New Approach to Solving Generalised Linear Goal Programming Problem

Orumie, Ukamaka Cynthia
Lecturer
Department of Mathematics and Statistics,
University Of Port Harcourt, Nigeria

Abstract

A new approach to solving any linear goal programming problems is developed irrespective of whether it is in lexicographic, weighted or generalized form. Also the method by Orumie and Ebong 2013a and 2013b were modified to incorporate the rigid constraints in the initial table. Illustrative example were given, and the new algorithm proved better.

Keyword: Algorithm, goal programming, Lexicographic, Weighted,

INTRODUCTION

Since the development of Goal Programming, it’s study has been an important topic of interest for researchers in the area of multi-criteria decision making. In a view to solving especially Linear GPPs, the most commonly used goal programming solution methods were introduced by Lee (1972) and Ignizio (1976) (see Schniederjans and Kwak(1982)). According to Schniederjans and Kwak (1982), these algorithms consume a lot of time and space during computation involving a large number of constraints (Orumie and Ebong (2014)).
In 2011, Orumie and Ebong developed an algorithm and compared it with the most commonly used ones. In (2013a) and (2013b), Orumie and Ebong developed separate methods of solving lexicographic and weighted goal programming respectively which corrected most of the computational errors in other existing algorithms as highlighted in Orumie and Ebong (2014). This new algorithm corrected the error in the alternative method of solving Linear goal programming problem model by Orumie and Ebong (2011), with inclusion of the rigid constraints.

However, the new algorithms in testing for optimality did not consider the fact that there exist a situation where absolute value of coefficient of deviational variable under consideration that are not in the table can re-enter. Also in case of choosing the entering variable, the algorithm did not consider in case of ties in the minimum of the ratio of the right hand side to the positive coefficients of the columns corresponding to the ties. The initial table of both Orumie and Ebong 2011 and 2013 did not include rigid constraints.

Thus, a major drawback in the use of goal programming problems has been the lack of an algorithm, capable of reaching optimal solution in reasonable time (see Olson (1984)).

In order to reduce the computational time of GPP, and correct the above errors, the researcher decided to develop an algorithm for linear goal programming that can minimise computational time when compared with existing algorithms irrespective of its structure. The algorithm could be used to solve problems represented either in preemptive model, weighted model, or prioritized (generalized or combined models).

Section two shows several kinds of linear goal programming structure. The initial table is formulated and presented in section three. Section four, five, and six shows the modified lexicographic, modified weighted and the new generalized method respectively. Section seven is the illustrative example whereas section eight and nine is summary and conclusion.

**KINDS OF LINEAR GOAL PROGRAMMING REPRESENTATION**

Consider the Linear Goal programming model formulation for $n$ variables, $r$ goals (constraints), $s$ rigid constraints, and $K$ preemptive priority factors defined below:

Find $\bar{x} = (x_1, x_2, ..., x_n)$ that minimizes

$$z = \{ L_1(d^-, d^+), L_2(d^-, d^+), L_3(d^-, d^+), ..., L_K(d^-, d^+)\} \quad (1)$$

subject to
New Approach to Solving Generalised Linear Goal Programming Problem

\[ f_i(x) + d_i^- - d_i^+ = b_i \quad \text{for all } i = 1, \ldots, r \]  
(2)

\[ g_i(x) \leq h_i \quad \text{for all } i = 1, 2, \ldots, \eta \]  
(3)

and \( x \geq 0, \quad d^- \geq 0, \quad d^+ \geq 0 \)  
(4)

Here, \( x \) is the \( j \)-th decision variable;

\( z \) represents the objective function which measures the achievement of the goals at each priority level \( k \);

\( d^- = (d_1^-, d_2^-, \ldots, d_r^-) \), \( d^+ = (d_1^+, d_2^+, \ldots, d_r^+) \);

\( d_i^- \) is the \( i \)-th negative deviational variable;

\( d_i^+ \) is the \( i \)-th positive deviational variable;

\( L_k \) is the \( k \)-th linear function of the deviational variables associated with objectives at priority level \( k \), \( k = 1, 2, \ldots, K \);

\( K \) is the total number of priority factors in the model,

\( b_i \) is the right-hand side constant for goal constraint \( i \);

\( h_i \) is the right-hand side constant for rigid constraint \( i \);

\( f_i \) is the linear function on the left hand side of the \( i \)-th goal constraint;

\( g_i \) is the linear function on the left hand side of the \( i \)-th rigid constraint;

let \( m = r + \eta \)

Then, examples of linear goal programming structures will be enumerated to highlight the differences between the proposed and existing algorithms.

The first example (example 1) given below is one where the objective function is strictly lexicographic.

