A Global Method for Linear Multiplicative Programming Based on Simplicial Partition and Lagrange Dual Bound

Yonghong Zhang, Yaping Deng, Chunfeng Wang

Abstract—**We propose a global optimization method based on simplex subdivision and Lagrange dual bound to solve a specific type of problem known as linear multiplicative problem (LMP). The method utilizes the Lagrangian duality theory to solve the equivalent problem (ELMP) and obtain a lower bound by solving a linear programming problem (LP). To improve the convergence speed of the algorithm, we introduce two deleting rules that efficiently prune the feasible region without an optimal solution. These rules help accelerate the algorithm's convergence by reducing unnecessary computations. The convergence of the proposed algorithm is theoretically proven, ensuring its reliability and effectiveness. To validate its feasibility and performance, we conduct numerical experiments.**

Index Terms—**Linear multiplicative programming; Global optimization; Branch-and-bound; Lagrange duality**

I. Introduction

THE following form of problem LMP is considered in this paper:

$$\mathrm{LMP}:\begin{cases}\min & \sum\limits_{i=1}^{p}(\mathbf{c}_i^\top\mathbf{x}+d_i)(\mathbf{e}_i^\top\mathbf{x}+d_i)\\ \text{s.t.} & \mathbf{X}=\{\mathbf{x}\in\mathbf{R}^n:\mathbf{A}\mathbf{x}\le\mathbf{b},\mathbf{x}\ge\mathbf{0}\},\end{cases}$$

where $\mathbf{c}_i,\mathbf{e}_i\in\mathbf{R}^n, d_i, f_i\in\mathbf{R}, i=1,...,p,\mathbf{A}\in\mathbf{R}^{m\times n}$. Besides, $\mathbf{X}$ is a nonempty closed and bounded set.

Global optimization has been applied in many fields, including variance portfolio selections [1], multiobjective decision making [2,3], and so on. Since global optimization is a very challenging problem [4], it is very hard to solve such problem. For different kinds of optimization problems, a variety of methods have been presented. One of these approaches has been widely used is the branch and bound method [5]. For instance, when the objective function is a twice differentiable nonconvex function, by means of branch-and-bound framework, Jaroslav [6] proposed a deterministic global optimization method. Scholz [7] used geometric branch and bound approach to deal with problems in small dimensions. For standard quadratic programming, Liuzzi [8] proposed a branch and bound method on the basis of convex and LP bounds. For vertex triangulation feasible region, Zilinskas [9] presented a branch and bound method by simplicial partitions. For solving LMP, Wang et al. [10] developed a convexity method based on piecewise linear approximation for nonconvex part of the objective function.

Manuscript received May 10, 2023, revised August 31, 2023.

This work was supported by the Key cultivation project of Xianyang Normal University (XSYK21044).

Yonghong Zhang is a lecturer of the School of Mathematics and Statistics, Xianyang Normal University, Xianyang, 712000, PR China. (e-mail:zhangyonghong09@126.com)

Yaping Deng is a doctoral student of the School of Mathematics and Statistics, North China University of Water Resources and Electric Power, Zhengzhou, 450045, PR China. (email: 1514349724@qq.com)

Chunfeng Wang is a professor of the School of Mathematics and Statistics, Xianyang Normal University, Xianyang, 712000, PR China. (corresponding author, email:wangchunfeng09@126.com)

As a kind of global optimization, LMP has applications [11-15] in real life. Therefore, many optimization methods have been developed to solve this type of problem. For example, for a class of LMP with positive exponent, when $\mathbf{c}_i^\top\mathbf{x}+d_i>0$ and $\mathbf{e}_i^\top\mathbf{x}+f_i>0$, Zhang [16] developed a region-division-linearization method. To solve LMP, a rectangular branch and bound algorithm was presented by Shen [17]. When $\mathbf{c}_i^\top\mathbf{x}+d_i>0$, Zhou [18] designed a branch and bound algorithm based on simplicial duality strategy.

This paper gives a novel simplical partition and duality bound method. Compared with some other algorithms [16,17], there is no nonnegative requirement for the product term in LMP considered in this paper, i.e. the model considered in this paper has a wider range of applications. In this algorithm, firstly, the primal problem LMP is equivalently transformed into problem $\mathrm{ELMP}(Y)$. Then, the Lagrange weak duality technique is used to construct the lower bound of problem $\mathrm{ELMP}(Y)$, and construction method of feasible solution for problem LMP is also presented, which corresponding the objective function value is upper bound of problem LMP. Furethermore, combined with branch and bound framework, the new approach is designed. By using the proposed method, the primal problem LMP can be transformed into a series of linear programming, which greatly improves the efficiency. Finally, to speed up the algorithm, two deleting rules are presented.

The organization of this paper is as follows. In Section II, the transformation of problem LMP is firstly discussed. Then, the lower bound of problem LMP is given, which uses the weak duality theorem of Lagrange duality. The construction of feasible solution is also presented. The detailed algorithm and proof of convergence are given in Section III. In Section IV, computation experiences are provided and the data is analyzed.

II. Transformation

Through reorganization operation, the problem LMP can be equivalently transformed into the form below LMP1:

$$\begin{array}{ll}\min & \mathbf{x}^\top(\sum\limits_{i=1}^{p}\mathbf{c}_i\mathbf{e}_i^\top)\mathbf{x}+\sum\limits_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i)^T\mathbf{x}+\sum\limits_{i=1}^{p}d_if_i\\ \text{s.t.} & \mathbf{x}\in\mathbf{X}.\end{array}$$

To simply the representation, in LMP1, let Q $= \sum_{i=1}^{p} \mathbf{c}_i \mathbf{e}_i^\top$, and q_i be the i-th line of Q. We can rewrite LMP as follows:

$$\text{LMP1}: \begin{cases} \min & \mathbf{x}^\top Q\mathbf{x} + \sum_{i=1}^{p}(f_i\mathbf{c}_i + d_i\mathbf{e}_i)^T\mathbf{x} + \sum_{i=1}^{p} d_i f_i \\ \text{s.t.} & \mathbf{x} \in \mathbf{X}. \end{cases}$$

To tackle the optimization problem LMP1, we propose a branch and bound method based on simplex partition. In the following subsection, we outline the process of constructing a simplex and dividing it. We then proceed to explain how to obtain the lower and upper bounds for problem LMP.

