

RESEARCH

Open Access



An efficient method for generalized linear multiplicative programming problem with multiplicative constraints

Yingfeng Zhao^{1,2*}  and Sanyang Liu¹

*Correspondence:
zhaoyingfeng6886@163.com
¹ School of Mathematics
and Statistics, Xidian
University, Xi'an 710071,
China
Full list of author information
is available at the end of the
article

Abstract

We present a practical branch and bound algorithm for globally solving generalized linear multiplicative programming problem with multiplicative constraints. To solve the problem, a relaxation programming problem which is equivalent to a linear programming is proposed by utilizing a new two-phase relaxation technique. In the algorithm, lower and upper bounds are simultaneously obtained by solving some linear relaxation programming problems. 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.

Keywords: Generalized linear multiplicative programming, Global optimization, Branch and bound

Mathematics Subject Classification: 90C26, 90C30

Background

Multiplicative programming refers to a class of optimization problems which contains products of real functions in objective and (or) constraint functions. In this study, we consider the following generalized linear multiplicative programming problem:

$$(\text{GLMP}) : \begin{cases} \min f_0(x) = \sum_{j=1}^{p_0} f_{0j}(x) \\ \text{s.t. } f_i(x) = \sum_{j=1}^{p_i} f_{ij}(x) \leq 0, & i = 1, 2, \dots, M, \\ f_i(x) = \sum_{j=1}^{p_i} f_{ij}(x) \geq 0, & i = M + 1, M + 2, \dots, N, \\ x \in D = \{x \in R^n \mid Ax \leq b, x \geq 0\}, \end{cases}$$

where $f_{ij}(x) = \phi_{ij}(x)\psi_{ij}(x)$, $\phi_{ij}(x) = \sum_{k=1}^n a_{ijk}x_k + b_{ij}$, $\psi_{ij}(x) = \sum_{k=1}^n c_{ijk}x_k + d_{ij}$, while the coefficients a_{ijk} , c_{ijk} and the constant terms b_{ij} , d_{ij} are all arbitrary real numbers, $i = 0, 1, 2, \dots, N$, $j = 1, 2, \dots, p_i$, $k = 1, 2, \dots, n$; $A \in R^{m \times n}$ is a matrix, $b \in R^m$ is a vector, set D is nonempty and bounded.

Generalized linear multiplicative programming (GLMP) with multiplicative objective and constraint functions is a special case of multiplicative programming. It has attracted considerable attention of researchers and practitioners for many years. This is mainly because, from the practical point of view, it possesses important application foreground

in various fields, including microeconomics (Henderson and Quandt 1961), multiple-objective decision (Benson 1979; Keeney and Raiffa 1993; Geoffrion 1967), plant layout design (Quesada and Grossmann 1996), data mining\pattern recognition (Bennett and Mangasarian 1994), marketing and service planning (Samadi et al. 2013), robust optimization (Mulvey et al. 1995), and so on. And from the algorithmic design point of view, a product of two affine functions, as noted in Avriel et al. (2010), need not be convex (even not be quasi-convex), and hence problem (GLMP) is a global optimization problem which may have multiple non-global local solutions, global solution methods for problem (GLMP) have great difficulties and challenges. Due to the facts above, design efficient solution methods for globally solving the (GLMP) has important theoretical and the practical significance.

In the past few decades, many solution methods have been devised for solving the problem (GLMP). These methods are mainly classified as parameter-based methods (Konno et al. 1994; Thoai 1991), outer-approximation methods (Gao et al. 2006; Kuno et al. 1993), outcome-space cutting plane methods (Benson and Boger 2000), branch-and-bound methods (Ryoo and Sahinidis 2003; Shen and Jiao 2006; Konno and Fukai-shi 2000; Kuno 2001) and various heuristic methods (Benson and Boger 1997; Liu et al. 1999; Fan et al. 2016). Recently, Wang proposes a global optimization algorithm for a kind of generalized linear multiplicative programming by using simplicial partition techniques (Wang et al. 2012), but his method is only valid for problems in which the constraint functions are all linear. Jiao and Liu (2015) present an effective algorithm for solving the generalized linear multiplicative problem with generalized polynomial constraints by converting it into an equivalent generalized geometric programming problem, the problem they considered is more general but only valid under the assumption $\phi_{ij}(x) > 0, \psi_{ij}(x) > 0, \forall x \in X$. There are many other solution methods not mentioned for (GLMP) and its special case, nevertheless, most of these methods are either developed for special circumstances or can only obtain a local solution of problem (GLMP).

In this paper, we put forward a fast global optimization algorithm for generalized linear multiplicative programming problem (GLMP). Our research can be divided into three steps. First, a well performed linear relaxation programming problem for the (GLMP) is established by using a new two-phase relaxation technique. Second, two key operations for developing a branch and bound algorithm for the (GLMP) are described. Finally, global convergence property is proved and some numerical experiments are executed to illustrate the feasibility and robustness of the proposed algorithm. Compared with some existing methods, the new two-phase relaxation technique we used in the algorithm has a very good approximation effect, and it doesn't require the condition $\phi_{ij}(x) > 0, \psi_{ij}(x) > 0, \forall x \in X$. Further more, relative to the algorithm in Jiao (2009), Quesada and Grossmann (1996), the proposed algorithm can be applied to a more general case of linear multiplicative programming problem.

The reminder of this article is arranged as follows. Section "[Two-phase relaxation technique](#)" explains how the two-phase relaxation method is realized, section "[Algorithm and its convergence](#)" introduces the branch and bound operation for deriving the presented algorithm. The algorithm statement as well as the convergence property are described in section "[Numerical experiments](#)". In section "[Concluding remarks](#)", the results of some

numerical experiments appeared in recent literatures are listed and some concluding remarks are reported in the last section.