Example 1:

Min \( z = p_1 d_2^-, \quad p_2 d_1^-, \quad p_3 d_3^- \)

\[ 120x_1 + 90x_2 + d_1^--d_1^+ = 1950 \]

\[ x_1 + d_2^- - d_2^+ = 13 \]

\[ x_2 + d_3^- - d_3^+ = 5 \]

\[ 6x_1 + 3x_2 \leq 90 \]

\[ 3x_1 + 6x_2 \leq 72 \]
Table 1: Initial table of problem 1 of new algorithm

<table>
<thead>
<tr>
<th>Var in z with priority</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$d_1^-$</th>
<th>$d_2^-$</th>
<th>$d_3^-$</th>
<th>$s_1$</th>
<th>$s_2$</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_2 d_1^-$</td>
<td>120</td>
<td>90</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1950</td>
</tr>
<tr>
<td>$p_1 d_2^-$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
</tr>
<tr>
<td>$p_3 d_3^-$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>$s_1$</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>90</td>
</tr>
<tr>
<td>$s_2$</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>72</td>
</tr>
</tbody>
</table>

Column one consists of the deviational variables that appeared in the objective with priority assigned to each of them, together with slack variables from the rigid constraints. These form the basis. All other columns excepts the last column forms coefficients of the decision variables from both goal and rigid constraints, deviational variables that appeared in z, and slack variables from the rigid constraints as appeared in constraint equation.

The first column of row two is the deviational variable of the first goal constraint that appears in the objective function. This corresponds to priority level two ($p_2$), and the rest of the entries of row two are the coefficients of decision variables from both goal and rigid constraints, deviational variables that appeared in z, and slack variables from the rigid constraints as appeared in constraint equation. Similarly, on the 3rd row of column one refers to the deviational variable of the 2nd goal constraint that appeared in the objective function. This corresponds to priority $p_1$, and the rest of the entries of row three follows from that of row two. Row four of column one refers to the deviational variable from 3rd goal constraint that appeared in the objective function, which corresponds to the 3rd priority $p_3$. The last two entries on the first column of the last two rows refer to the respective slacks of the rigid constraints.
REMARKS
The new approach utilizes only the deviational variables that appeared in z on both basis and non basis, and does not include objective function rows. Instead, priority is attached to the deviational variables in column one and any other deviational variable column augmented if need be. This type of problem can be handled by Orumie and Ebong (2013) but its initial table omitted the slack and surplus rows from rigid constraints.

Another example (example 2) to be represented in a table is a case where the objective function is a weighted sum of the deviational variables (Weighted GP).

Example 2

\[
\begin{align*}
\min z &= (2d_1^- + 4d_2^+) \\
\text{s.t} & \\
4x_1 + 8x_2 + d_1^- - d_1^+ &= 45 \\
8x_1 + 24x_2 + d_2^- - d_2^+ &= 100 \\
x_1 + 2x_2 &\leq 10 \\
x_1 &\leq 9
\end{align*}
\]

Therefore, initial table of example 2 using new algorithm is represented below;

<table>
<thead>
<tr>
<th>Variable in basis as appeared in z with ( s_i )</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( d_1^- )</th>
<th>( d_2^+ )</th>
<th>( s_1 )</th>
<th>( s_2 )</th>
<th>Solution value ( b_i ) (rhs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 ( d_1^- )</td>
<td>4</td>
<td>8</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>45</td>
</tr>
<tr>
<td>4( d_2^+ )</td>
<td>8</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>( s_1 )</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>( s_2 )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>9</td>
<td></td>
</tr>
</tbody>
</table>

Column one consists of the deviational variables that appeared in z with weight assigned to each of them, and slack variables from the rigid constraints which forms the basis. The rest of the entries are obtained in manner similar to that used to obtain the entries in example 1.
The third example (example 3) given below is one where both weighted and preemptive approaches are combined to form a model to a problem. (i.e goals categorised into groups, where the goals within each group are of equal importance, but there are slight differences between the groups in their level of importance such that weight is used within each group)(Generalised variant).

Example 3

\[
\begin{align*}
\min z &= p_1(d_1^- + 3d_2^- + 2d_3^-), \ p_2(d_4^+), \ p_3(d_5^+) \\
\text{s.t} & \\
& x_1 + x_2 + d_1^- - d_1^+ = 30 \\
& x_3 + x_4 + d_2^+ - d_2^- = 30 \\
& 3x_1 + 2x_3 + d_3^- - d_3^+ = 120 \\
& 3x_2 + 2x_4 + d_4^- - d_4^+ = 20 \\
& 10x_1 + 9x_2 + 8x_3 + 7x_4 + d_5^- - d_5^+ = 800 \\
& 6x_1 + 3x_2 \leq 90, \\
& 3x_1 + 6x_2 \leq 72 \\
& x_1, x_2, x_3, x_4, d_1^-, d_1^+, d_2^+, d_2^-, d_3^-, d_3^+, d_4^-, d_4^+, d_5^-, d_5^+ \geq 0
\end{align*}
\]

Table 3: Initial table of the proposed method on example 3

<table>
<thead>
<tr>
<th>Variable in basis as appeared in z with s_i</th>
<th>x_1</th>
<th>x_2</th>
<th>x_3</th>
<th>x_4</th>
<th>d_1^-</th>
<th>d_1^+</th>
<th>d_2^-</th>
<th>d_2^+</th>
<th>d_3^-</th>
<th>d_3^+</th>
<th>s_1</th>
<th>s_2</th>
<th>Solution value b_i (rhs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>p_1 d_1^-</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>3p_1 d_2^-</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>2p_1 d_3^-</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>120</td>
</tr>
<tr>
<td>p_3 d_4^-</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>p_3 d_5^-</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>800</td>
</tr>
<tr>
<td>s_1</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>90</td>
</tr>
<tr>
<td>s_2</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>72</td>
</tr>
</tbody>
</table>
Column one is the basis which consists of the deviational variables in the objective function with priority and weight assigned to each of them, and slack variables from the rigid constraints in order at which they appeared in constraint equations.