A. *Initial simplex and simplicial partition*

Let $\mathbf{y} = (\sum_{i=1}^{p} \mathbf{c}_i \mathbf{e}_i^\top)\mathbf{x} =$ Q$\mathbf{x}$, i.e. $y_j = q_j\mathbf{x}, j = 1, \cdots, n$. We first construct an initial simplex such that it contains $\mathbf{y}$. The detailed process is given below.

To construct the initial simplex, compute $r = \min_{\mathbf{x}\in\mathbf{X}} \sum_{i=1}^{n} y_i$, and $r_i = \max_{\mathbf{x}\in\mathbf{X}} y_i = \max_{\mathbf{x}\in\mathbf{X}} q_i\mathbf{x}$, firstly. Then, let Y^0 be the convex hull with vertices $V_0^0, V_1^0, ..., V_n^0$, where $V_0^0 = (r_1, r_2, ..., r_n)^T$ For $j = 1, \cdots, n$, let

$$V_j^0 = (r_1, ..., r_{j-1}, \alpha_j, r_{j+1}, ..., r_n)^T, \alpha_j = r - \sum_{i\neq j} r_i.$$

It is easily got that $V_j^0 - V_0^0 = (0, ..., 0, r - \sum_{i=1}^{n} r_i, 0, ..., 0)^T$, where $r - \sum_{i=1}^{n} r_i$ is the j-th component of $V_j^0 - V_0^0$. The following Theorem 1 shows that Y^0 is the initial simplex we need.

Theorem 1. Y^0 is a vertex or n-dimensional simplex.

Proof: Since

$$r = \min_{\mathbf{x}\in\mathbf{X}} \sum_{i=1}^{n} y_i \leq \max_{\mathbf{x}\in\mathbf{X}} \sum_{i=1}^{n} y_i \leq \sum_{i=1}^{n} \max_{\mathbf{x}\in\mathbf{X}} y_i = \sum_{i=1}^{n} r_i,$$

we have

$$r = \sum_{i=1}^{n} r_i \text{ or } r < \sum_{i=1}^{n} r_i.$$

If $r = \sum_{i=1}^{n} r_i$, Y^0 is a vertex, $Y^0 = V_0^0$; else $r - \sum_{i=1}^{n} r_i \neq 0$, Y^0 is a n-dimensional simplex.

If Y^0 is a vertex, the optimal solution is determinate and unique. Therefore, we only consider the case that Y^0 is a n-dimensional simplex. According to the constructed simplex, an equivalent problem ELMP(Y^0) of LMP can be derived as follows:

$$\begin{array}{lll} v(Y) = & \min & \mathbf{x}^\top\mathbf{y} + \sum_{i=1}^{p}(f_i\mathbf{c}_i + d_i\mathbf{e}_i)^\top\mathbf{x} + \sum_{i=1}^{p} d_i f_i, \\ & \text{s.t.} & A\mathbf{x} \leq \mathbf{b}, \\ & & \mathbf{y} = \sum_{i=1}^{p} \mathbf{c}_i\mathbf{e}_i^\top\mathbf{x}, \\ & & \mathbf{x} \geq 0, \mathbf{y} \in Y^0. \end{array}.$$

The relationship between ELMP and LMP is given in the theorem below.

Theorem 2. If $(\mathbf{x}^*, \mathbf{y}^*)$ is the optimal solution of ELMP, then $\mathbf{x}^*$ is the optimal solution of LMP; on the contrary, if $\mathbf{x}^*$ is the optimal solution of LMP, then $(\mathbf{x}^*, \mathbf{y}^*)$ is the optimal solution of ELMP with $\mathbf{y}^* = \sum_{i=1}^{p} \mathbf{c}_i\mathbf{e}_i^\top\mathbf{x}^*$.

Proof: It can be obtained directly from the above derivation process.

As we know, in branch and bound framework, the division of simplex plays an important role. In this paper, the division process based on the longest side of the simplex is given as follows.

Let Y be a n-dimensional simplex with vertexes $\{V_0, V_1, ..., V_n\}$, and c be the midpoint of the longest side $[V_m, V_{\tilde{m}}]$, i.e.

$$V_m - V_{\tilde{m}} = \max_{i<\tilde{i}}\{\| V_i - V_{\tilde{i}} \|\}, i = 0, 1, ..., n,$$

then, Y^1 and Y^2 are subsimplex of Y, where the vertices of Y^1 and Y^2 are $\{V_0, V_1, ..., V_s, V_c, V_{s+1}, ..., V_n\}$ and $\{V_0, V_1, ..., V_{\tilde{s}}, V_c, V_{\tilde{s}+1}, ..., V_n\}$ respectively; $Y^1 \bigcup Y^2 = Y$, and $\mathrm{int}Y^1 \bigcap \mathrm{int}Y^2 = \emptyset$.

B. *Lower and upper bounds*

Assume that Y is the initial simplex or its sub-simplex sequence. We can then consider the equivalent problem ELMP(Y) below:

$$\begin{array}{lll} v(Y) = & \min & \mathbf{x}^\top\mathbf{y} + \sum_{i=1}^{p}(f_i\mathbf{c}_i + d_i\mathbf{e}_i)^\top\mathbf{x} + \sum_{i=1}^{p} d_i f_i \\ & \text{s.t.} & A\mathbf{x} \leq \mathbf{b}, \\ & & \mathbf{y} = \sum_{i=1}^{p} \mathbf{c}_i\mathbf{e}_i^\top\mathbf{x}, \\ & & \mathbf{x} \geq 0, \ \mathbf{y} \in Y. \end{array}.$$

In order to solve the problem ELMP(Y), constructing a lower bound is a crucial step in our proposed algorithm. This lower bound is derived using the weak Lagrangian duality of nonlinear programming. The construction process of the lower bound is outlined in Theorem 3, which is presented below.

