 | 0 | 0 | 0 | 1 | 0 | -1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| -15 | X_7 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| 0 | S_5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 50 |
| 0 | S_7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 |

Optimal solution by one basic variable replacement method:

The optimal value is $Z = 191/6$ and the optimal solution is $x_1 = 0$, $x_2 = 7/2$, $x_3 = 16/3$ presented in table 3.

Optimal solution by our method:

The second table of Paranjape’s method:

Example 2:

Max $Z = 2x_1 + 3x_2 + 4x_3$

s.t. $x_1 + x_2 + x_3 \geq 5$, $x_1 + 2x_2 = 7$, $5x_1 - 2x_2 + 3x_3 \leq 9$, $x_1, x_2, x_3 \geq 0$

The optimal value is $Z = 191/6$ and the optimal solution is $x_1 = 0$, $x_2 = 7/2$, $x_3 = 16/3$ presented in table 3.
Table 3:

<table>
<thead>
<tr>
<th>Basis</th>
<th>C_B</th>
<th>X_1</th>
<th>X_2</th>
<th>X_3</th>
<th>S_1</th>
<th>A_1</th>
<th>A_2</th>
<th>S_2</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>-M</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>-M</td>
<td>-M</td>
<td>0</td>
<td>RHS</td>
</tr>
<tr>
<td></td>
<td>A_L</td>
<td>0</td>
<td>0</td>
<td>-\frac{1}{4}</td>
<td>-1</td>
<td>1</td>
<td>-\frac{1}{12}</td>
<td>-\frac{1}{12}</td>
<td>\frac{1}{6}</td>
</tr>
<tr>
<td>3</td>
<td>X_2</td>
<td>0</td>
<td>1</td>
<td>-\frac{1}{4}</td>
<td>0</td>
<td>0</td>
<td>\frac{5}{12}</td>
<td>-\frac{1}{12}</td>
<td>\frac{1}{6}</td>
</tr>
<tr>
<td>2</td>
<td>X_1</td>
<td>0</td>
<td>-\frac{1}{4}</td>
<td>0</td>
<td>0</td>
<td>\frac{1}{6}</td>
<td>\frac{1}{6}</td>
<td>\frac{1}{3}</td>
<td></td>
</tr>
</tbody>
</table>

Optimal table of our method:

Table 4:

<table>
<thead>
<tr>
<th>Basis</th>
<th>C_B</th>
<th>X_1</th>
<th>X_2</th>
<th>X_3</th>
<th>S_1</th>
<th>A_1</th>
<th>A_2</th>
<th>S_2</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>-M</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>-M</td>
<td>-M</td>
<td>0</td>
<td>RHS</td>
</tr>
<tr>
<td></td>
<td>A_L</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>\frac{1}{2}</td>
<td>\frac{1}{2}</td>
<td>\frac{15}{2}</td>
</tr>
<tr>
<td>3</td>
<td>X_2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>\frac{1}{2}</td>
<td>0</td>
<td>\frac{7}{2}</td>
</tr>
<tr>
<td>0</td>
<td>S_1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>\frac{5}{6}</td>
<td>\frac{1}{2}</td>
<td>\frac{19}{6}</td>
</tr>
<tr>
<td>\bar{C}_3</td>
<td>\bar{C}_3 - \bar{Z}_3</td>
<td>-\frac{15}{2}</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-M</td>
<td>-\frac{17}{6}</td>
<td>-M</td>
<td>-\frac{4}{2}</td>
</tr>
</tbody>
</table>

We have to input ‘l’ seven times to ensure all constraints are less than type with ‘A’, the coefficient matrix, right hand side constants ‘b’ and cost coefficient ‘C’. The program will generate the required number on slack variables. The solution obtained using our program is identical to that of the Dantzig’s single variable replacement method. We observe that our method reduces the number of iterations by 40%. The optimal tables of one basic variable replacement method and our method are also presented. Example 2 also supports this claim.

IV. Conclusion

In this paper, we developed a computer technique to implement the existing 2-basic variable replacement method of Paranjape [8] for solving linear programming (LP) problems. To our knowledge there is no such computer oriented program which implemented Paranjape’s method. Since that method is based on hand calculation, it is time consuming and laborious. On the other hand, our computer oriented program is convenient for solving linear programs. A number of numerical examples are illustrated to demonstrate our program.