Column one of row two is the deviational variable of the first goal constraint that appears in the objective function. Similarly, on the 3rd row of column one refers to the deviational variable of the 2nd goal constraint that appeared in the objective function with weight 3 attached to it. Row four of column one refers to the deviational variable from 3rd goal constraint that appeared in the objective function with weight 2 attached to it. These three rows correspond to priority p1. The rest of the entries are obtained in manner similar to that used to obtain the entries in previous examples.

The above problem can be solved using Orumie and Ebong (2011), but its initial table did not consider the slack and surplus rows from rigid constraints.

Another example (example 4) is a generalised case where both weighted and preemptive approaches are combined to form a model to a problem, and also d− and d+ (positive and negative deviational variables) from the same constraint equation are in the achievement function with different priorities assigned to each of them.

**Example 3.4**

\[
\min \ z = \ p_1 (d_1^- + 3d_2^- + 2d_3^-), \ p_2 (d_4^+), \ p_3 (d_5^+), \ p_4 d_1^+
\]

s.t

\[
\begin{align*}
& x_1 + x_2 + d_1^- - d_1^+ = 30 \\
& x_3 + x_4 + d_2^- - d_2^+ = 30 \\
& 3x_1 + 2x_3 + d_3^- - d_3^+ = 120 \\
& 3x_2 + 2x_4 + d_4^- - d_4^+ = 20 \\
& 10x_1 + 9x_2 + 8x_3 + 7x_4 + d_5^- - d_5^+ = 800 \\
& 6x_1 + 3x_2 \leq 90, \\
& 3x_1 + 6x_2 \leq 72 \\
& x_1, x_2, x_3, x_4, d_1, d_1^+, d_2, d_2^+, d_3, d_3^+, d_4, d_4^+, d_5, d_5^+ \geq 0
\end{align*}
\]

The initial table of the proposed method of solving the same example is given below;
**Table 4:** Initial table of example 4 for the proposed method

<table>
<thead>
<tr>
<th>Variable in basis as appeared in z with $s_i$</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$d_i^-$</th>
<th>$d_i^+$</th>
<th>$s_i$</th>
<th>$s_2$</th>
<th>Solution value $b_i$ (rhs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_1 d_i^-$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>$3p_1 d_i^-$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$2p_1 d_i^+$</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$p_1 d_i^+$</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$p_1 d_i^+$</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>$s_1$</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$s_2$</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

In example 4 above, it is observed that both $d_i^-$ and $d_i^+$ appeared in the objective function with different priorities attached to them. In the new method, if both $d^-$ and $d^+$ (negative and positive deviational variables) from the same goal constraint are in the achievement function but with different priorities, then the one with higher priority will be in the basis whereas the lesser one will be placed alongside with other variables in the non basis. This situation was not considered in the previous methods. Therefore, column one of row two is the deviational variable ($d_i^-$) of the first goal constraint that appears in the objective function which corresponds to priority $p_1$. The variable $d_i^+$ which corresponds to $p_4$ appears only in the non-basis, since the one with higher priority $d_i^-$ has been represented in basis. The rest of the entries are obtained in manner similar to that used to obtain the entries in previous examples.

**INITIAL TABLE FOR THE GENERAL LINEAR GOAL PROGRAMMING**

We now present the initial table of the general Linear GP using the guidelines we develop for our proposed method. The general linear GP we will consider is the following:
New Approach to Solving Generalised Linear Goal Programming Problem

Find \( \tilde{x} = (x_1, x_2, \ldots, x_n) \) that minimizes

\[
z = \{ L_1(d^-, d^+), L_2(d^-, d^+), L_3(d^-, d^+) \}
\]

subject to

\[
\sum_{j} a_{ij} x_j + d_i^- - d_i^+ = b_i \quad \text{for all } i = 1, \ldots, r
\]

\[
\sum_{j} c_{ij} x_j \leq h_i \quad \text{for all } i = 1, 2, \ldots, \eta
\]

and \( x_j \geq 0, d^- \geq 0, d^+ \geq 0 \)

where

\[
\sum_{j} a_{ij} x_j \quad \text{is the linear function on the left hand side of the } i^{th} \text{ goal constraint;}
\]

\[
\sum_{j} c_{ij} x_j \quad \text{is the linear function on the left hand side of the } i^{th} \text{ rigid constraint;}
\]

The initial table of the above problem (formulation) is given in Table 5 below. In the table, the basic variables are determined in the manner illustrated in the above examples. The \( d^v_i \) are the positive \( (d_i^+) \) and negative \( (d_i^-) \) deviational variables. If \( v_i = 1 \), then \(-v_i = -1\) and if \( v_i = -1 \), then \(-v_i = 1\) in the table.