Theorem 3. Let Y be a p-dimensional simplex, which is the initial simplex or its subsimplex, and its vertex set is $\{V_0, V_1, ..., V_n\}$. Then $LB(Y) \leq v(Y)$, where $LB(Y)$ is the optimal value obtained by following linear programming LP(Y):

$$\begin{array}{ll} \max & -\mathbf{b}^\top\lambda + t + \sum_{i=1}^{p} d_i f_i \\ \text{s.t.} & t \leq \xi^\top V_j, j = 0, 1, .., n, \\ & \mathbf{A}^\top\lambda - (\sum_{i=1}^{p} \mathbf{c}_i\mathbf{e}_i^\top)^\top\xi + V_j + \sum_{i=1}^{p}(f_i\mathbf{c}_i + d_i\mathbf{e}_i) \geq 0, \\ & \lambda \geq 0, \xi \in R^n. \end{array}$$

Proof: For problem LP(Y), by utilizing the weak Lagrangian duality, we have

$$\begin{aligned} LB(Y) = & \max_{\lambda\geq 0, \xi\in\mathbf{R}^n}\{\min_{\mathbf{x}\geq 0, \mathbf{y}\in Y}[\mathbf{x}^\top\mathbf{y} + \sum_{i=1}^{p}(f_i\mathbf{c}_i + d_i\mathbf{e}_i)^\top\mathbf{x} \\ & + \sum_{i=1}^{p} d_i f_i + \lambda^\top(\mathbf{A}\mathbf{x} - \mathbf{b}) + \xi^\top(\mathbf{y} - \sum_{i=1}^{p} \mathbf{c}_i\mathbf{e}_i^\top\mathbf{x})]\} \\ = & \max_{\lambda\geq 0, \xi\in\mathbf{R}^n}\{-\mathbf{b}^\top\lambda + \sum_{i=1}^{p} d_i f_i + \min_{\mathbf{x}\geq 0, \mathbf{y}\in Y}[\mathbf{x}^T(\sum_{i=1}^{p}(f_i\mathbf{c}_i \\ & + d_i\mathbf{e}_i) + \mathbf{y} + \mathbf{A}^\top\lambda - \sum_{i=1}^{p} \mathbf{e}_i\mathbf{c}_i^\top\xi) + \xi^\top\mathbf{y}]\}. \end{aligned}$$

Since

$$\min_{\mathbf{x}\geq 0,\mathbf{y}\in Y} \mathbf{x}^T(\sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i)+\mathbf{y}+\mathbf{A}^\top\lambda-\sum_{i=1}^{p}\mathbf{e}_i\mathbf{c}_i^\top\xi) = \begin{cases} 0, \text{ if } \sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i)+\mathbf{y}+\mathbf{A}^\top\lambda-\sum_{i=1}^{p}\mathbf{e}_i\mathbf{c}_i^\top\xi\geq 0, \\ -\infty, \text{ otherwise,} \end{cases}$$

we have

$$\begin{aligned} \max \quad & -\mathbf{b}^\top\lambda+\min_{\mathbf{y}\in Y}\{\xi^\top\mathbf{y}\}+\sum_{i=1}^{p}d_if_i \\ \text{s.t.} \quad & \sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i)+\mathbf{y}+\mathbf{A}^\top\lambda-\sum_{i=1}^{p}\mathbf{e}_i\mathbf{c}_i^\top\xi\geq 0, \quad (1) \\ & \lambda\geq 0,\xi\in\mathbf{R}^n,\forall\mathbf{y}\in Y. \end{aligned}$$

For $x\geq 0,\mathbf{y}\in Y$, if $\exists\mathbf{y}\in Y$ such that $\sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i)+\mathbf{y}+\mathbf{A}^\top\lambda-\sum_{i=1}^{p}\mathbf{e}_i\mathbf{c}_i^\top\xi<0$, the inner minimization problem can get the value of $-\infty$, therefore $\sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i)+\mathbf{y}+\mathbf{A}^\top\lambda-\sum_{i=1}^{p}\mathbf{e}_i\mathbf{c}_i^\top\xi\geq 0$.

By introducing a new variable t, the problem (1) can be equivalently transformed into the problem (2) below:

$$\begin{aligned} \max \quad & -\mathbf{b}^\top\lambda+t+\sum_{i=1}^{p}d_if_i \\ \text{s.t.} \quad & \sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i)+\mathbf{y}+\mathbf{A}^\top\lambda-\sum_{i=1}^{p}\mathbf{e}_i\mathbf{c}_i^\top\xi\geq 0,\forall\mathbf{y}\in Y, \\ & t\leq\xi^\top\mathbf{y}, \quad (2) \\ & \lambda\geq 0,\xi\in\mathbf{R}^n. \end{aligned}$$

For simplex Y with extreme points $V_i, i=0,1,...,n$, since the functions $t-\xi^\top\mathbf{y}$ and $\sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i)+\mathbf{y}+\mathbf{A}^\top\lambda-\sum_{i=1}^{p}\mathbf{e}_i\mathbf{c}_i^\top\xi$ are concave, $t<\xi^\top\mathbf{y}$ and $\sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i)+\mathbf{y}+\mathbf{A}^\top\lambda-\sum_{i=1}^{p}\mathbf{e}_i\mathbf{c}_i^\top\xi\geq 0$ hold if and only if $t\leq\xi^\top V_j, \sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i)+V_j+\mathbf{A}^\top\lambda-\sum_{i=1}^{p}\mathbf{e}_i\mathbf{c}_i^\top\xi\geq 0, j=0,1...,n$. Based on the above analysis, a linear programming problem LP(Y) can be obtained, and its optimal value is a lower bound of the initial problem.

$$\begin{aligned} \max \quad & -\mathbf{b}^\top\lambda+t+\sum_{i=1}^{p}d_if_i \\ \text{s.t.} \quad & t\leq\xi^\top V_j, j=0,1,..,n, \\ & \mathbf{A}^\top\lambda-(\sum_{i=1}^{p}\mathbf{c}_i\mathbf{e}_i^\top)^\top\xi+V_j+\sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i)\geq 0, \\ & \lambda\geq 0,\xi\in R^n. \end{aligned}$$

The proof is completed.