Two-phase relaxation technique

As is known to all, construct a well performed relaxation problem can bring great convenience for designing branch and bound algorithm of global optimization problems. In this section, equivalent transformation technique and a new two-phase relaxation skill will be used to establish a linear relaxation programming problem for underestimating the optimal value of problem (GLMP).

First, we compute the initial variable bounds by solving the following linear programming problems:

$$x_i^l = \min_{x \in D} x_i, \quad x_i^u = \max_{x \in D} x_i, \quad i = 1, 2, \dots, n,$$

then an initial rectangle $X^0 = \{x \in R^n \mid x_i^l \leq x_i \leq x_i^u, \quad i = 1, 2, \dots, n\}$ will be obtained.

To construct the first-phase relaxation programming problem of the (GLMP) over sub-rectangle $X \subset X^0$, we further solve some linear programming problems as follows:

$$\begin{aligned} l_{ij} &= \min_{x \in D \cap X} \phi_{ij}(x), & u_{ij} &= \max_{x \in D \cap X} \phi_{ij}(x), \\ L_{ij} &= \min_{x \in D \cap X} \psi_{ij}(x), & U_{ij} &= \max_{x \in D \cap X} \psi_{ij}(x). \end{aligned} \tag{1}$$

Upon criteria (1), it is clear that

$$(\phi_{ij}(x) - l_{ij})(\psi_{ij}(x) - L_{ij}) \geq 0, \quad (\phi_{ij}(x) - u_{ij})(\psi_{ij}(x) - U_{ij}) \geq 0, \tag{2}$$

and

$$(\phi_{ij}(x) - l_{ij})(\psi_{ij}(x) - U_{ij}) \leq 0, \quad (\phi_{ij}(x) - u_{ij})(\psi_{ij}(x) - L_{ij}) \leq 0, \tag{3}$$

by taking (2) and (3) together, we have

$$\phi_{ij}(x)\psi_{ij}(x) \geq \max \{u_{ij}\psi_{ij}(x) + U_{ij}\phi_{ij}(x) - U_{ij}u_{ij}, l_{ij}\psi_{ij}(x) + L_{ij}\phi_{ij}(x) - L_{ij}l_{ij}\}, \tag{4}$$

and

$$\phi_{ij}(x)\psi_{ij}(x) \leq \min \{u_{ij}\psi_{ij}(x) + L_{ij}\phi_{ij}(x) - u_{ij}L_{ij}, l_{ij}\psi_{ij}(x) + U_{ij}\phi_{ij}(x) - U_{ij}l_{ij}\}. \tag{5}$$

For each $i = 0, 1, 2, \dots, N$, by denoting

$$\underline{g}_{ij}^1(x) \triangleq u_{ij}\psi_{ij}(x) + U_{ij}\phi_{ij}(x) - U_{ij}u_{ij}, \quad \underline{g}_{ij}^2(x) \triangleq l_{ij}\psi_{ij}(x) + L_{ij}\phi_{ij}(x) - L_{ij}l_{ij},$$

and

$$\bar{g}_{ij}^1(x) \triangleq u_{ij}\psi_{ij}(x) + L_{ij}\phi_{ij}(x) - u_{ij}L_{ij}, \quad \bar{g}_{ij}^2(x) \triangleq l_{ij}\psi_{ij}(x) + U_{ij}\phi_{ij}(x) - U_{ij}l_{ij},$$

conclusion (4) and (5) can be expressed as

$$\phi_{ij}(x)\psi_{ij}(x) \leq \min \{\bar{g}_{ij}^1(x), \bar{g}_{ij}^2(x)\} \triangleq \bar{g}_{ij}(x), \tag{6}$$

and

$$\phi_{ij}(x)\psi_{ij}(x) \geq \max \left\{ \underline{g}_{ij}^1(x), \underline{g}_{ij}^2(x) \right\} \triangleq \underline{g}_{ij}(x), \tag{7}$$

respectively. Then we can obtain a lower bound function $\underline{g}_i(x)$ and upper bound function $\bar{g}_i(x)$ for $f_i(x)$, which satisfy $\underline{g}_i(x) \leq f_i(x) \leq \bar{g}_i(x), i = 0, 1, 2, \dots, N$, where

$$\underline{g}_i(x) = \sum_{j=1}^{p_i} \underline{g}_{ij}(x), \quad \bar{g}_i(x) = \sum_{j=1}^{p_i} \bar{g}_{ij}(x), \quad i = 0, 1, 2, \dots, N. \tag{8}$$

So far, based on the above discussion, we can get the first-phase relaxation programming problem for the (GLMP) which we formulated as follows:

$$(RMP0) : \begin{cases} \min \underline{g}_0(x) = \sum_{j=1}^{p_0} \underline{g}_{0j}(x) \\ \text{s.t. } \underline{g}_i(x) = \sum_{j=1}^{p_i} \underline{g}_{ij}(x) \leq 0, \quad i = 1, 2, \dots, M, \\ \bar{g}_i(x) = \sum_{j=1}^{p_i} \bar{g}_{ij}(x) \geq 0, \quad i = M + 1, M + 2, \dots, N, \\ x \in D \cap X = \{x \in X \mid Ax \leq b, x \geq 0\}, \end{cases}$$