**Table 5: Initial table of the New Algorithms on LGP**

| Variable in basis as appeared in z with \( s_i \) | \( x_1 \) | \( x_2 \) | \( \ldots \) | \( x_n \) | \( d_1^v \) | \( d_2^v \) | \( d_3^v \) | \( \ldots \) | \( d_r^v \) | \( s_1 \) | \( s_2 \) | \( \ldots \) | \( s_\eta \) | Solution value (rhs) |
|------------------------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| \( L_d d_k^v \) | \( a_{11} \) | \( a_{12} \) | \( \ldots \) | \( a_{1n} \) | \( -v_1 \) | \( 0 \) | \( 0 \) | \( \ldots \) | \( 0 \) | \( 0 \) | \( 0 \) | \( \ldots \) | \( 0 \) | \( b_1 \) |
| \( a_{21} \) | \( a_{22} \) | \( \ldots \) | \( a_{2n} \) | \( 0 \) | \(-v_2 \) | \( 0 \) | \( \ldots \) | \( 0 \) | \( 0 \) | \( 0 \) | \( \ldots \) | \( 0 \) | \( b_2 \) |
| \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( b_r \) |
| \( a_{r1} \) | \( a_{r2} \) | \( \ldots \) | \( a_{rn} \) | \( 0 \) | \( 0 \) | \( \ldots \) | \( -v_r \) | \( 0 \) | \( 0 \) | \( \ldots \) | \( 0 \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) |
| \( s_1 \) | \( c_{11} \) | \( c_{12} \) | \( \ldots \) | \( c_{1n} \) | \( 0 \) | \( 0 \) | \( \ldots \) | \( 0 \) | \( 1 \) | \( 0 \) | \( \ldots \) | \( 0 \) | \( h_1 \) |
| \( s_2 \) | \( c_{21} \) | \( c_{22} \) | \( \ldots \) | \( c_{2n} \) | \( 0 \) | \( 0 \) | \( \ldots \) | \( 0 \) | \( 0 \) | \( 1 \) | \( \ldots \) | \( 0 \) | \( h_2 \) |
| \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) | \( \ldots \) |
\[ L_k d_k^i : k = 1, 2, \ldots, r \] in row 2 column 1 in table 5 are obtained as illustrated in the examples above.

The solution procedure for the above problem in table 5 considers the goal constraints \((g_k)\) as both objective functions and the constraints, where \(g_k\) is the \(k^{th}\) priority row(s). That is, there are no inclusions of \(c_j\) rows on the table. It starts by not including the deviational variables column that did not appear in the achievement function in the tableau while searching for the optimal solution, but can be augmented when necessary. This is because, positive deviational variables columns coefficient is the same as negative of the negative deviational variable column coefficient, that is \(d_i^- = -d_i^+\). In this new method, the algorithm starts with the highest priority row constraints and terminates with the lowest priority. Solution is obtained provided that for all \(p_k\) the steps in the algorithm are strictly maintained as shown in section 4 below.

**MATERIALS AND METHOD (THE MODIFIED ALGORITHM FOR THE LGP MODEL)**

Let \(p_k\) be the \(k^{th}\) priority assigned to deviational variable \((d_k^i)\) which is associated with \(L_k\); where \(p_1 > p_2 \ldots > p_K\) and \(K\) is the number of priority level in \(z\). Let \(g_k\) be the priority row or row (as applicable) corresponding to \(p_k\), then the algorithm is as follows:

Step 1. Initialization:
   
   Set \(k \leftarrow 1\)

Step 2. Feasibility:
   
   If \(b_i < 0\) for \(i = 1, 2, \ldots, r\) then stop. \{Solution infeasible\}

Step 3. Optimality test:

   - If \(k > K\), stop.
   - If all the entries of \(g_k\) are less than or equal to zero except for the value corresponding to the column of \(d_k^i\) under consideration and absolute value
of coefficient of deviational variable that are not in the table, go to Step 7.  
{All coefficients of priority row \( g_k \) non positive, so \( p_k \) is satisfied}.

- If the right hand side corresponding to \( p_k \) is 0, go to step 7
- If \( b_i = 0 \), for \( i=1,2,\ldots,r \) then stop.  \{Solution optimal\}
- If all the coefficients of \( g_k \) row are all less than or equal to zero for all \( k \), \( k=1,2,\ldots,K \), then stop.  \{Compromised solution reached\}.

Step 4. Entering variable:

(a) Entering variable is the variable with highest positive coefficient (if it is decision or slack variable) or highest absolute value of coefficient (if deviational variable) in the row \( g_k \). In the case of deviational variable with highest absolute coefficient, we enter the negative of the deviational variable (noting that \(-d^- = d^+\)), provided that the deviational variable to enter is not already in the table.

(b) In case of tie in the coefficients, the entering variable becomes the variable for which the minimum of the ratio of the right hand side to the positive coefficients of the columns corresponding to the ties is maximum.

(c) In case of tie in (b), the variable with the least priority will become the entering variable.

(Priority attached to the entering variable should be placed alongside with it into the basis).

Step 5. Leaving variable:

i. If \( y_0 \) is the column corresponding to the entering variable in Step 4, then the leaving variable is the basic variable with minimum ratio of the right hand side value to the positive entries in \( y_0 \). If \( y_0 \) is deviational variable \((d^+)\) with highest absolute coefficient, then coefficient of \( d^+ \) will become coefficient of \(-d^-\).

ii. In case of tie in (i), the variable with the smallest right hand side leaves the basis.

iii. In case of tie in (ii), the variable with the highest priority leaves the
Step 6. Perform Gauss Jordan row operations to update the table just as in simplex method of solving LPP. If $p_k$ row(s) are still in the basis of the updated table, go to Step 2.