Proposition 1. (1) Let Y^1, Y^2 be two n-dimensional sub-simplexes, and satisfy $Y^1\subseteq Y^2$, then, $LB(Y^1)\geq LB(Y^2)$; (2) Let Y be a n-dimensional simplex with a vertex set $\{V_0,V_1,...,V_n\}$, then $LB(Y)>-\infty$.

Proof: (1) According to the transformation process of LP(Y), it can be obtained directly.

(2) On the basis of Theorem 3, we have

$$LB(Y)=\max_{\lambda\geq 0,\xi\in\mathbf{R}^n}\{\min_{\mathbf{x}\geq 0,\mathbf{y}\in Y}[\mathbf{x}^\top\mathbf{y}+\sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i)^\top\mathbf{x} +\sum_{i=1}^{p}d_if_i+\lambda^\top(\mathbf{A}\mathbf{x}-\mathbf{b})+\xi^\top(\mathbf{y}-\sum_{i=1}^{p}\mathbf{c}_i\mathbf{e}_i^\top\mathbf{x})]\}.$$

Let $\lambda=0,\xi=0$, then

$$LB(Y)=\min_{\mathbf{x}\geq 0,\mathbf{y}\in Y}[\mathbf{x}^\top\mathbf{y}+\sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i)^\top\mathbf{x}+\sum_{i=1}^{p}d_if_i].$$

Since $G(x)=\mathbf{x}^\top\mathbf{y}+\sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i)^\top\mathbf{x}+\sum_{i=1}^{p}d_if_i$ is continuous function, the feasible region $B=\{(\mathbf{x},\mathbf{y})|\mathbf{x}\geq 0,\mathbf{y}\in Y\}$ is compact, so $LB(Y)$ is finite, which means that $LB(Y)>-\infty$.

From Proposition 1, in the process of determining the lower bound, the lower bound is $+\infty$ or finite. When $LB(Y)=+\infty$, it means that ELMP has no feasible solution. With the progress of the algorithm, the corresponding simplex will be eliminated. So, the lower bound is finite.

To determine a upper bound, we need to find a feasible solution for LMP. Towards this end, the Theorem 5 below shows how to get a feasible solution for LMP.

Theorem 5. Let Y be a n-dimensional simplex with vertices $\{V_0,V_1,...,V_n\}$, and suppose that $LB(Y)\neq+\infty$, $(p_0,p_1,p_2,...,p_n)\in R^{n\times(n+1)}$ represents the optimal value of the dual variable corresponding to the first $n(n+1)$ constrains of the problem LP(Y), then $\omega=\sum_{i=0}^{n}p_i$ is feasible for problem LMP.

Proof: For problem LP(Y), it is easy to drive its dual problem as follows:

$$\begin{aligned} \min \quad & p_0^T(V_0+\sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i))+p_1^T(V_1+\sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i)) \\ & +...+{p_n}^T(V_n+\sum_{i=1}^{p}(f_i\mathbf{c}_i+d_i\mathbf{e}_i))+\sum_{i=1}^{p}d_if_i \\ \text{s.t.} \quad & -\mathbf{A}(\sum_{i=0}^{n}p_i)\geq-\mathbf{b}, \\ & \sum_{i=1}^{p}\mathbf{c}_i\mathbf{e}_i^\top(\sum_{i=0}^{n}p_i)-\sum_{i=0}^{n}V_ih_i=0, \\ & \sum_{i=0}^{n}h_i=1, \\ & p_i\geq 0, i=0,1,...,n. \end{aligned}$$

By the first constraint, we have $\mathbf{A}(\sum_{i=0}^{p}p_i)\leq\mathbf{b}$, so $\omega=\sum_{i=0}^{n}p_i$ is a feasible solution of problem LMP.

According to Theorem 5, we know that ω is feasible for problem LMP, thus, its objective function value provides an upper bound for the optimal value of problem LMP.

C. Deleting techniques

In order to improve the efficiency of the proposed method, we introduce two deleting rules. These rules are designed to eliminate simplices that are unable to contain the global optimal solutions. By applying these deleting rules, we can accelerate the speed of our method and focus on the simplices that have a higher likelihood of containing the global optimal solutions.

Let $\{V_0,V_1,\cdots,V_n\}$ be the vertices of a simplex.

(a) Deleting technique 1

Let $m_i=\min_{\mathbf{x}\in X}q_i\mathbf{x}$, $M_i=\max_{\mathbf{x}\in X}q_i\mathbf{x}$, if there exist $i\in\{1,2,\cdots,n\}$, such that

$$\min\{V_{0_i},\cdots,V_{n_i}\}>M_i \text{ or } \max\{V_{0_i},\cdots,V_{n_i}\}<m_i,$$

the simplex will be eliminated.

(b) Deleting technique 2

If there exists $i \in \{1, 2, \cdots, m\}$, which satisfies

$$\min\{A_i Q^{-1} V_0 - b_i, A_i Q^{-1} V_1 - b_i, \cdots, A_i Q^{-1} V_n - b_i\} > 0,$$

the simplex will be eliminated.