To get the second-phase linear relaxation programming problem, we will once again relax each nonlinear function appeared in problem (RMP0) according the following conclusion:

$$\begin{aligned} \underline{g}_i(x) &= \sum_{j=1}^{p_i} \max \left\{ \underline{g}_{ij}^1(x), \underline{g}_{ij}^2(x) \right\} \\ &\geq \max \left\{ \sum_{j=1}^{p_i} \underline{g}_{ij}^1(x), \sum_{j=1}^{p_i} \underline{g}_{ij}^2(x) \right\} \triangleq g_i(x), \quad i = 0, 1, 2, \dots, M, \end{aligned} \tag{9}$$

and

$$\begin{aligned} \bar{g}_i(x) &= \sum_{j=1}^{p_i} \min \left\{ \bar{g}_{ij}^1(x), \bar{g}_{ij}^2(x) \right\} \\ &\leq \min \left\{ \sum_{j=1}^{p_i} \bar{g}_{ij}^1(x), \sum_{j=1}^{p_i} \bar{g}_{ij}^2(x) \right\} \triangleq g_i(x), \quad i = M + 1, M + 2, \dots, N. \end{aligned} \tag{10}$$

With conclusion (9) and (10), the second-phase relaxation programming problem (RMP1) of the (GLMP) can be expressed as follows:

$$(RMP1) : \begin{cases} \min g_0(x) = \max \left\{ \sum_{j=1}^{p_0} g_{0j}^1(x), \sum_{j=1}^{p_0} g_{0j}^2(x) \right\} \\ \text{s.t. } g_i(x) = \max \left\{ \sum_{j=1}^{p_i} g_{ij}^1(x), \sum_{j=1}^{p_i} g_{ij}^2(x) \right\} \leq 0, & i = 1, 2, \dots, M, \\ g_i(x) = \min \left\{ \sum_{j=1}^{p_i} \bar{g}_{ij}^1(x), \sum_{j=1}^{p_i} \bar{g}_{ij}^2(x) \right\} \geq 0, & i = M + 1, M + 2, \dots, N, \\ x \in D \cap X = \{x \in X \mid Ax \leq b, \quad x \geq 0\}, \end{cases}$$

which is proved equivalent to the following linear programming problem:

$$(ERMP) : \begin{cases} \min t \\ \text{s.t. } \sum_{j=1}^{p_0} g_{0j}^1(x) - t \leq 0, \\ \sum_{j=1}^{p_0} g_{0j}^2(x) - t \leq 0, \\ \sum_{j=1}^{p_i} g_{ij}^1(x) \leq 0, & i = 1, 2, \dots, M, \\ \sum_{j=1}^{p_i} g_{ij}^2(x) \leq 0, & i = 1, 2, \dots, M, \\ \sum_{j=1}^{p_i} \bar{g}_{ij}^1(x) \geq 0, & i = M + 1, M + 2, \dots, N, \\ \sum_{j=1}^{p_i} \bar{g}_{ij}^2(x) \geq 0, & i = M + 1, M + 2, \dots, N, \\ x \in D \cap X = \{x \in X \mid Ax \leq b, \quad x \geq 0\}. \end{cases}$$

Theorem 1 *If $(x^*, t) \in R^{n+1}$ is a global optimal solution for the (ERMP), then $x^* \in R^n$ is a global optimal solution for the (RMP1). Conversely, if $x^* \in R^n$ is a global optimal solution for the (RMP1), then $(x^*, t) \in R^{n+1}$ is a global optimal solution for the (ERMP), where $t = g_0(x^*)$.*

Proof The proof of this theorem can be easily followed according to the definition of problems (RMP1) and (ERMP), therefore, it is omitted here. □

Theorem 2 (1) *For any $x \in X$, we have*

$$g_i(x) \leq f_i(x), \quad i = 0, 1, 2, \dots, M,$$

and

$$g_i(x) \geq f_i(x), \quad i = M + 1, M + 2, \dots, N.$$

(2) $|g_i(x) - f_i(x)| \rightarrow 0$, as $\|U_i - L_i\| \rightarrow 0$, $\|u_i - l_i\| \rightarrow 0$, where $U_i = (U_{i1}, U_{i2}, \dots, U_{ip_i})$, $L_i = (L_{i1}, L_{i2}, \dots, L_{ip_i})$ and $u_i = (u_{i1}, u_{i2}, \dots, u_{ip_i})$, $l_i = (l_{i1}, l_{i2}, \dots, l_{ip_i})$.

Proof (1) can be easily verified from conclusion (4), (5), (9) and (10), thus the detailed proof is omitted here.

For (2), according to the Cauchy–Schwarz inequality, we know that for $i = 0, 1, \dots, M$,

$$\begin{aligned}
 & |g_i(x) - f_i(x)| \\
 &= \left| \max \left\{ \sum_{j=1}^{p_i} g_{ij}^1(x), \sum_{j=1}^{p_i} g_{ij}^2(x) \right\} - \sum_{j=1}^{p_i} \phi_{ij}(x)\psi_{ij}(x) \right| \\
 &= \left| \max \left\{ \sum_{j=1}^{p_i} g_{ij}^1(x) - \sum_{j=1}^{p_i} \phi_{ij}(x)\psi_{ij}(x), \sum_{j=1}^{p_i} g_{ij}^2(x) - \sum_{j=1}^{p_i} \phi_{ij}(x)\psi_{ij}(x) \right\} \right| \\
 &= \left| \max \left\{ \sum_{j=1}^{p_i} (l_{ij}\psi_{ij}(x) + L_{ij}\phi_{ij}(x) - l_{ij}L_{ij} - \phi_{ij}(x)\psi_{ij}(x)), \right. \right. \\
 &\quad \left. \left. \sum_{j=1}^{p_i} (u_{ij}\psi_{ij}(x) + U_{ij}\phi_{ij}(x) - U_{ij}u_{ij} - \phi_{ij}(x)\psi_{ij}(x)) \right\} \right| \\
 &= \left| \max \left\{ \sum_{j=1}^{p_i} (\psi_{ij}(x) - L_{ij})(\phi_{ij}(x) - l_{ij}), \sum_{j=1}^{p_i} (\psi_{ij}(x) - U_{ij})(\phi_{ij}(x) - u_{ij}) \right\} \right| \\
 &\leq \max \{ \|(U - L)\| \|(u - l)\| \},
 \end{aligned}$$

for the case $i = M + 1, M + 2, \dots, N$, it can be proved with the similar method, so omitted here, and thus the Proof of Theorem 2 is completed. \square

Remark 1 From Theorems 1 and 2, we only need to solve problem (ERMP) instead of solving the (RMP1) to obtain the lower and upper bounds of the optimal value in problem (GLMP).