Step 7. Increment process:
Set $k \leftarrow k + 1$, go to Step 2

REMARKS:
1. The algorithm stops if
   i.) The coefficient of the priority rows are all negative or zero for all $k$
       {compromise solution}
   ii.) The right hand sides of the priority rows are all zero  (optimal solution)
   iii.) The priority rule is satisfied and $z$ cannot be achieved further
       (compromised solution)

2. The solution is the value of the prioritised deviational variables in the objective function as appeared in the last iteration tableau.

3. Just as in the method of artificial variables, ensure that a variable of higher or equal priority that has been previously satisfied does not re-enter the basis, instead the variable with the next higher coefficient in $g_k$ enters.

4. If both $d^-$ and $d^+$ (negative and positive deviational variables) from the same goal constraint are in the achievement function but with different priorities, then the one with higher priority will be in the basis whereas the lesser one will be placed alongside with other variables in the non basis.

THE MODIFIED ALGORITHM ON WEIGHTED GOAL PROGRAMMING PROBLEMS
In weighted goal programming problem, the algorithm in 4 will be modified to suit its variant. The variable with the highest coefficient in $z$ is assigned weight $w_1$; $w_1 > w_2 >, ..., > w_F$ and $F$ is the number of weight in $z$. Relating this to algorithm in 4, $w_f; f=1,2, ..., F$ takes the place of $p_k$ and $g_f$ takes the place of $g_k$ and the algorithm follows.

The solution becomes the value of the weighted sum of all the deviational variables in
the objective function as appeared in the last iteration tableau; that is the value of the achievement function becomes a single-valued function.

THE NEW ALGORITHM ON GENERALISED MODEL (GGP)

Recall that in generalised goal programming variant, goals are categorised into groups, where the goals within each group are of equal importance, but there are slight differences between the groups in their level of importance such that weight is used within each group). In this case, the algorithm in section 3.7 will be modified in order to suit it.

Let $p_k$ be the $k^{th}$ priority assigned to deviational variable(s) $(d_{i,v})$ which is associated with group $L_k$; where $p_1 > p_2 > \ldots > p_k$ and $K$ is the number of priority level in $z$. Let $g_k$ be the set of rows corresponding to $k^{th}$ priority $p_k$.

Also, Let $w_f$ be the $f^{th}$ weight assigned to $(d_{i,v})$ which is associated with $p_k$ in group $L_k$; where $w_f > 0 \forall f$, then the algorithm is similar to that of 3.7 but with modification in step 4 as shown below follows;

Step 4. Entering variable:
1. Entering variable is the variable with the highest coefficient in $g_k$.
2. When two or more deviational variables in $z$ have the same priority, then the entering variable is considered from the deviational variable row(s) with the highest weight ($w_f$). This will be the variable with highest positive coefficient (if it is decision or slack variable) or highest absolute value of coefficient (if deviational variable) in the row(s) corresponding to $w_f$.
3. In case of ties in the coefficient of row(s), then the entering variable is the variable for which minimum of the ratio of the right hand side to the positive coefficients of the columns corresponding to the ties is maximum.
4. In case of tie in (3), the variable with the least priority or weight will become the entering variable (Priority or weight attached to the entering variable should be placed alongside with it into the basis).

Go through step 5 to 6 and check if $w_1$ is still in the basis. Continue the process until $f=F$, then go to step 7.

Then solution becomes a vector of priority level in $z$. 
**STEPS BY STEP ILLUSTRATION OF THE PROPOSED ALGORITHM ON GENERALISED MODEL (GGP)**

The problem in (5) below will be solved using the new proposed approach.

**Problem 5**

\[
\begin{align*}
\text{min} & \quad z = p_1 d_1^- + p_2 (w_1 d_1^- + w_2 d_3^-) \\
\text{s.t} & \quad 10 x_1 + 12 x_2 + d_1^- - d_1^+ = 2000 \\
& \quad 6 x_1 + 12 x_2 + d_2^- - d_2^+ = 120 \\
& \quad 8 x_1 + 4 x_2 + d_3^- - d_3^+ = 64
\end{align*}
\]

Problem 5 above has two groups. Group one has only one deviational variable, where as group two has two deviational variable. Within the second group weight “\(w_1\) and \(w_2\)” are used.

**Table 5.1:** Initial Table of problem 5

<table>
<thead>
<tr>
<th></th>
<th>(x_1)</th>
<th>(x_2)</th>
<th>(d_1^-)</th>
<th>(d_2^-)</th>
<th>(d_3^-)</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(p_2 w_1 d_1^-)</td>
<td>10</td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2000</td>
</tr>
<tr>
<td>(p_1 d_2^-)</td>
<td>6</td>
<td>12</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>120</td>
</tr>
<tr>
<td>(p_2 w_2 d_3^-)</td>
<td>8</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>64</td>
</tr>
</tbody>
</table>

\(x_2\) enters, \(d_2^-\) leaves

Column one represents the variables in the achievement function with priorities and weights assigned to each of them which forms the bases. Column two and three represent the coefficients of the decision variables in the goal constraints equation. Column four to six represent coefficients of deviational variables (\(-v_i\)) in the goal constraint equations that appeared in the achievement function. Applying the algorithm,

**Step 1.** Initialization: Here, \(k=1\), \(r=3\), \(K=2\)

Set \(k \leftarrow 1\)

**Step 2.** \(b_1, b_2, b_3 > 0\) \{So solution feasible.\}
Step 3. $g_k = \{6, 12, 0, 1, 0\}$. Therefore, some entries of $g_k$ is non negative, and $b_2 \neq 0$. {So solution not optimal}.

