A Deterministic Global Optimization Method for Solving Generalized Linear Multiplicative Programming Problem with Multiplicative Constraints

Bo ZHANG and Yue-lin GAO*  
Research Institute of Information and System Computation Science, North Minzu University, Yinchuan, 750021, China  
*Corresponding author

Keywords: Global optimization, Linear multiplicative programming, Linear relaxation, Branch and bound.

Abstract. This paper presents a deterministic global optimization algorithm for solving generalized linear multiplicative programming problem with multiplicative constraints (GLMP). By utilizing equivalent transformation and linear relaxation method, a linear relaxation programming (LRP) of equivalent problem (GLMPH) is established. In the algorithm, lower and upper bounds are simultaneously obtained by solving some linear relaxation programming problems (LRP). Global convergence has been proved and results of some sample examples and a small random experiment show that the proposed algorithm is feasible and efficient.

Introduction

The Generalized linear multiplicative programming(GLMP) refers to a type of optimization problems, which contains the sum of the product terms of two real functions in objective or constraint functions, is a special category of multiplicative programming. The plentiful researchers and scholars have attached importance to this topic for many years. In this paper, we consider the following generalized linear multiplicative programming (GLMP):

\[
\begin{align*}
\min & \quad G_s(x) = \sum_{i=1}^{p_s} \left( c_{si}^T x + d_{si} \right) \left( e_{si}^T x + f_{si} \right) \\
\text{s.t.} & \quad G_i(x) = \sum_{i=1}^{p_i} \left( c_{si}^T x + d_{si} \right) \left( e_{si}^T x + f_{si} \right) \leq 0, \quad s = 1, 2, \ldots, N, \\
& \quad x \in X = \{ x \in \mathbb{R}^n \mid Ax \leq b \}.
\end{align*}
\]

Where \( c_{si} \in \mathbb{R}^n, e_{si} \in \mathbb{R}^n, i = 1, 2, \ldots, p_s, \) \( s = 1, 2, \ldots, N \), while the constant terms \( d_{si}, f_{si} \) are all arbitrary real numbers, \( A \in \mathbb{R}^{m \times n} \) is a matrix, \( b \in \mathbb{R}^m \) is a vector, set \( X \) is nonempty and bounded. During the past years, (GLMP) has received significant attention in the literature because there are many practical applications in different fields of research, including multiple-objective decision[1], robust optimization[2], plant layout design[3], financial optimization[4] and so on. Of course, there are many ways to solve this problem and we can see these methods in articles[5-11].

New Linearization Skill

For each \( j = 1, 2, \ldots, n \), compute \( l_j = \min_{x \in X} x_j \) and \( u_j = \max_{x \in X} x_j \), and define the vector \( l \) and \( u \) as follows: \( l = (l_1, l_2, \ldots, l_n)^T \in \mathbb{R}^n, u = (u_1, u_2, \ldots, u_n)^T \in \mathbb{R}^n \). Obviously, \( l_j \leq x_j \leq u_j \). Then, the hyper-rectangle \( H = [l, u] = \{ x \mid l \leq x \leq u \} \), which can cover the feasible domain \( F = X \) is obtained.

It is distinct that

\[
G_s(x) = \sum_{i=1}^{p_s} \left( c_{si}^T x + d_{si} \right) \left( e_{si}^T x + f_{si} \right) = x^T (c_s e_s^T) x + (c_s f_s + e_s d_s)^T x + d_s^T f_s.
\]

Where,
\[ c_i \in R^{n \times p_i}, e_i \in R^{n \times p_i}, d_i \in R^{p_i}, f_i \in R^{p_i}, s = 1, 2, \cdots, N. \]

In view of \( x^T (c_i e_i^T)x = x^T (e_i c_i^T)x \), we can easily obtain

\[
x^T (c_i e_i^T)x = \frac{x^T (c_i e_i^T)x + x^T (e_i c_i^T)x}{2} = x^T \left( \frac{c_i e_i^T + e_i c_i^T}{2} \right)x.
\]

Let \( Q = \frac{c_i e_i^T + e_i c_i^T}{2} \), then, \( Q \in R^{n \times n} \) is a real symmetric matrix. Assume \( \lambda_{\min} \) is the minimum eigenvalue of the matrices \( Q \), for \( i = 1, 2, \cdots, p \). If \( \lambda_{\min} \geq 0 \), let \( \eta_i = 0 \); otherwise, let \( \eta_i = |\lambda_{\min}| + \sigma \), where \( \sigma \geq 0 \) then \( Q_i + \eta_i I \) is semi-positive definite.