Remark 2 Based on the continuity of linear function, $\|U_i - L_i\| \rightarrow 0$ and $\|u_i - l_i\| \rightarrow 0$ will hold when the diameter of X approximate to zero, this indicated that we can perform the branching operation in variable space X with the convergence property is guaranteed.

Remark 3 Theorem 2 ensures that problem (ERMP) can infinitely approximate the problem (GLMP), as $\|X\| \rightarrow 0$, this will guarantee the global convergence of the proposed algorithm.

Algorithm and its convergence

In this section, we will describe two key operation for designing an efficient branch and bound algorithm for problem (GLMP), that is, branching and bounding. Then the algorithm steps will be summarized with proof process of global convergence property followed.

Branching and bounding

The branching operation iteratively subdivides the rectangle X into subregions according to an exhaustive partition rule, such that any infinite iterative sequence of partition sets shrinks to a singleton. For this, we shall adopt an standard range bisection approach, which is adequate to insure global convergence of the proposed algorithm. Detailed process is described as follows.

For any region $X = [x^l, x^u] \subset X^0$, let $r \in \operatorname{argmin}\{x_i^u - x_i^l \mid i = 1, 2, \dots, n\}$ and $mid = (x_r^l + x_r^u)/2$, then the current region X can be divided into the following two sub-regions:

$$\tilde{X} = \left\{ x \in R^n \mid x_i^l \leq x_i \leq x_i^u, i \neq r, x_r^l \leq x_r \leq mid \right\},$$

and

$$\bar{X} = \left\{ x \in R^n \mid x_i^l \leq x_i \leq x_i^u, i \neq r, mid \leq x_r \leq x_r^u \right\}.$$

For each partition subset X generated by the above branching operation, the bounding operation is mainly concentrate on estimating a lower bound $LB(X)$ and a upper bound $UB(X)$ for the optimal value of problem (GLMP). This operation is realized by solving the linear relaxation programming problem (ERMP) over all partition sets in the k_{th} iteration, and the one with the smallest optimal value will provide the lower bound for optimal value of problem (GLMP) over the initial region X^0 . Moreover, since any feasible solution of the relaxation programming problem will also be feasible to the (GLMP), so we can evaluate the initial objective value and make the one with smallest value as a new upper bound if possible.

Algorithm and its convergence

Based on the former discussion, the algorithm steps can be summarized as follows:

Step 0 (Initialization) Choose convergence tolerance $\epsilon = 1 \times 10^{-8}$, set iteration counter $k := 0$ and the initial partition set as $\Omega_0 = X^0$. Solve the initial linear relaxation problem (ERMP) over region X^0 , if the (ERMP) is not feasible then there is no feasible solution for the initial problem. Otherwise, denote the optimal value and solution as f_{bar} and x_{opt}^0 , respectively. Then we can obtain the initial upper and lower bound of the optimal value for problem (GLMP), that is, $UB := f_0(x_{opt}^0)$, and $LB := f_{bar}$. And then, if $UB - LB < \epsilon$, the algorithm can stop, and x_{opt}^0 is the optimal solution of the (GLMP), otherwise proceed to step 1.

Step 1 (Branching) Partition X^k into two new sub-rectangles according to the partition rule described in section “**Branching and bounding**”. Deleting X^k and add the new nodes into the active nodes set \tilde{X}^k , still denote the set of new partitioned sets as \tilde{X}^k .

Step 2 (Bounding) For each subregion still of interest $X^{k\mu} \subseteq X^0, \mu = 1, 2$, obtain the optimal solution and value for problem (RMFP) by solving the relaxation linear programming problem over $X^{k\mu}$, if $LB(X^{k,\mu}) > UB$, delete $X^{k\mu}$ from \tilde{X}^k . Otherwise, we can update the lower and upper bounds: $LB = \min\{LB(X^{k,\mu}) \mid \mu = 1, 2\}$ and $UB = \min\{UB, f(x^{k,\mu}) \mid \mu = 1, 2\}$.

Step 3 (Termination) If $UB - LB \leq \epsilon$, the algorithm can be stopped, UB is the global optimal value for (GLMP). Otherwise, set $k := k + 1$, and select the node with the smallest optimal value as the current active node, and return to Step 1.

Theorem 3 *The proposed algorithm either terminates within finite iterations with an optimal solution for (GLMP) be found, or generates an infinite sequence of iterations such that along any infinite branches of the branch-and-bound tree, any accumulation point of the sequence $\{x^k\}$ will be the global optimal solution of the (GLMP).*

Proof (1) If the proposed algorithm is finite, assume it stops at the k th iteration, $k \geq 0$. From the termination criteria, we know that

$$UB - LB \leq \epsilon.$$

Based on the upper bounding technique described in Step 3, it implies that

$$f(x^k) - LB \leq \epsilon.$$

Let v_{opt} be the optimal value of problem (GLMP), then by section “Branching and bounding” and Step 3 above, we know that

$$UB = f(x^k) \geq v_{opt} \geq LB.$$

Hence, taken together, it implies that

$$v_{opt} + \epsilon \geq LB + \epsilon \geq f(x^k) \geq v_{opt},$$

and thus the proof of part (1) is completed.