Step 4. $\max g_1 = \max\{6, 12, 0, 1, 0\} = 12$ which corresponds to $x_2$. So $x_2$ enters the bases.

Step 5. $y_0 = \{g_{i2}\} \Rightarrow \min \left\{ \frac{\text{RHS}}{g_{i2}}, g_{i2} > 0, \right\} = \min \{2000/12, 120/12, 64/4\} = 10$ at $d_2^-$. Therefore $d_2^-$ leaves the bases.

Step 6. Perform Gauss Jordan row operations to update the table and check if $p_i$ is still in the basis to test for feasibility and optimality. (see tableau 5.2)

**Table 5.2:** 1st iteration table of problem 5

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$d_1^-$</th>
<th>$d_2^-$</th>
<th>$d_3^-$</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_2 w/d_1^-$</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>1880</td>
</tr>
<tr>
<td>$x_2$</td>
<td>1/2</td>
<td>1</td>
<td>0</td>
<td>1/12</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>$p_2 w_2d_2^-$</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>-1/3</td>
<td>1</td>
<td>24</td>
</tr>
</tbody>
</table>

$x_1$ enters, $d_3^-$ leaves

Table 5.2 shows that $p_i$ has left the basis, so go to step 7.

Step 7. Set k=2, go to Step 2.

Step 2. $b_1, b_3 > 0$ {So solution feasible.}

Step 3. $g_k = \{4, 0, 1, -1, 0\}, \{6, 0, 0, -1/3, 1\}.$ Therefore, some entries of $g_k$ is non negative, and $b_1$ and $b_2 \neq 0$. {So solution not optimal}.

Step 4. Therefore the entering variable will be choosing from the deviational variable row with the highest weight which is $p_2 w_2d_2^-$. $\Rightarrow \max \{ \{4, 0, 1, -1, 0\}\} = 4$ at $g_{i1}$. So $x_1$ enters the basis.

Step 5. $y_0 = g_{i1} \Rightarrow$ Therefore, $\min \left\{ \frac{\text{RHS}}{g_{i1}}, g_{i1} > 0, \right\} = \min \{1880/4, 20, 24/6\} = 4$. So, in this case, $d_3^-$ leaves the bases.
Step 6. Perform the same operation to update the new tableau and check if \( p_2 \) \( w_1 \) is still in the basis to test for optimality. (See Table (5.3).

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( d_i^- )</th>
<th>( d_2^- )</th>
<th>( d_3^- )</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_2 ) ( w_1 ) ( d_1 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-7/9</td>
<td>-2/3</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1/9</td>
<td>-1/12</td>
</tr>
<tr>
<td>( x_1 )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1/18</td>
<td>1/6</td>
</tr>
</tbody>
</table>

\( d_2^+ \) enters
\( X_1 \) leaves

Table (5.3) shows that \( p_2 \) \( w_1 \) is in the basis, so go to step 2.

Step 2. \( b_1 > 0 \) {So solution feasible.}

Step 3. \( g_k = g_2 = \{0, 0, 1, [-7/9], [-2/3]\} \). Therefore, some entries of \( g_k \) is non-negative, and \( b_1 \neq 0 \). {So solution not optimal}.

Step 4. \( \max \{0, 0, 1, [-7/9], [-2/3]\} = 7/9 \) at \(-g_{14} \) which corresponds to \( d_2^+ \). Recall that \( d_i^+ = -d_i^- \). Therefore, \(-(-g_{14}) = g_{14} = 7/9 \). So \( d_2^+ \) enters the basis.

Step 5. \( y_0 = -g_{14} : \{ \frac{RHS}{-4} : -g_{14} > 0 \} = \left\{ \left( \frac{1864 * 9}{7} \right), (18 * 4) \right\} = 72 \) which corresponds to \( x_1 \). So, \( x_1 \) leaves and column for \( d_2^+ \) created.

Step 6. Perform the same operation to update the new tableau and check if \( p_2 \) is still in the basis to test for feasibility and optimality. (See Table (5.4).
New Approach to Solving Generalised Linear Goal Programming Problem

Table 5.4: 3rd iteration table of problem 5

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$d_1^-$</th>
<th>$d_2^-$</th>
<th>$d_3^-$</th>
<th>$d_2^+$</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_2 \cdot w_1 d_2$</td>
<td>-14</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-3</td>
<td>0</td>
<td>1808</td>
</tr>
<tr>
<td>$x_2$</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1/4</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>$d_2^+$</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>3</td>
<td>1</td>
<td>72</td>
</tr>
</tbody>
</table>

$d_3^+$ enters  
$d_1^-$ leaves

Table (5.4) shows that $p_2 \cdot w_1 d_1^-$ is still in the basis, so go to step 2.

Step 2.  

$b_1 > 0$  
\{So solution feasible.\}

Step 3.  
$g_k = \{-14, 0, 1, 0, \{-3, 0\}\}$. Therefore, some entries of $g_k$ is non negative, and $b_1 \neq 0$. \{So solution not optimal\}.

Step 4.  
Max =\{-14, 0, 1, 0, \{-3, 0\}\}=3 which is $-d_3^- = d_3^+$. So $d_3^+$ enters the basis.