On the rectangle \( H^t = \{ x \in R^n : l^t \leq x \leq u^t \} \subseteq H \), for each \( s \), we construct a linear lower function of \( G_i(x) \) on \( H^t \). Because

\[ G_i(x) = x^T (c_i e_i^T)x + (c_i f_i + e_i d_i)^T x + d_i^T f_i, \]

\[ = \phi_i (x) + (c_i f_i + e_i d_i)^T x + d_i^T f_i + \eta_i \phi(x) \]

(1)

Where, \( \phi_i (x) = x^T (Q_i + \eta_i I)x \) is a convex function and \( \phi(x) = \sum_{j=1}^{n} (-x_j^2) \) is a concave function. Then without any loss of generality, we can reformulate the (GLMP) problem as the following problem (GLMPH):

\[
\begin{align*}
\text{min} & \quad G_i(x) = \phi_i (x) + (c_i f_i + e_i d_i)^T x + d_i^T f_i + \eta_i \phi(x) \\
\text{s.t.} & \quad G_i(x) = \phi_i (x) + (c_i f_i + e_i d_i)^T x + d_i^T f_i + \eta_i \phi(x) \leq 0, s = 1, 2, \cdots, N \\
& \quad x \in H \setminus X.
\end{align*}
\]

Suppose \( l_i^t \) and \( u_i^t \) is the \( j \)-th indicator of \( l^t \) and \( u^t \) respectively. We know that, for each \( j \in \{1, 2, \cdots, n \} \), the linear equation through the two endpoints on the parabola \( x_j^t \) is \( L: y_j = (l_j^t + u_j^t)x_j - l_j^t u_j^t \geq x_j^t \), therefore, \( -x_j^t \geq -y_j = -(l_j^t + u_j^t)x_j + l_j^t u_j^t \), and then,

\[
\phi(x) = \sum_{j=1}^{n} (-x_j^2) \geq \sum_{j=1}^{n} \left[ -(l_j^t + u_j^t)x_j + l_j^t u_j^t \right] = -(l^t + u^t)x + (l^t)^T u^t, x \in H^k.
\]

(2)

According to the convexity of \( \phi_i (x) \), we can have

\[ \phi_i (x) \geq \phi_i (x_0) + (x - x_0)^T \nabla \phi_i (x_0). \]

On the basis of the sub-differential definition, we know that \( \nabla \phi_i (x_0) = \nabla \phi_i (x) \). Without loss of generality, for all the arbitrary real vector \( x_0 \in H^k \), so \( \nabla \phi_i (x_0) = \nabla \phi_i (x_0) = 2(Q_i + \eta_i I)x_0 \).

This implies that

\[ \phi_i (x) \geq 2x_0^T (Q_i + \eta_i I)x - x_0^T (Q_i + \eta_i I)x_0, x \in H^k. \]

(3)

Let \( x_0 = x_0^t = \frac{l^t + u^t}{2} \), thereupon, the lower bound estimation function of \( \phi(x) \) and \( \phi_i (x) \) is obtained, we construct the following linear function

\[ G_i^t(x) = (\alpha_i^t)^T x + \beta_i^t, x \in H^k, s = 0, 1, 2, \cdots, N. \]

(4)

where,
\[
\alpha^*_s = 2(Q_s + \eta_s I) \left( \frac{t^* + u^*}{2} \right) + c_s f_s + e_s d_s - \eta_s (l^* + u^*).
\]
\[
\beta^*_s = d^*_s f_s + \eta_s (l^*)^T u^* - \left( \frac{t^* + u^*}{2} \right) (Q_s + \eta_s I) \left( \frac{t^* + u^*}{2} \right).
\]

**Theorem 1** For each \( s = \{0,1,2,\cdots, m\} \), let \( Q_s + \eta_s I \) be semi-positive definite. For each \( s = \{0,1,2,\cdots, N\} \), the linear function \( G^*_s(x) \) is a lower function of \( G^*_s(x) \) on the rectangle \( H^s \). i.e. \( G^*_s(x) \geq G^*_s(x), \forall x \in H^s \).