III. Algorithm description and convergence analysis

To incorporate the aforementioned techniques into the branch and bound framework, we provide the pseudo code of the proposed method in Algorithm 1. This algorithm outlines the step-by-step procedure for solving the problem ELMP, taking into account the deleting rules and other optimizations discussed earlier. The convergence analysis of the algorithm will also be provided to ensure the effectiveness of the proposed method.

Algorithm 1: Branch and bound algorithm

01: Initialization: Compute $LB(Y_0)$ by solving problem $\mathrm{LP}(Y^0)$, and let $(p_0^0, p_1^0, \cdots, p_n^0) \in R^{n\times(n+1)}$ be the optimal values of the dual variables corresponding to the first $n(n+1)$ constraints of problem $\mathrm{LP}(Y^0)$. Let $\omega_0 = \sum\limits_{i=1}^{n} p_i^0$, $u_0 = f(\omega_0)$, $l_0 = LB(Y^0)$, $B = \{Y^0\}$, $k = 1$.
02: Main Loop
03: Set $\omega_k = \omega_{k-1}$, $u_k = u_{k-1}$, $Y^k = Y^{k-1}$.
04: If $u_k = l_k$, stop: ω_k is an optimal solution of problem (ELMP), $v = l_k$, else continue.
05: Using bisectional method to divide Y^k into Y_1^k, Y_2^k, let $\widetilde{T} = \{Y_1^k, Y_2^k\}$. By using deleting techniques, delete simplex from $\widetilde{T}$. Let T be the obtained subset of $\widetilde{T}$.
06: For each $Y \in T$, running steps below:
07: Compute the optimal value $LB(Y)$ by solving linear program $\mathrm{LP}(Y)$;
08: If $LB(Y)$ is finite, let $(p_0, p_1, \cdots, p_n) \in R^{n\times(n+1)}$ be the optimal values of the dual variables corresponding to the first $n(n+1)$ constraints of problem $\mathrm{LP}(Y)$, let $\widehat{\omega} = \sum\limits_{i=1}^{n} p_i$.
09: If $f(\widehat{\omega}) < f(\omega_k), \omega_k = \widehat{\omega}, u_k = f(\widehat{\omega})$.
10: Set $B^k = \{B^{k-1} \setminus Y^k\} \bigcup \{T\}$.
11: Deleting the simplex Y for which satisfy $LB(Y) \geq u_k$.
12: If $B^k = \varnothing$, set $l_k = u_k$, and stop.
13: Else, set $l_k = \min\{LB(Y) \mid Y \in B^k\}$, and choose $Y^k \in B^k$ with $LB(Y^k) = l_k$.
13: Set $k = k + 1$, and go to step 02.

The following Theorem 6 gives the proof of convergence of the algorithm.

Theorem 6. Assuming the algorithm is infinite, $\{Y^r\}$ is an infinite nested sequence generated by the algorithm, and ω^* is the accumulation point of sequence $\{\omega^r\}_{r=0}^{\infty}$, then ω^* is the global optimal solution of problem LMP.

Proof: Suppose that $\{Y^r\}$ is the nested sequence obtained by algorithm. According Horst and Tuy [19], $\bigcap_r V^r = \{y^*\}$, $y^* \in \mathbf{R}^n$. For each simplex Y^r, the vertices are denoted by $V_j^r, j = 0, 1, \cdots, n$. Let $(p_0^r, p_1^r, \cdots, p_n^r) \in \mathbf{R}^{n\times(n+1)}$, $h^r = (h_0^r, h_1^r, \cdots, h_n^r) \in \mathbf{R}^{n+1}$ be the optimal solution of the dual problem of linear programming $\mathrm{LP}(Y^r)$. $U = \{h \in \mathbf{R}^{n+1} | \sum\limits_{i=0}^{n} h_i = 1, i = 0, 1, \cdots, n\}$, which is compact set.

According to Theorem 5, $\omega^r \in X$, as X is a closed and bounded set, $\{\omega^r\}$ has at least one convergent subsequence. Using $\{\omega^r\}_{r\in R}$ to represent any convergent subsequence, since $\omega^* = \lim_{r\in R} \omega^r$, we have $\omega^* \in X$. Next, we will show that ω^* is a global optimal solution of LMP.

Since U is bounded and closed, there is a finite subsequence R' of R, which makes $\lim_{r\in R'} h_j^r = h_j^*, j = 0, 1, 2, \cdots, n$ and satisfies $h^* \in U$. As $\lim_{r\in R} \omega^r = \omega^*$, so $\lim_{r\in R'} \omega^r = \omega^*$. As $\bigcap_r V^r = \{y^*\}$, so $\lim_{r\in R'} V_j^r = y^*, j = 0, 1, 2, \cdots, n$. For $R' \subset R$, we have $LB(Y^r) \leq LB(Y^{r'}) \leq v$. For the finite subsequence R' of R, $\lim_{r\in R'} LB(Y^r)$ exists and satisfies $\lim_{r\in R'} LB(Y^r) \leq v$. For $\forall r \in R'$, we plug $(p_0^r, p_1^r, \cdots, p_n^r)$ into objective function of problem $\mathrm{DLP}(Y)$:

$$\begin{aligned} LB(Y) &= p_0^{rT}(V_0^r + \sum_{i=1}^{p}(f_i\mathbf{c}_i + d_i\mathbf{e}_i)) \\ &+ p_1^{rT}(V_1^r + \sum_{i=1}^{p}(f_i\mathbf{c}_i + d_i\mathbf{e}_i)) \\ &+ \cdots + p_n^{rT}(V_n^r + \sum_{i=1}^{p}(f_i\mathbf{c}_i + d_i\mathbf{e}_i)) + \sum_{i=1}^{p} d_i f_i. \end{aligned}$$