Step 5.  
$y_0 = -g_5^* : \min(\frac{RHS}{x_5}: -g_5^* > 0) = \min\{\frac{1808}{3}, -, -\} = 1808/3$ which corresponds to $d_1^-$. So, $d_1^-$ leaves and a column of is created for $d_3^+$ created.

Step 6.  
Perform the same operation to update the new tableau and check if $p_2 \cdot w_2 d_1^-$ is still in the basis to test for feasibility optimality. (See Table (5.5)).

Table 5.5: 4th and final iteration table of problem 5

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$d_1^-$</th>
<th>$d_2^-$</th>
<th>$d_3^-$</th>
<th>$d_2^+$</th>
<th>$d_3^+$</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_3^+$</td>
<td>-14/3</td>
<td>0</td>
<td>1/3</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>1808/3</td>
</tr>
<tr>
<td>$x_2$</td>
<td>-3/2</td>
<td>1</td>
<td>1/12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>500/3</td>
</tr>
<tr>
<td>$x_1$</td>
<td>-4</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1880</td>
</tr>
</tbody>
</table>
Table (5.5) shows that $p_2 w_1$ has left the basis, so go to step 7.

**Step 7.** Set $k=3$, go to step 2

**Step 2.** $b_i > 0$ \{So solution feasible.\}

**Step 3.** $k>K=2$, so stop.

The current solution is $z = p_1 (d_1^-), p_2 (3d_1^- + d_3^+) = 0$

**ILLUSTRATION OF THE ALGORITHM ON GENERALISED MODEL**

(WHEN BOTH $d^-$ AND $d^+$ (POSITIVE AND NEGATIVE DEVIATIONAL VARIABLES) FROM THE SAME CONSTRAINT ARE IN THE ACHIEVEMENT FUNCTION $Z$)

The problem in (6) below as shown in Journal of operational Res. Soc. Vol 33, No3, 1983, will be solved using the new proposed method.

**Problem 6**

$$\text{Min } z = p_1 d_1^- + p_2 d_2^- + p_3 d_3^+ + p_4 d_1^+$$

s.t

$$2x_1 + 4x_2 + d_1^- - d_1^+ = 80$$

$$8x_1 + 10x_2 + d_2^- - d_2^+ = 320$$

$$8x_1 + 6x_2 + d_3^- - d_3^+ = 240$$

$x_i \geq 0, d_i^+, d_i^- \geq 0, d_i^- \cdot d_i^+ = 0$

Jnl of optl Res. Soc. Vol 33, No3, 1983

**Table 6.1:** Initial iteration of problem 6

<table>
<thead>
<tr>
<th>Basis</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$d_1^-$</th>
<th>$d_2^-$</th>
<th>$d_3^+$</th>
<th>$d_1^+$</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_1 d_1^-$</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>80</td>
</tr>
<tr>
<td>$p_2 d_2^-$</td>
<td>8</td>
<td>10</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>320</td>
</tr>
<tr>
<td>$p_3 d_3^+$</td>
<td>8</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>240</td>
</tr>
</tbody>
</table>

$x_2$ enters, $d_1^-$ leaves
Column one is the basis which consists of the deviational variables in $z$ with priority attached to each of them. Here, $d_1^-$ and $d_1^+$ (positive and negative deviational variables) from the first constraint are in the achievement function $z$ (i.e. $p_1 d_1^-$ and $p_4 d_1^+$). Therefore, the one with highest priority which is $d_1^-$ is placed on the basis, whereas the lesser one $d_1^+$ is placed alongside with other variables in the non basis. Column two to six is the matrix of the coefficients of the decision variables from both goal and rigid constraints and deviational variables that appeared in $z$.

Row one to three are the coefficients of the variables from the constraint equations corresponding to $p_1$, $p_2$, $p_3$ ($g_1, g_2, g_3$) respectively. The variable $d_1^+$ which corresponds to $p_4$ appeared only in the non-basis, since the one with higher priority $d_1^-$ has been represented in basis.

Applying the algorithm,

Step 1. Initialization: Here, $k=1$, $r=4$, $K=4$

Set $k \leftarrow 1$

Step 2. $b_1, b_2, b_3>0$ {So solution feasible.}

Step 3. $p_k=g_k\{2,4,1,1,0,-1\}$. Some entries of $g_k$ is non negative, and $b_1 \neq 0$

So solution not optimal}.

Step 4. Max $g_k=\max\{2,4,1,1,0,-1\}=4$ which corresponds to $x_2$. So $x_2$ enters the bases.

Step 5. $y_0=\min \{ \frac{RHS}{g_{i_2}}, g_{i_2} > 0 \} = \min \{ 80/4, 320/10, 240/6 \}=20.$

Therefore $d_1^-$ leaves the basis.

Step 6. Perform Gauss Jordan row operations to update the table and check if $p_1$ is still in the basis to test for feasibility and optimality. (see tableau 6.2).

Table 6.2: 1st iteration of problem 6

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$d_1^-$</th>
<th>$d_2^-$</th>
<th>$d_3^+$</th>
<th>$d_1^+$</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_2$</td>
<td>1/2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1/4</td>
<td>20</td>
</tr>
<tr>
<td>$p_2$ $d_2^-$</td>
<td>3</td>
<td>0</td>
<td>-10/4</td>
<td>1</td>
<td>0</td>
<td>10/4</td>
<td>120</td>
</tr>
<tr>
<td>$p_3d_3^+$</td>
<td>5</td>
<td>0</td>
<td>-6/4</td>
<td>0</td>
<td>-1</td>
<td>6/4</td>
<td>120</td>
</tr>
</tbody>
</table>

$x_1$ enters, $d_3^+$ leaves
Table (6.2) shows that \( p_1 \) has left the basis, so go to step 7.