**Proof** From the formula (1), (2) and (3), for each \( s = \{0,1,2,\cdots, N\} \), we have

\[
G_s(x) = \phi(x) + (c_s f_s + e_s d_s)^T x + d^*_s f_s + \eta_s \phi(x) \geq (\alpha^*_s)^T x + \beta^*_s, \forall x \in H^s.
\]

Consequently, \( G^*_s(x) \geq G^*_s(x), \forall x \in H^s, s = \{0,1,2,\cdots, N\} \).

**Theorem 2** Let \( \varepsilon_s = u_j - l_j, (j = 1,2,\cdots, n) \), for every \( x \in H^s \), when \( \varepsilon_s \to 0 \), the gap between \( G_s(x) \) and \( G^*_s(x) \) satisfies \( G_s(x) - G^*_s(x) \to 0, s = \{0,1,2,\cdots, N\} \).

**Proof** For convenience of proving Theorem 2, Let \( x^*, x^*_1, x^*_2, \cdots, x^*_m \) denote \( x, x_1, x_2, l, u \) at the beginning of the k-th iteration, respectively. From the formula (1) and (4), we have

\[
G_s(x^*_s) - G^*_s(x^*_s) = \left( x^*_s - x^*_md \right)^T (Q_s + \eta_s I) (x^*_s - x^*_md) + \eta_s (x^*_s - l^*)^T (u^*_s - x^*_s)
\]

\[
\leq \rho(Q_s + \eta_s I) \left\| x^*_s - x^*_md \right\|^2 + \eta_s \left\| x^*_s - l^* \right\|^2 \left\| x^*_s - x^*_s \right\|^2
\]

\[
= (\rho(Q_s + \eta_s I) + \eta_s) \left\| x^*_s - l^* \right\|^2.
\]

Where \( \rho(Q_s + \eta_s I) \) is the spectral radius of the rectangle \( Q_s + \eta_s I \), apparently, when \( \varepsilon_s \to 0, j = 1,2,\cdots, n \),

\[
\left\| u - l \right\|^2 \to 0, G_s(x) - G^*_s(x) \to 0, s = \{0,1,2,\cdots, N\} \). Hence, the conclusion is established.

Therefore, from the theorem 1, we obtain the linear relaxation programming problem of \( (GLMPH) \) on the rectangle \( H^s \):

\[
\begin{align*}
\text{min} & \quad G^*_s(x) \\
\text{st} & \quad \alpha^*_s x \leq -\beta^*_s, s = 1,2,\cdots, N, \\
& \quad x \in H^s \cap X.
\end{align*}
\]

According to the above, we only need to solve the problem \( (LRP^s) \), whose optimal value \( v(LRP^s) \) is a lower bound of the global optimum value \( v(GLMPH) \) of the problem \( (GLMPH) \) on rectangle \( H^k \), the optimal value \( v(LRP^k) \) also is an effective lower bound of the global optimum value \( v(GLMP^k) \) of the original problem \( (GLMP) \) on \( H^k \), i.e. \( v(LRP^k) \leq v(GLMP^k) \).

Meanwhile, Theorem 2 ensures that problem \( (LRP^s) \) can infinitely approximate the primitive problem, as \( \left\| X \right\| \to 0 \), this will guarantee the global convergence of the proposed algorithm.

**The Subdivision and Reducing of the Rectangle**

In this section, the bisection and reducing methods of the rectangle are given. Let \( H^s = [l^s, u^s] \subseteq H \) is a rectangle on \( R^n \), and \( x^s \in H^s \).