(2) If the algorithm doesn't terminate within finite iterations and generates an infinite feasible solution sequence $\{x^k\}$ for the (GLMP) via solving the (RMP1). According to the structure of the proposed algorithm, we have

$$LB_k \leq \min_{x \in X} f_0(x), \tag{11}$$

assume that:

$$X^k \in \arg \min_{X \in \Omega_k} LB(X), \quad x^k = x(X^k) \in X^k \subseteq X^0. \tag{12}$$

Horst (1998) has proved that LB_k is non-decrease and bounded above by $\min_{x \in X} f_0(x)$, thus the existence of the limit $LB := \lim_{k \rightarrow \infty} LB_k \leq \min_{x \in X} f_0(x)$ can be guaranteed. Further more, since x^k is a sequence on a compact set, it must have a convergent subsequence. For any accumulation point \hat{x} of $\{x^k\}$, there exists a subsequence of $\{x^k\}$ which, without loss of generality, we might still denote as $\{x^k\}$ satisfied $\lim_{k \rightarrow \infty} x^k = \hat{x}$. With similar method in Tuy (1991), we can easily follow that the subdivision of partition sets in step 1 is exhaustive on X^0 , and the selection of elements to be partitioned is bound improving, thus there exists a decreasing subsequence $X^r \subset X^k$ where $X^r \in \Omega_r$ with $x^r \in X^r$, $LB_r = LB(X^r) = g_0(x^r)$, $\lim_{r \rightarrow \infty} x^r = \hat{x}$. Based on the construction process of the relaxation problem, we know that the linear relaxation functions $g_i(x) (i = 0, 1, \dots, N)$ used in problem (RMP1)(and thus for (ERMP)) are strongly consistent on X^0 , hence it follows that $\lim_{k \rightarrow \infty} LB_k = LB = g_0(\hat{x})$. Since \hat{x} is feasible to (GLMP) and combining with (11) we can deduce that \hat{x} is a global solution for the (GLMP). □

Numerical experiments

To verify the performance of the proposed algorithm, we solve some test problems in recent literatures (Thoai 1991; Wang et al. 2012; Jiao and Liu 2015; Wang and Liang 2005; Gao et al. 2010; Chen and Jiao 2009; Shen et al. 2008; Shen and Jiao 2006; Jiao 2009) and construct a

problem to illustrate the nature that (GLMP) may have multiple local optimal solutions (see Fig. 1), computational results are given in Table 1, where the following notations have been used in row headers: Exa.: the serial number of experiments; Ref.: reference which we contrast with; Opt.Val.: optimal value; Opt.Sol.: optimal solution; Iter: numbers of iterations; Time: CPU time in seconds; Pre.: precision we used in the algorithm. We used the TPRM to represent the two-phase relaxation method given in this paper.

We coded the algorithms in Matlab 2014a, and ran the tests in a micro computer with Intel(R) Xeon(R) processor of 2.4 GHz, 4 GB of RAM memory, under the Win10 operational system. We used linprog solver to solve all linear programming problems.

Table 1 shows that our algorithm performs more efficient than that in references Ryoo and Sahinidis (2003), Shen and Jiao (2006), Thoai (1991), Tuy (1991), Wang et al. (2012) and Wang and Liang (2005). Especially for Examples 1, 2, 5, 6, 8 and 10, our algorithm only need one iteration to determine the global optimal solutions, this indicates that our new relaxation technique is so efficient that the global optimal solution can be founded in the initialization step. Further more, we constructed an example (Example 11 and Fig. 1) with multiple local optimum to test our algorithm.

Example 1 (Refs. Wang and Liang 2005; Jiao 2009).

$$\begin{cases} \min x_1^2 + x_2^2 \\ \text{s.t. } 0.3x_1x_2 \geq 1, \\ \quad 2 \leq x_1 \leq 5, \\ \quad 1 \leq x_2 \leq 3, \end{cases}$$