Step7. Set \( k=2 \), go to Step 2.

Step 2. \( b_2, b_3>0 \) \{So solution feasible.\}

Step 3. \( g_k = \{3,0, -10/4, 1, 0, 10/4\} \). Some entries of \( g_k \) is non negative, and \( b_2 \neq 0 \) \{So solution not optimal\}.

Step 4. Max\{3, 0, -10/4, 1, 0, 10/4\} = 3. So \( x_1 \) enters the basis.

Step 5. \( y_0 = g_i \Rightarrow \) Therefore, \( \min \left\{ \frac{\text{RHS}}{g_i} : g_{i1} > 0 \right\} = \min\{40, 40, 24\} = 24. \) So, in this case, \( d_3^+ \) leaves the bases.

Step 6. Perform the same operation to update the new tableau and check if \( p_2 \) is still in the basis to test for feasibility and optimality. (See Table (6.3)).

**Table 6.3:** 2\textsuperscript{nd} iteration of problem 6

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( d_1^- )</th>
<th>( d_2^- )</th>
<th>( d_3^+ )</th>
<th>( d_1^+ )</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_2 )</td>
<td>0</td>
<td>1</td>
<td>2/5</td>
<td>0</td>
<td>1/10</td>
<td>-2/5</td>
</tr>
<tr>
<td>( p_2 )</td>
<td>0</td>
<td>0</td>
<td>-8/5</td>
<td>1</td>
<td>3/5</td>
<td>8/5</td>
</tr>
<tr>
<td>( d_2^- )</td>
<td>0</td>
<td>1</td>
<td>-3/10</td>
<td>0</td>
<td>-1/5</td>
<td>3/10</td>
</tr>
</tbody>
</table>

\( d_1^+ \), enters, \( d_2^- \) leaves

Table (6.3) shows that \( p_2 \) is still in the basis, so go to step 2.

Step 2. \( b_2>0 \) \{So solution feasible.\}

Step 3. \( p_k = g_k = \{0, 0, -8/5, 1, 3/5, 8/5\} \). Some entries of \( g_k \) is non negative, and \( b_2 \neq 0 \) \{So solution not optimal\}.

Step 4. Max \( g_k = \max \{0, 0, -8/5, 1, 3/5, 8/5\} = 8/5 \) which corresponds to \( d_1^- \). Therefore, \( d_1^- \) enters the basis.

Step 5. \( y_0 = g_i \Rightarrow \min \left\{ \frac{\text{RHS}}{g_i} : g_{i6} > 0 \right\} = \left\{ -\left( \frac{48 * 5}{8} \right), \left( \frac{24 * 10}{3} \right) \right\} = \{-30, 80\} = 30 \)
which corresponds to \( d_2^- \). So, \( d_2^- \) leaves.

Step 6. Perform the same operation to update the new tableau and check if \( p_2 \) is still in the basis to test for feasibility and optimality.
Table 6.4: Last iteration of problem 6

<table>
<thead>
<tr>
<th></th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$d_1^-$</th>
<th>$d_2^-$</th>
<th>$d_3^+$</th>
<th>$d_4^+$</th>
<th>RHS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_2$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>$\frac{1}{4}$</td>
<td>1/4</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>$p_4 d_4^+$</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>$\frac{5}{8}$</td>
<td>3/8</td>
<td>1</td>
<td>30</td>
</tr>
<tr>
<td>$x_1$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>$-\frac{3}{16}$</td>
<td>$-\frac{5}{16}$</td>
<td>0</td>
<td>15</td>
</tr>
</tbody>
</table>

Table (6.4) shows that $p_2$ has left the basis, so go to step 7.

Step 7.  Set $k = 4$.

Step 2.  $b_2 > 0$  {So solution feasible.}

Step 3.  $g_k = \{0, 0, -1, \frac{5}{8}, \frac{3}{8}, 1\}$. Some entries of $g_k$ is non-negative, and $b_2 \neq 0$ {So solution not optimal}.

Step 4.  Max $g_k = \max \{0, 0, -1, \frac{5}{8}, \frac{3}{8}, 1\} = 1$ at which corresponds to $d_1^+$. But this cannot enter for itself to leave. So consider the next higher value which is $\frac{5}{8}$ that corresponds to $d_2^-$. But variable with higher priority will not re-enter for lesser one to leave. So we consider the next higher value which is $\frac{3}{8}$ that corresponds to $d_3^+$ with higher priority also. Since there is no other positive variable in the row, go to 7.

Step 7.  Set $k = 5$.

Step 2.  $b_2 > 0$  {So solution feasible.}

Step 3.  $k > K$, so stop.

Therefore, $z = p_1 d_1^- + p_2 d_2^- + p_3 d_3^+ + p_4 d_4^+ = \{0, 0, 0, 30\}$.

**SUMMARY AND CONCLUSION**

A new approach for solving GLGP is developed, along side with modification of the existing ones. Difference structure of LGP were enumerated and solved using the new approach. The new algorithm is efficient.
REFERENCES