**The Subdivision OF the Rectangle**

The method of the subdivision of the sub-hyper-rectangle is depicted as follows:

Firstly, computing \( \omega = \max \{x^s_j - l^s_j, u^s_j - x^s_j\}; j = 1,2,\cdots, n \} \). if \( \omega = 0 \), select the longest edge of the rec-Tangle \( H^s \), i.e. \( u^s_j - l^s_j = \max \{u^s_j - l^s_j; j = 1,2,\cdots, n\} \), then \( x^s_j = (u^s_j + l^s_j)/2 \). if \( \omega \neq 0 \), then finding the first \( x^s_j \in \arg \max \omega \), let \( x^s_j = x^s_j \). Secondly, let \( x^s = (l^s_1, l^s_2, \cdots, l^s_j, x^s_j, l^s_{j+1}, \cdots, l^s_n) \), \( x^s = (u^s_1, u^s_2, \cdots, u^s_j, x^s_j, u^s_{j+1}, \cdots, u^s_n) \),

9
Through the straight line or plane about the point \( x' \) and point \( x'' \), the rectangular \( H^k \) is divided into two sub super rectangular \( H^{1k} = [l', u'^{i}] \) and \( H^{2k} = [l'', u''^{k}] \), then the sub-super-rectangular \( H^{1k} \) and \( H^{2k} \) are respectively:

\[
H^{1k} = \prod_{j=1}^{\mu_1} \left[ l^{(j)}_k, u_j^{(k)} \right] \times \prod_{j=\mu_1+1}^{n} \left[ l^{(j)}_k, u_j^{(k)} \right], \quad H^{2k} = \prod_{j=1}^{\mu_1} \left[ l^{(j)}_k, u_j^{(k)} \right] \times \prod_{j=\mu_1+1}^{n} \left[ l^{(j)}_k, u_j^{(k)} \right].
\]

**The Reducing of the Rectangle**

Based on literature [12], in order to improve the convergence of the algorithm, we present two pruning methods, which can be used to eliminate the region in which the global optimal solution of \((GLMP)\) problem does not exist. For any \( H^i = \{ x \in \mathbb{R}^n \mid l^i \leq x \leq u^i \} \subseteq H \), assume the objective of \((LRP^\alpha)\) is

\[
G^\alpha_0(x) = \sum_{j=1}^{\mu_0} a^0_j x_j + \beta^0_0,
\]

and the constraints are \( \sum_{j=1}^{\mu_0} a^0_j x_j \leq b^0_0 \), the current known upper bound of the optimal value \( G^\alpha_0 \) of \((GLMP)\) is denoted by \( UB \). For every \( i = 1, 2, \ldots, N + m, s = 1, 2, \ldots, n \), we must follow the following rules:

**Rule 1** Compute \( \gamma^i_s \) and \( \min \{ \alpha^i_0 l^i_s, \alpha^i_0 u^i_s \} \), if \( \gamma^i_s < \min \{ \alpha^i_0 l^i_s, \alpha^i_0 u^i_s \} \) then \( H^i \) is deleted; otherwise, if \( \alpha^i_0 > 0 \) and \( \gamma^i_s < \alpha^i_0 u^i_s \), let \( u^i_s = \min \left\{ \frac{\gamma^i_s}{\alpha^i_0}, u^i_s \right\} \); if \( \alpha^i_0 < 0 \) and \( \gamma^i_s < \alpha^i_0 l^i_s \), let \( l^i_s = \max \left\{ \frac{\gamma^i_s}{\alpha^i_0}, l^i_s \right\} \).

**Rule 2** Compute \( \delta^i_s \) and \( \min \{ \alpha^i_0 l^i_s, \alpha^i_0 u^i_s \} \), if \( \delta^i_s < \min \{ \alpha^i_0 l^i_s, \alpha^i_0 u^i_s \} \) then \( H^i \) is deleted; otherwise, if \( \alpha^i_0 > 0 \) and \( \delta^i_s < \alpha^i_0 u^i_s \), let \( u^i_s = \min \left\{ \frac{\delta^i_s}{\alpha^i_0}, u^i_s \right\} \); if \( \alpha^i_0 < 0 \) and \( \delta^i_s < \alpha^i_0 l^i_s \), let \( l^i_s = \max \left\{ \frac{\delta^i_s}{\alpha^i_0}, l^i_s \right\} \).

Where,

\[
\gamma^k_s = UB - \sum_{j=1, j \neq s}^{\mu_0} \min \{ \alpha^k_j l^k_j, \alpha^k_j u^k_j \} - \beta^k_0, \quad \delta^k_s = b^k - \sum_{j=1, j \neq s}^{\mu_0} \min \{ \alpha^k_j l^k_j, \alpha^k_j u^k_j \}.
\]

**Algorithm and its Convergence**

Next, a branch and bound reduced algorithm of problem \((GLMP)\) is described.