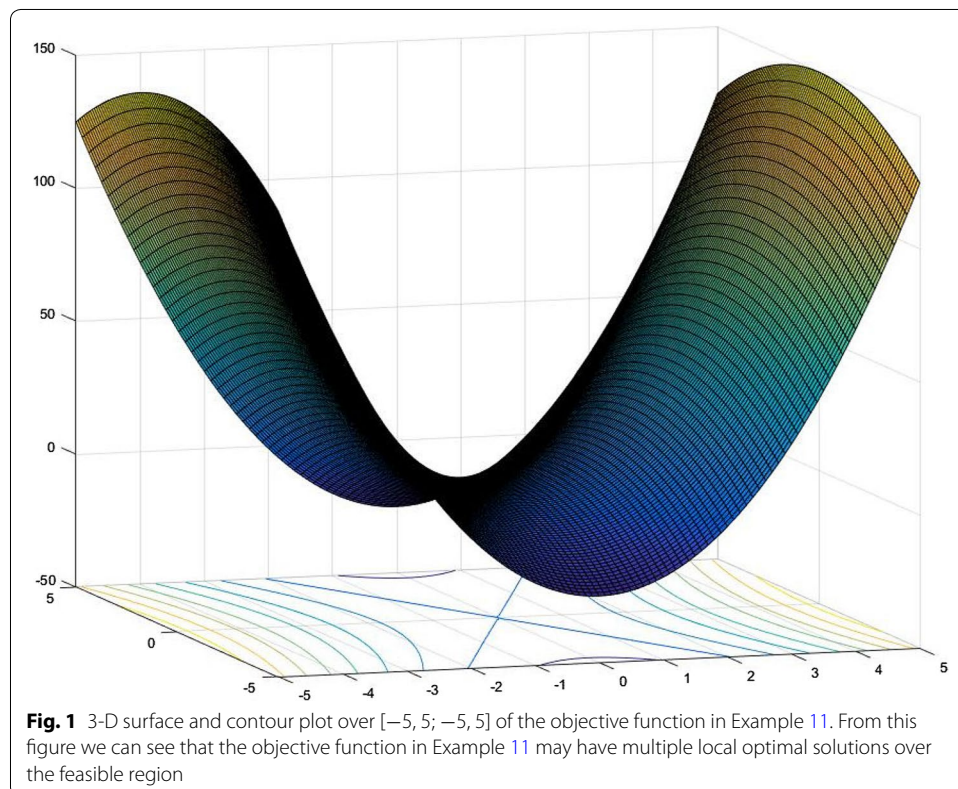


Table 1 Results of the numerical contrast experiments 1–11

Exa.	Ref.	Opt. val.	Opt. sol.	Iter	Time	Pre.
1	Wang and Liang (2005)	6.7780	(2.00003, 1.66665)	44	0.18	10 ⁻⁴
	Jiao (2009)	6.77778	(2.0, 1.666667)	58	<1	10 ⁻⁸
	TPRM	6.77778	(2.0000, 1.6667)	1	0.027	10 ⁻⁸
2	Jiao (2009)	-4.0	(2.0, 1.0, 3.0)	43	-	10 ⁻⁸
	TPRM	-4.0	(2.0000, 1.0000, 3.0000)	1	0.054	10 ⁻⁸
3	Gao et al. (2010)	10.0042	(2.0003, 7.9999)	27	10.83	10 ⁻³
	Chen and Jiao (2009)	10.00009	(1.999998, 7.9999988)	41	0.02	10 ⁻⁵
	TPRM	10.0	(2.0000, 8.0000)	2	0.0407	10 ⁻⁸
4	Gao et al. (2010)	0.0000	(0.0002, 0.0001, 0, 0, 0, 0, 0, 0, 0, 0)	36	16.03	10 ⁻³
	TPRM	0.0000	(0.00, 2.3453, 0.0000, 6.4121, 1.9434, 0.00, 2.4858, 8.4448, 6.9770, 5.8001, 5.1340)	13	1.2758	10 ⁻⁸
5	Thoai (1991)	0.8902	(1.314792, 1.39555, 0, 0.42329)	6	0.1880	10 ⁻⁶
	TPRM	0.89019	(1.3148, 0.1396, 0.0000, 0.4233)	1	0.00355	10 ⁻⁸
6	Shen et al. (2008)	11.475	(0.61824, 0.0000)	29	0.01	10 ⁻³
	TPRM	4.0000	(0.0000, 0.0000)	1	0.022	10 ⁻⁸
7	Chen and Jiao (2009)	-15.000	(2.0, 1)	1657	120.58	10 ⁻⁶
	TPRM	-15.0000	(2.0000, 1.0000)	110	57.224	10 ⁻⁸
8	Shen and Jiao (2006)	0.0000	(2.00, 1.00)	24	-	10 ⁻³
	Jiao and Liu (2015)	0.00000003	(2.0000061, 1.0)	16	0.018	10 ⁻⁸
	TPRM	0.0000	(2.0000, 1.0000)	1	0.05406	10 ⁻⁸
9	Shen and Jiao (2006)	1.1771	(1.17709, 2.1772)	434	1	10 ⁻³
	Jiao and Liu (2015)	1.17708	(1.17709, 2.1772)	189	0.226	10 ⁻⁶
	TPRM	1.1770	(1.177088, 2.17718)	3	0.66936	10 ⁻⁸
10	Jiao and Liu (2015)	3.0000	(0.0000, 4.0000)	25	0.750	10 ⁻⁸
	TPRM	3.0000	(0.0000, 4.0000)	1	0.02456	10 ⁻⁸
11	TPRM	-25.0000	(0.0000, -5.0000)	47	22.64563	10 ⁻⁸

Example 2 (Refs. Jiao 2009).

$$\left\{ \begin{array}{l} \min x_1^2 + x_2^2 - x_3^2 \\ \text{s.t. } 0.3x_1x_2 + 0.3x_2x_3 + 0.6x_1x_3 \geq 4, \\ 2 \leq x_1 \leq 5, \\ 1 \leq x_2 \leq 3, \\ 1 \leq x_3 \leq 3, \end{array} \right.$$

Example 3 (Refs. Gao et al. 2010; Chen and Jiao 2009).

$$\left\{ \begin{array}{l} \min (x_1 + x_2)(x_1 - x_2 + 7) \\ \text{s.t. } 2x_1 + x_2 \leq 14, \\ x_1 + x_2 \leq 10, \\ -4x_1 + x_2 \leq 0, \\ 2x_1 + x_2 \geq 6, \\ x_1 + x_2 \geq 6, \\ x_1 \leq 5, \\ x_1 + x_2 \geq 0, \\ x_1 - x_2 + 7 \geq 0. \end{array} \right.$$

Example 4 (Refs. Gao et al. 2010; Chen and Jiao 2009).

$$\begin{cases} \min (c_1^T x + d_1)(c_2^T x + d_2) \\ \text{s.t. } Ax \leq b, \end{cases}$$

where

$$b = (81, 72, 72, 9, 9, 9, 8, 8)^T, \quad d_1 = 0, d_2 = 0,$$

$$c_1 = \left(1, 0, \frac{1}{9}, 0, 0, 0, 0, 0, 0, 0\right)^T, \quad c_2 = \left(0, 1, \frac{1}{9}, 0, 0, 0, 0, 0, 0, 0\right)^T.$$

$$A = \begin{pmatrix} 9 & 9 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 1 & 8 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 8 & 8 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 7 & 1 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 7 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 7 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

Example 5 (Refs. Wang et al. 2012; Thoai 1991).