Taking the limit of $r \in R'$, we have

$$\begin{aligned} &\lim LB(Y) \\ &= p_0^{*T}(y^* + \sum_{i=1}^{p}(f_i\mathbf{c}_i + d_i\mathbf{e}_i)) + p_1^{*T}(y^* + \sum_{i=1}^{p}(f_i\mathbf{c}_i + d_i\mathbf{e}_i)) \\ &\quad + \cdots + < p_n^*, y^* + \sum_{i=1}^{p}(f_i\mathbf{c}_i + d_i\mathbf{e}_i) > + \sum_{i=1}^{p} d_i f_i \\ &= (\sum_{i=0}^{n} p_i^*)^\top (y^* + \sum_{i=1}^{p}(f_i\mathbf{c}_i + d_i\mathbf{e}_i)) + \sum_{i=1}^{p} d_i f_i \\ &= \omega^{*\top}(y^* + \sum_{i=1}^{p}(f_i\mathbf{c}_i + d_i\mathbf{e}_i)) + \sum_{i=1}^{p} d_i f_i \\ &= \omega^{*\top} y^* + \sum_{i=1}^{p}(f_i\mathbf{c}_i + d_i\mathbf{e}_i)^\top \omega^* + \sum_{i=1}^{p} d_i f_i \leq v. \end{aligned}$$

Since ω^* is feasible for problem LMP, i.e. $\omega^* \in X$, we have $\mathbf{A}\omega^* \leq \mathbf{b}, \omega^* \geq 0$. For $r \in R'$, taking the limit of the constraints of $\mathrm{DLP}(Y)$, we have

$$\sum_{i=1}^{p} \mathbf{c}_i \mathbf{e}_i^\top (\sum_{i=1}^{p} p_i^r) - \sum_{i=1}^{p} V_i^r h_i = 0,$$

i.e.

$$\sum_{i=1}^{p} \mathbf{c}_i \mathbf{e}_i^\top (\sum_{i=1}^{p} p_i^*) - \sum_{i=1}^{p} y^* h_i = 0,$$

so $\sum\limits_{i=1}^{p} \mathbf{c}_i \mathbf{e}_i^\top \omega^* - y^* = 0$. On the basis of Theorem 5, $v(y^*) \geq v$ is held that is $\omega^{*\top} y^* + \sum\limits_{i=1}^{p}(f_i\mathbf{c}_i + d_i\mathbf{e}_i)^\top \omega^* + \sum\limits_{i=1}^{p} d_i f_i \geq v$.

To sum up, ω^* is the global optimal solution of problem LMP.

According to Theorem 6, if Algorithm 1 terminates in finite steps, a global optimal solution of problem LMP will be returned, else, an infinite nested sequence $\{Y^k\}$ will be generated, which satisfies $Y^{k+1} \subset Y^k$, then a global optimal solution of the initial problem can also be determined.

IV. Numerical experiments

In this section, to investigate the feasibility and effectiveness of algorithm, we give some specific examples 1-13 and a random example 14. This algorithm is coded in Matlab 2018a, which is implemented on the [Intel(R) Core(TM) i5-4200M CPU (2.5GHz)]. For examples 1-13, the results obtained by the proposed method are compared with Refs.[16,17,20,21]. The computational results are given in Table I, and the tolerance error is set $1e-4$. For example

14, a random example is designed, which is used to test Algorithm 1 further.

Example 1([20,21])

$$\begin{array}{ll}\min & (x_1+x_2)(x_1-x_2+7)\\ \text{s.t.} & 2x_1+x_2\le 14,\ x_1+x_2\le 10,\\ & -4x_1+x_2\le 0,\ 2x_1+x_2\ge 6,\\ & x_1+2x_2\ge 6,\ x_1-x_2\le 3,\\ & x_1+x_2\ge 0,\ x_1-x_2+7\ge 0,\\ & x_1,x_2\ge 0.\end{array}$$

Example 2([21])

$$\begin{array}{ll}\min & 3x_1-4x_2+(x_1+2x_2-1.5)(2x_1-x_2+4)\\ & +(x_1-2x_2+8.5)(2x_1+x_2-1)\\ \text{s.t.} & 5x_1-8x_2\ge -24,\ 5x_1+8x_2\le 44,\\ & 6x_1-3x_2\le 15,\ 4x_1+5x_2\ge 10,\\ & x_1,x_2\ge 0.\end{array}$$

Example 3([21])

$$\begin{array}{ll}\min & -x_1^2-x_2^2+(-x_1-3x_2+2)(4x_1+3x_2+1)\\ \text{s.t.} & x_1+x_2\le 5,\ -x_1+x_2\le 6,\\ & x_1,x_2\ge 0.\end{array}$$

Example 4([21])

$$\begin{array}{ll}\min & (2x_1-2x_2+x_3+2)(-2x_1+3x_2+x_3-4)\\ & +(-2x_1+x_2+x_3+2)(x_1+x_2-3x_3+5)\\ & +(-2x_1-x_2+2x_3+7)(4x_1-x_2-2x_3-5)\\ \text{s.t.} & x_1+x_2+x_3\le 10,\ x_1-2x_2+x_3\le 10,\\ & -2x_1+2x_2+2x_3\le 10,\ -x_1+x_2+3x_3\ge 6,\\ & x_1\ge 1,\ x_2\ge 1,\ x_3\ge 1.\end{array}$$

Example 5([21])

$$\begin{array}{ll}\min & x_1+(2x_1-3x_2+13)\times(x_1+x_2-1)\\ \text{s.t.} & -x_1+2x_2\le 8,\ -x_2\le -3,\\ & x_1+2x_2\le 12,\ x_1-2x_2\le -5,\ x_1,x_2\ge 0.\end{array}$$

Example 6([21])

$$\begin{array}{l}\min\ -2x_1^2-x_2^2-2+(-2x_1-3x_2+2)(4x_1+6x_2+2)\\ \quad +(3x_1+5x_2+2)(6x_1+8x_2+1)\\ \text{s.t. } 2x_1+x_2\le 10,\ -x_1+2x_2\le 10,\\ \quad x_1,x_2\ge 0.\end{array}$$