Suppose when the iteration proceed in step \( k \), the feasible region of the problem \((GLMP)\) is denoted by \( F \); \( H^i \) represent the divided rectangle soon, \( W \) represent the feasible set at present, the set of remained the super rectangle after pruning is denoted by \( T \). \( L \) is the lower bound of global optimal value of the problem \((GLMP)\); \( U \) said the upper bound of global optimal value of \((GLMP)\).

**Step 1 (initializing)**

Constructing n-dimensional super-rectangle \( H = [l, u] \) covering the feasible region \( F \); Solving the problem \((LRP^\alpha)\), its optimal value and optimal solution is denoted by \( G^\alpha_0(x^\alpha) \) and \( x^\alpha \) respectively. \( L = G^\alpha_0(x^\alpha) \) is a lower bound of global optimal value of the problem \((GLMP)\); if \( x^i \in F \), let \( W = W \cup \{ x^i \} \); if \( W \neq \phi \), then let \( U = \min \{ G_0(x) : x \in W \} \), and finding a current optimal solution \( x^\alpha \in \arg \min G_0 \). Set \( \varepsilon > 0 \) and let \( T = [H], k = 1 \).

**Step 2 (termination rule)**

If \( U - L \leq \varepsilon \) or \( T = \phi \), then the calculation is stopped, The global optimal solution \( G^\alpha_0(x^\alpha) \) of the output problem \((GLMP)\). Otherwise go to the next step.

**Step 3 (selection rule)**

The super rectangle \( H^i \), which corresponds to the minimum lower bound \( L \), is selected. in \( T \), i.e. \( H^i = H^i \).

**Step 4 (subdivision rule)**

Using the subdivision method of the first part of the third section, then the super rectangle \( H^i \) can be cut apart into sub-rectangles \( H^{1i} \) and \( H^{2i} \), and \( H^i \cap H^i = \phi \).
Step 5 (reducing technique)
Reducing the sub-rectangles after dividing using the reducing method of the second part of the third section, for convenience, the new sub-rectangles after reducing are also denoted by $H^*$, and $i \in \Gamma$, where $\Gamma$ is the index set of the rectangles after reducing.

Step 6 (upper bounding)
If $W = \emptyset$, let $U = +\infty$, else $U = \min\{G_0(x) : x \in W\}$. The current optimal solution is $x^* \in \arg\min G_0$.

Step 7 (pruning rule)
Let $T = T \setminus \{H^* : G^*_0(x^*) \geq U, x^* \in H^* \subseteq T\}$.

Step 8 (lower bounding)
If
$$T = \emptyset, \text{let } L = U, \text{else } L = \min\left\{G^*_k(x^k) : x^k \in H^k \subseteq T\right\}$$

Step 9 Set $k \leftarrow k + 1$, go to step 2.

Theorem 3
(a) If the algorithm terminates within finite iterations with an globally optimal solution for (GLMP) be found.

(b) If the algorithm generates an infinite sequence of iterations, then any accumulation point of the sequence $\{x^k\}$ is a global optimal solution of the (GLMP).

Proof
(a) If the algorithm is finite, assume it stops at the $k$-th iteration, $k \geq 1$. From the termination rule of Step 2, we know that $U - L \leq \varepsilon$ . Based on the upper bounding technique described in Step 6, it implies that $G_0(x^k) - L \leq \varepsilon$. From the Step 4 and Step 6, we also know the globally optimal solution is $x^*$, we known that $U = G_0(x^*) \geq G_0(x^0) \geq L$. Hence, Combined these inequalities, we obtained that $G_0(x^k) + \varepsilon \geq G_0(x^k) + \varepsilon \geq L + \varepsilon \geq G_0(x^k)$, and then the part (a) has been proven.