$$\begin{cases} \min (0.813396x_1 + 0.67440x_2 + 0.305038x_3 + 0.129742x_4 + 0.217796) \\ \quad \times (0.224508x_1 + 0.063458x_2 + 0.932230x_3 + 0.528736x_4 + 0.091947) \\ \text{s.t. } 0.488509x_1 + 0.063565x_2 + 0.945686x_3 + 0.210704x_4 \leq 3.562809, \\ \quad -0.324014x_1 - 0.501754x_2 - 0.719204x_3 + 0.099562x_4 \leq -0.052215, \\ \quad 0.445225x_1 - 0.346896x_2 + 0.637939x_3 - 0.257623x_4 \leq 0.427920, \\ \quad -0.202821x_1 + 0.647361x_2 + 0.920135x_3 - 0.983091x_4 \leq 0.840950, \\ \quad -0.886420x_1 - 0.802444x_2 - 0.305441x_3 - 0.180123x_4 \leq -1.353686, \\ \quad -0.515399x_1 - 0.424820x_2 + 0.897498x_3 + 0.187268x_4 \leq 2.137251, \\ \quad -0.591515x_1 + 0.060581x_2 - 0.427365x_3 + 0.579388x_4 \leq -0.290987, \\ \quad 0.423524x_1 + 0.940496x_2 - 0.437944x_3 - 0.742941x_4 \leq 0.373620, \\ \quad x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0. \end{cases}$$

Example 6 (Refs. Chen and Jiao 2009).

$$\begin{cases} \min (6x_1 + x_2 + 1)(x_1 + 2x_2 + 1) + (-x_1 + 3)(x_1 + x_2 + 1) \\ \text{s.t. } -2x_1 + x_2 \leq 0, \\ \quad x_1 + x_2 \leq 8, \\ \quad 0 \leq x_1 \leq 2.5, \\ \quad x_2 \geq 0. \end{cases}$$

Example 7 (Refs. Shen et al. 2008).

$$\begin{cases} \min -4x_1^2 - 5x_2^2 + x_1x_2 + 2x_1 \\ \text{s.t. } x_1 - x_2 \geq 0, \\ \quad \frac{1}{3}x_1^2 - \frac{1}{3}x_2^2 \leq 1, \\ \quad \frac{1}{2}x_1x_2 \leq 1, \\ \quad 0 \leq x_1 \leq 3, \\ \quad x_2 \geq 0. \end{cases}$$

Example 8 (Refs. Shen and Jiao 2006; Jiao and Liu 2015).

$$\begin{cases} \min & x_1x_2 - 2x_1 + x_2 + 1 \\ \text{s.t.} & 8x_2^2 - 6x_1 - 16x_2 \leq -11, \\ & -x_2^2 + 3x_1 + 2x_2 \leq 7, \\ & 1 \leq x_1 \leq 2.5, \\ & 1 \leq x_2 \leq 2.225. \end{cases}$$

Example 9 (Refs. Shen and Jiao 2006; Jiao and Liu 2015).

$$\begin{cases} \min & x_1 \\ \text{s.t.} & \frac{1}{4}x_1 + \frac{1}{2}x_2 - \frac{1}{16}x_1^2 - \frac{1}{16}x_2^2 \leq 1, \\ & \frac{1}{14}x_1^2 + \frac{1}{14}x_2^2 - \frac{3}{7}x_1 - \frac{3}{7}x_2 \leq -1, \\ & 1 \leq x_1 \leq 5.5, \\ & 1 \leq x_2 \leq 5.5. \end{cases}$$

Example 10 (Refs. Jiao and Liu 2015).

$$\begin{cases} \min & x_1 + (2x_1 - 3x_2 + 13)(x_1 + x_2 - 1) \\ \text{s.t.} & -x_1 + 2x_2 \leq 8, \\ & -x_2 \leq -3, \\ & x_1 + 2x_2 \leq 12, \\ & x_1 - 2x_2 \leq -5, \\ & x_1 \geq 0, \\ & x_2 \geq 0. \end{cases}$$

Example 11 Figure 1.

$$\begin{cases} \min & 6x_1^2 - x_2^2 \\ \text{s.t.} & -2x_1 + x_2 \leq 0, \\ & x_1 + x_2 \leq 8, \\ & \frac{1}{64}x_1^2 - \frac{1}{64}x_1x_2 \leq 1, \\ & \frac{1}{4}x_1x_2 - \frac{1}{8}x_2^2 \leq 1, \\ & -5 \leq x_1 \leq 5, \\ & -5 \leq x_2 \leq 5. \end{cases}$$

Example 12

$$\begin{cases} \min & f(x) = \sum_{i=1}^p (a_{0i}^T x + d_{0i})(c_{0i}^T x + e_{0i}) \\ \text{s.t.} & \sum_{i=1}^p (a_{1i}^T x + b_{1i})(c_{1i}^T x + d_{1i}) \leq 0, \\ & \sum_{i=1}^p (a_{2i}^T x + b_{2i})(c_{2i}^T x + d_{2i}) \geq 0, \\ & x \in D = \{x \in R^n \mid Ax \leq b\}. \end{cases}$$

Table 2 Numerical results of Example 12

<i>p</i>	<i>m</i>	<i>n</i>	Iter	Time
5	10	10	8.8	5.0245
5	10	20	10.3	7.2315
5	10	30	28.0	45.1347
10	10	20	37.4	65.0159
10	20	40	49.2	75.1032

where the real numbers a_{ij}, c_{ij}, d_{ij} and e_{ij} are randomly generated in the range $[-1, 1]$, the real elements of A and b are randomly generated in the range $[0, 1]$. For this problem, we tested twenty different random instances and listed the computational results in Table 2, where 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; m and n denote the number of linear constraints and variables, respectively.