Example 7([21])

$$\begin{array}{l}\min\ (x_1+x_2)(x_1-x_2)+(x_1+x_2+2)(x_1-x_2+2)\\ \text{s.t. } x_1+2x_2\le 20,\quad x_1-3x_2\le 20,\\ \quad 1\le x_1\le 4,\quad 1\le x_2\le 4.\end{array}$$

Example 8([21])

$$\begin{array}{ll}\min & x_1+(x_1-x_2+5)(x_1+x_2-1)\\ \text{s.t.} & -2x_1-3x_2\le -9,\quad 3x_1-x_2\le 8,\\ & -x_1+2x_2\le 8,\quad x_1+2x_2\le 12,\\ & x_1\ge 0.\end{array}$$

Example 9([21])

$$\begin{array}{ll}\min & (2x_1-2x_2+x_3+2)(-2x_1+3x_2+x_3-4)\\ & +(-2x_1+x_2+x_3+2)(x_1+x_2-3x_3+5)\\ & +(-2x_1-x_2+2x_3+7)(4x_1-x_2-2x_3-5)\\ \text{s.t.} & x_1+x_2+x_3\le 10,\ x_1-2x_2+x_3\le 10,\\ & -x_1+x_2+3x_3\ge 6,\ x_1\ge 1,\ x_2\ge 1,\ x_3\ge 1.\end{array}$$

Example 10([21])

$$\begin{array}{l}\min\ (x_1+x_2)(x_1-x_2)+(x_1+x_2+2)(x_1-x_2+2)\\ \text{s.t. } x_1+2x_2\le 10,\quad x_1-3x_2\le 20,\\ \quad 0\le x_1\le 4,\quad 0\le x_2\le 4.\end{array}$$

Example 11([16])

$$\begin{array}{ll}\min & (x_1+2x_2-2)(-2x_1-x_2+3)\\ & +(3x_1-2x_2+3)(x_1-x_2-1)\\ \text{s.t.} & -2x_1+3x_2\le 6,\quad 4x_1-5x_2\le 8,\\ & 5x_1+3x_2\le 15,\quad -4x_1-3x_2\le -12,\\ & x_1\ge 0, x_2\ge 0.\end{array}$$

Example 12([18])

$$\begin{array}{ll}\min & (-x_1+2x_2-0.5)(-2x_1+x_2+6)\\ & +(3x_1-2x_2+0.5)(x_1+x_2-1)\\ \text{s.t.} & -5x_1+8x_2\le 24,\ 5x_1+8x_2\le 44,\\ & 6x_1-3x_2\le 15,\ -4x_1-5x_2\le -10,\\ & x_1\ge 0, x_2\ge 0.\end{array}$$

Example 13([18])

$$\begin{array}{ll}\min & (-x_1+2x_2-0.5)(-2x_1+x_2+6)\\ & +(3x_1-2x_2+0.5)(x_1+x_2-1)\\ \text{s.t.} & -5x_1+8x_2\le 24,\quad 5x_1+8x_2\le 44,\\ & 6x_1-3x_2\le 15,\quad -4x_1-5x_2\le -10,\\ & x_1\ge 0, x_2\ge 0.\end{array}$$

Table I: Computational results of test Examples 1-13($\epsilon=10^{-4}$)

Examples	Method	Time(s)	Iter	Optimal value	Optimal solution
1	[20]	0.3	53	10	(2, 8)
	[21]	0.7090	29	10	(2, 8)
	Ours	0.2371	5	10	(2, 8)
2	[21]	7.0751	182	-2.5	(0, 3)
	Ours	2.7772	93	-2.5	(0, 3)
3	[21]	2.1484	83	-233	(0, 5)
	Ours	0.1475	4	-233	(0, 5)
4	[21]	19.0745	602	-109.75	(5.5, 1, 3.5)
	Ours	4.5604	155	-109.75	(5.5, 1, 3.5)
5	[21]	3.5307	172	3	(0, 4)
	Ours	0.1045	2	3	(0, 4)
6	[21]	0.3551	15	4	(0,0)
	Ours	0.3517	11	4	(0,0)
7	[21]	1.0279	52	-22	(1, 4)
	Ours	0.2864	9	-22	(1, 4)
8	[21]	2.4927	89	3	(0, 4)
	Ours	1.8508	66	3	(0, 4)
9	[21]	19.0745	602	-109.75	(5.5, 1, 3.5)
	Ours	14.9170	136	-109.75	(5.5, 1, 3.5)
10	[21]	0.2143	19	-22	(1, 4)
	Ours	0.2918	9	-28	(0, 4)
11	[16]	10.013	7	-16.2837	(1.547, 2.421)
	Ours	5.6460	81	-16.5049	(1.5486, 2.4190)
12	[18]	0.06	29	10.6750	(1.5549, 0.7561)
	Ours	1.6093	22	10.6810	(1.5825, 0.7340)
13	[18]	0.06	29	10.6750	(1.5549, 0.7561)
	Ours	1.6093	22	10.6810	(1.5825, 0.7340)

From the results presented in Table I, it is evident that our algorithm consistently outperforms other methods on the majority of the test functions. One notable example is example 11, where our method achieves a significantly superior optimal value compared to that of Zhang [16]. Similarly, for example 12, although our method requires more iterations compared to Zhou [18], the final optimal result is superior to theirs. These results demonstrate the effectiveness and efficiency of our proposed algorithm in solving these test functions.