(b) If the algorithm is infinite, and an infinite feasible solution sequence $\{x^k\}$ is generated of the problem (GLMP) by solving the problem (LRP^k). According to the Step 6 and Step 8 of the algorithm, we have
$$L \leq G^*_0(x^k) \leq G^*_0(x^k) \leq G^*_0(x^k), k = 1, 2, \ldots$$

Because the series $\{G^*_0(x^0)\}$ is non-decreasing and bounded, and $\{G^*_0(x^0)\}$ is non-increasing and bounded, then the series $\{G^*_0(x^0)\}$ and $\{G^*_0(x^0)\}$ are both convergent. Take the limit on the both side of (5), we have
$$\lim_{k \to \infty} G^*_0(x^k) \leq \lim_{k \to \infty} G^*_0(x^k) \leq \lim_{k \to \infty} G^*_0(x^k).$$

And then $L = \lim_{k \to \infty} G^*_0(x^k)$, $U = \lim_{k \to \infty} G_0(x^k)$, the formula (6) convert into $L \leq G_k(x^k) \leq \lim_{k \to \infty} G_k(x^k) = U$.

Without loss of generality, assume the rectangle sequence $\{H^k = [l^k, u^k]\}$ satisfy $x^k \in H^k$ and $H^{k+1} \subseteq H^k$. In our algorithm, the rectangles are divided into two parts of the equal width continuously, then $\bigcap_{k \to \infty} H^k \subseteq \{x^k\}$ because of the continuity of function $G_0(x^k)$, $L = G_0(x^k) = \lim_{k \to \infty} G(x^k) = U = G_0(x^k)$. So the sequence of $\{x^k\}$ whose any accumulation point is a global optimal solution of problem (GLMP).

Numerical Experiments
We coded the algorithms in Matlab 2016a, and ran the tests in a PC with Intel(R) Core(TM) i5-4210M processor of 2.6GHz, 4 GB of RAM memory, under the Win7 operational system. The simplex method is applied to solve the LRP. Table 1 or 2 shows that our algorithm performs are efficient. In our experiments, the convergence tolerance is $10^{-8}$.

Table 1 shows that our algorithm performs more efficient than that in references [9-12] et al. Especially for Example 2, our algorithm can determine the global optimal solution with fewer
iterations, this indicates that our new relaxation technique can be better applied to some problems where the optimal solution is located in the corner of the feasible region. At here, the notations used in the head line have the following means: Iter: average numbers of iterations in the algorithm; Time: average CPU time in seconds; lam: denote the number of the associated tolerance; p,m and n denote the number of the product term,linear constraints ,variables and the multiplicative constraints, respectively.

Example 1

\[
\begin{align*}
\text{min} & \quad -4x_1^2 - 5x_2^2 + x_1x_2 + 2x_1 \\
\text{s.t} & \quad x_1 - x_2 \geq 0, \\
& \quad \frac{1}{3}x_1^2 - \frac{1}{3}x_2^2 \leq 1, \\
& \quad 0.5x_1x_2 \leq 1, \\
& \quad 0 \leq x_1 \leq 3, \\
& \quad x_2 \geq 0.
\end{align*}
\]

Example 2

\[
\begin{align*}
\text{min} & \quad (2x_1 - 2x_2 + x_3 + 2)(-2x_1 + 3x_2 + x_3 - 4) + \\
& \quad (-2x_1 + x_2 + x_3 + 2)(x_1 + x_2 - 3x_1 + 5) + \\
& \quad (-2x_1 - x_2 + 2x_3 + 7)(4x_1 - x_2 - 2x_3 - 5) \\
\text{s.t} & \quad x_1 + x_2 + x_3 \leq 10, \\
& \quad x_1 - 2x_2 + 3x_3 \leq 10, \\
& \quad -2x_1 + 2x_2 + 3x_3 \leq 10, \\
& \quad -x_1 + 2x_2 - 3x_3 \leq 10, \\
& \quad -x_1 + 2x_2 + 3x_3 \geq 6, \\
& \quad x_1 \geq 1, x_2 \geq 1, x_3 \geq 1.
\end{align*}
\]

Example 3

\[
\begin{align*}
\min & \quad \sum_{i=0}^{p} (e_i^T x + d_i)(e_i^T x + f_i) \\
\text{s.t} & \quad \sum_{i=0}^{p} (e_i^T x + d_i)(e_i^T x + f_i) \leq 0, k = 1,2,\cdots, N, \\
Ax & \leq b, 0 \leq x \leq 1, j = 1,2,\cdots, n.
\end{align*}
\]

Table 1. Results of the numerical contrast experiments 1–2.