Concluding remarks

In this study, a new global optimization algorithm is presented for solving generalized linear multiplicative programming problem with multiplicative constraints. This method has three main features. First, the relaxation problem performs well in approximation effect. Second, to obtain the lower and upper bounds of the optimal value, we only need to solve some linear programming problems. Finally, the problem we investigated is more general than those in many other literatures and results of numerical contrast experiments show that our method performs better than those methods.

Authors' contributions

Both authors contributed equally to the manuscript. Both authors read and approved the final manuscript.

Author details

¹ School of Mathematics and Statistics, Xidian University, Xi'an 710071, China. ² School of Mathematical Science, Henan Institute of Science and Technology, Xinxiang 453003, China.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China (61373174); the Science and Technology Key Project of Education Department of Henan Province (14A110024), (15A110023) and (16A110030); the natural science foundation of Henan Province (152300410097); the Major Scientific Research Projects of Henan Institute of Science and Technology (2015ZD07).

Competing interests

The authors declare that they have no competing interests.

Received: 23 April 2016 Accepted: 2 August 2016

Published online: 09 August 2016

References

Avriel M, Diewert WE, Schaible S, Zhang J (2010) Generalized concavity. Plenum Press, New York
 Bennett KF, Mangasarian OL (1994) Bilinear separation of two sets in n-space. *Comput Optim Appl* 2:207–227
 Benson HP (1979) Vector maximization with two objective functions. *J Optim Theory Appl* 28:253–257
 Benson HP, Boger GM (1997) Multiplicative programming problems: analysis and efficient point search heuristic. *J Optim Theory Appl* 94(2):487–510
 Benson HP, Boger GM (2000) Outcome-space cutting-plane algorithm for linear multiplicative programming. *J Optim Theory Appl* 104(2):301–22
 Chen YQ, Jiao HW (2009) A nonisolated optimal solution of general linear multiplicative programming problems. *Comput Oper Res* 36:2573–2579

- Fan QW, Wu W, Zurada JM (2016) Convergence of batch gradient learning with smoothing regularization and adaptive momentum for neural networks. *SpringerPlus* 5(295):1–17
- Gao YL, Xu CY, Yang YJ (2006) An outcome-space finite algorithm for solving linear multiplicative programming. *Appl Math Comput* 179:494–505
- Gao YL, Wu GR, Ma WM (2010) A new global optimization approach for convex multiplicative programming. *Appl Math Comput* 216:1206–1218
- Geoffrion M (1967) Solving bicriterion mathematical programs. *Oper Res* 5:39–54
- Henderson JM, Quandt RE (1961) *Microeconomic theory*. McGraw-Hill, New York
- Horst R (1998) Deterministic global optimization with partition sets whose feasibility is not known: application to concave minimization, reverse convex constraints, dc-programming, and lipschitzian optimization. *J Optim Theory Appl* 58(1):11–37
- Jiao HW (2009) A branch and bound algorithm for globally solving a class of nonconvex programming problems. *Nonlinear Anal* 70:1113–1123
- Jiao HW, Liu SY, Zhao YF (2015) Effective algorithm for solving the generalized linear multiplicative problem with generalized polynomial constraints. *Appl Math Model* 39:7568–7582
- Keeney RL, Raiffa H (1993) *Decisions with multiple objective*. Cambridge University Press, Cambridge
- Konno H, Kuno T, Yajima Y (1994) Global minimization of a generalized convex multiplicative function. *J Glob Optim* 4:47–62
- Konno H, Fukaiishi K (2000) A branch-and-bound algorithm for solving low rank linear multiplicative and fractional programming problems. *J Glob Optim* 18:283–299
- Kuno T, Yajima Y, Konno H (1993) An outer approximation method for minimizing the product of several convex functions on a convex set. *J Glob Optim* 3:325–335
- Kuno T (2001) A finite branch-and-bound algorithm for linear multiplicative programming. *Comput Optim Appl* 20:119–135
- Liu XJ, Umegaki T, Yamamoto Y (1999) Heuristic methods for linear multiplicative programming. *J Glob Optim* 4(15):433–447
- Mulvey JM, Vanderbei RJ, Zenios SA (1995) Robust optimization of large-scale systems. *Oper Res* 43:264–81
- Quesada I, Grossmann IE (1996) *Alternative bounding approximations for the global optimization of various engineering design*. Springer, US
- Ryoo HS, Sahinidis NV (2003) Global optimization of multiplicative programs. *J Glob Optim* 26:387–418
- Samadi F, Mirzazadeh A, Pedram MM (2013) Fuzzy pricing, marketing and service planning in a fuzzy inventory model: a geometric programming approach. *Appl Math Model* 37:6683–6694
- Shen PP, Duan YP, Ma Y (2008) A robust solution approach for nonconvex quadratic programs with additional multiplicative constraints. *Appl Math Comput* 201:514–526
- Shen PP, Jiao HW (2006) Linearization method for a class of multiplicative programming with exponent. *Appl Math Comput* 183:328–336
- Shen PP, Jiao HW (2006) A new rectangle branch and pruning approach for generalized geometric programming. *Appl Math* 183:1027–1038
- Thoai NV (1991) A global optimization approach for solving the convex multiplicative programming problem. *J Glob Optim* 1:341–357
- Tuy H (1991) Effect of the subdivision strategy on convergence and efficiency of some global optimization algorithms. *J Glob Optim* 1:23–26
- Wang CF, Liu SY, Shen PP (2012) Global minimization of a generalized linear multiplicative programming. *Appl Math Model* 36:2446–2451
- Wang YJ, Liang ZA (2005) A deterministic global optimization algorithm for generalized geometric programming. *Appl Math Comput* 168:722–737

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