Example 14([21])

$$\begin{array}{ll}\min & \sum_{i=1}^{p}(\mathbf{c}_i^\top\mathbf{x}+d_i)(\mathbf{e}_i^\top\mathbf{x}+f_i)\\ \text{s.t.} & \mathbf{A}\mathbf{x}\le\mathbf{b}, 0\le\mathbf{x}\le 1.\end{array}$$

where $\mathbf{c}_i, \mathbf{e}_i$ are randomly generated in $[-1, 1]^n$, d_i, f_i are randomly generated in $[0, 100]$, all elements of $\mathbf{A}$ are randomly generated in $[0, 1]^{m \times n}$, the value of $\mathbf{b}$ is randomly generated in $[0, n]^n$. The tolerance error is set $1e - 1$. For this example, the computational results are given in Table II. In Table II, some notations are used as follows:
n: number of variables;
p: number of sums;
m: number of constrains;
AVG.Time: average iteration time in second;
AVG.Iter: average iteration numbers.

The average time (AVG.Time) and average number of iterations (AVG.Iter) were calculated based on 10 randomly generated examples for each problem size. As shown in Table II, it is evident that both the number of iterations and the running time increase as the values of n and p increase.

Furthermore, the results presented in Table II demonstrate that the proposed method is capable of effectively solving the LMP problem. It is also worth noting that the proposed method performs well for problems of moderate size. These findings highlight the effectiveness and scalability of the proposed method in addressing the LMP problem.

Table II: Computational results of test Examples 14($\epsilon = 10^{-1}$)

(n, p, m)	Our proposed algorithm	
	AVG.Time(s)	AVG.Iter
(5,10,50)	0.3741	8.3
(8,10,50)	2.5107	66.2
(10,10,50)	3.3300	77.3
(10,50,50)	10.7013	244.4
(10,100,50)	18.3719	406
(10,300,50)	41.8359	834.8
(10,500,50)	61.6894	1180.7
(10,800,50)	71.3844	1286
(10,1000,50)	85.0888	1590
(12,10,50)	111.1209	1634.4

REFERENCES

[1] H.M. Markowitz, "Portfolio selection", *Journal of the Institute of Actuaries*, vol. 7, pp. 77-91, 1952,

[2] H.P. Benson, "Vector maximization with two objective functions", *Journal of Global Optimization*, vol. 28, no. 2, pp. 253-257, 1997.

[3] D. Dennis, "Analyzing public inputs to multiple objective decisions on national forests using conjoint analysis", *Forest Science*, vol. 44, no. 3, pp. 421-429, 1998.

[4] V. Kreinovich, R. Kearfott, "Beyond convex? global optimization is feasible only for convex objective functions: a theorem", *Journal of Global Optimization*, vol. 33, no. 4, pp. 617-624, 2005.

[5] A. Neumaier, "Complete search in continuous global optimization and constraint satisfaction", *Acta Numerica*, vol. 13, no. 1, pp. 271-369, 2004.

[6] M. Jaroslav, I. M. Nicholas, and L.F. Chris, "A branch and bound algorithm for the global optimization of hession lipschitz continuous functions", *Journal of Global Optimization*, vol. 56, pp. 1791-1815, 2013.

[7] D. Scholz, "Geometric branch and bound methods for constrained global optimization problems", *Journal of Global Optimization*, vol. 57, no. 3, pp. 771-782, 2013.

[8] G. Liuzzi, M. Locatelli, and V. Piccialli, "A new branch and bound algorithm for standard quadratic programming problems", *Optimization Methods and Software*, vol. 34, no. 1, pp. 79-97, 2019.

[9] J. Zilinskas, "Branch and bound with simplicial partitions for global optimization", *Mathemacial Monelling and Analysis*, vol. 13, no. 1, pp. 145-159, 2008.

[10] C.F. Wang, Y.P. Deng, and X.D. Wu, "A convexity method for linear multiplicative programming", *IAENG International Journal of Applied Mathematics*, vol. 51, no. 1, pp. 66-70, 2022.

[11] C.D. Marinas, I.P. Andreoulakis, and C.A. Floudas, "Solving long term financial planning problems via global optimization", *Journal of Economic Dynamics and Control*, vol. 21, pp. 1405-1425, 1997.

[12] H. Konno, H. Shirakawa, and H. Yamazaki, "A mean-absolute deviation-skewness portfolio optimization model", *Annals of Operations Research*, vol. 45, pp. 205-220, 1993.

[13] M.C. Dorneich, and N.V. Sahinidis, "Global optimization algorithms for chip design and compaction", *Engineering Optimization*, vol. 25, no. 3, pp. 131-154, 1995.

[14] K.P. Bennett, and O.L. Mangasarian, "Bilinear separation of two sets in n-space", *Computational Optimization Applications*, vol. 2, pp. 207-227, 1994.

[15] J.M. Mulvey, R.J. Vanderbei, and S.A. Zenios, "Robust optimization of large-scale systems", *Operations Research*, vol. 43, pp. 264-281, 1995.

[16] B. Zhang, Y.L. Gao, and X. Liu, et al., "An efficient polynomial time algorithm for a class of generalized linear multiplicative programs with positive exponents", *Mathematical Problems in Engineering*, doi:10.1155/2021/8877037, 2021.

[17] P.P. Shen, and B.D. Huang, "Global Algorithm for Solving Linear Multiplicative Programming Problems", *Optimization Letters*, vol. 14, pp. 693-710, 2020.

[18] X.G. Zhou, and B.Y. Cao, "A simplicial branch and bound duality-bounds algorithm to linear multiplicative programming", *Journal of Applied Mathematics*, vol. 4, no. 8, pp. 289-325, 2013.

[19] R. Horst, and H. Tuy, "Global Optimization: Deterministic Approaches", Springer, Berlin, Germany, 1993.

[20] Y.L. Gao, C.X. Xu, and Y.T. Yang, "Outcome-space branch andbound algorithmfor solving linear multiplicative programming", *Computational Intelligence and Security*, vol. 3801, pp. 675C681, 2005.

[21] H. Jiao, W. Wang, R. Chen, Y. Shang, and J. Yin, "An efficient outer space algorithm for generalized linear multiplicative programming problem", *IEEE Access*, vol. 8, pp. 80629-80637, 2020.