<table>
<thead>
<tr>
<th>E</th>
<th>M</th>
<th>x*</th>
<th>f(x*)</th>
<th>Iter</th>
<th>Time</th>
<th>lam</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours</td>
<td>(2.0000, 1.0000)</td>
<td>-15.000</td>
<td>20</td>
<td>0.4837</td>
<td>1e-8</td>
<td></td>
</tr>
<tr>
<td>[9]</td>
<td>(2.0000, 1.0000)</td>
<td>-15.000</td>
<td>110</td>
<td>57.224</td>
<td>1e-8</td>
<td></td>
</tr>
<tr>
<td>[10]</td>
<td>(2.0000, 1.0000)</td>
<td>-15.000</td>
<td>1657</td>
<td>120.58</td>
<td>1e-6</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>(5.5556,1.7778, 2.6667)</td>
<td>-112.754</td>
<td>49</td>
<td>1.4302</td>
<td>1e-8</td>
<td></td>
</tr>
<tr>
<td>[11]</td>
<td>(5.5556,1.7778, 2.6667)</td>
<td>-112.754</td>
<td>57</td>
<td>-</td>
<td>1e-3</td>
<td></td>
</tr>
<tr>
<td>[12]</td>
<td>(5.5556,1.7778, 2.6667)</td>
<td>-112.754</td>
<td>5</td>
<td>1.1491</td>
<td>1e-6</td>
<td></td>
</tr>
</tbody>
</table>

Table 2. Numerical results of Example 3.

<table>
<thead>
<tr>
<th>p</th>
<th>m</th>
<th>n</th>
<th>N</th>
<th>Iter</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>33.10</td>
<td>2.3154</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>10</td>
<td>5</td>
<td>31.61</td>
<td>2.2663</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>10</td>
<td>5</td>
<td>44.35</td>
<td>2.3731</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>20</td>
<td>5</td>
<td>61.50</td>
<td>3.6550</td>
</tr>
<tr>
<td>10</td>
<td>20</td>
<td>10</td>
<td>10</td>
<td>23.15</td>
<td>1.3220</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>20</td>
<td>5</td>
<td>94.61</td>
<td>6.8459</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>20</td>
<td>10</td>
<td>45.55</td>
<td>4.4356</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>30</td>
<td>10</td>
<td>105.73</td>
<td>13.7012</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>40</td>
<td>5</td>
<td>381.75</td>
<td>39.9987</td>
</tr>
</tbody>
</table>

where the elements of $e_{0i}$, $e_{0i}$, $d_{0i}$ and $f_{0i}$ are pseudo-randomly generated in the range [-1, 1], the real elements of $e_{ki}$, $e_{ki}$, $d_{ki}$, $A$ and $b$ are pseudo-randomly generated in the range [0,1] and ,the
elements of $f_{w_k}$ is pseudo-randomly generated in the range $[-1, 0]$, $k \in \{1, 2, \ldots, N\}$. For this problem, we tested twenty different random instances and listed the computational results in Table 2.

As can be seen from Table 2, the size of $p$, $m$, $n$, and $N$ have a corresponding effect on them. With the change of $p$, $I$ and $t$ alter correspondingly when the other three variables are fixed, and there is no rule. If $p$, $m$, and $n$ are fixed, both $t$ and $I$ decrease with the increase of $n$. As soon as the $p$, $m$, and $N$ has been determined, with the increase of $N$, both $t$ and $I$ are also increasing. When $p$, $n$, and $N$ remain unchanged, $t$ and $I$ also decrease with the increase of $m$.

**Concluding Remarks**

In this study, the performance of the algorithm is different depending on the situation of the four variables. Our algorithm have no requirement for the size of $p$ if the number of $m$, $n$ and $N$ is determined. Moreover, our algorithm has a wider range of applications than the method for solving a single linear multiplicative programming problem.

**Acknowledgement**

This research was financially supported by National Natural Science Foundation of China (No. 61561001), National Science and technology leadership project (No. 2015KJ003), Postgraduate Innovation Project of North Minzu University (YCX18084) and the Project of First-rate discipline Construction in Ning xia higher Education (NXYLXK2017B09).

**References**


