# An Effective Computational Algorithm for a Class of Linear Multiplicative Programming

Jingben Yin

Department of Mathematics, Henan Institute of Science and Technology, Xinxiang 453003, China Email: jingbenyin@163.com

Yutang Liu

Henan Mechanical and Electrical Engineering College, Xinxiang 453002, China Email: liuyutang168@126.com

Baolin Ma

School of Mathematical Sciences, Henan Institute of Science and Technology, Xinxiang 453003, China Email: xxdsmbl@163.com

Dongwei Shi Department of Mathematics, Henan Institute of Science and Technology, Xinxiang 453003, China Email:shidw99@163.com

Abstract-In this paper, an effective computational algorithm is proposed for a class of linear multiplicative problem (P), which have broad applications in financial optimization, economic plan, engineering designs and stability analysis of nonlinear systems, and so on. By utilizing piecewise linearization technique underestimates the objective function, linear relaxation programming of the original linear multiplicative programming problem **(P)** is established, and the proposed global optimization algorithm is convergent to the global optimal solution of the original problem (P). And finally the numerical experiments are given to illustrate that the feasibility of proposed algorithm and can be used to globally solve the class of linear multiplicative programming problem (P).

*Index Terms*—linear multiplicative programming, global optimization, effective computational algorithm

#### I. INTRODUCTION

Consider the following a class of linear multiplicative programming problem:

(P): 
$$\begin{cases} \min f(x) = \sum_{j=1}^{p} (\sum_{i=1}^{n} c_{ji} x + d_{j}) (\sum_{i=1}^{n} e_{ji} x_{i} + f_{j}) \\ s.t. \quad Ax \le b \end{cases}$$

where  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ . In general, the problem (P) corresponds to a nonlinear optimization problem with non-convex objective function.

When p = 2, the problem (P) is linear multiplicative programming problems (abbreviated as LMP), which is a special type of nonconvex quadratic programming problems whose objective function is the product of two linear functions [1, 2, 3]. We introduced an auxiliary variable and defined the master problem which is equivalent to the original one. Then we applied a parametric simplex algorithm to the master problem. We demonstrated that our algorithm can solve LMP in a little more computational time than needed for solving the associated linear program (i.e., a linear program with the same constraints).

When  $p \ge 3$ , problem (P) is generalized linear multiplicative programming problems (GLMP), whose objective function is the sum of a convex function and a product of two linear functions [3, 4]. We showed that a parametric programming approach gives us a practical method to calculate a global minimum of GLMP.

Linear multiplicative programming (P) has attracted considerable attention in the literature because of their large number of practical applications in various fields of study [5, 6], including financial optimization [1, 7, 8], plant layout design [2, 9, 10], robust optimization [3, 11, 12], and so on [13, 14]. Hence, it is very necessary to present effective algorithm for solving linear multiplicative programming problem (P) [15, 16, 17].

Since problem (P) may possess many local minima, it is known to the hardest problems [18, 19, 20]. In the last decade, many solution algorithms have been proposed for locally solving linear multiplicative programming problem (P) [21, 22]. They can be classified as follows: outer-approximation methods [4, 23, 24], decomposition method [5, 25, 26], finite branch and bound algorithm [6,

Corresponding author E-mail: jingbenyin@163.com (J. Yin);

7, 27], primal and dual simplex method [8], cutting plane method [9, 28], heuristic methods [10], etc. [11, 12, 29].

Though local optimization methods for solving linear multiplicative programming (P) are ubiquitous, the global optimization algorithm of the class of linear multiplicative programming problems has been little studied in the literatures.

In the paper, a new branch and bound algorithm is given via solving a sequence of linear relaxations over partitioned subsets to find a global optimal solution of the (P). By utilizing new linearization technique, the initial nonconvex nonlinear programming problem (P) is systematically converted into a series of linear programming problems. The solution of these converted problems can be as close as possible to the globally optimal solution of the (P) by successive refinement process. In this method, (1) a new linear relaxation for the original problem of the (P) is proposed, thus any effective linear programming algorithm can be used to solve nonlinear programming problem (P); (2) the generated relaxed linear programming problems are embedded within a branch-and-bound algorithm without increasing new variables and constraints; (3) numerical computation shows that the proposed method is feasible.

This paper is organized as follows. In Section 2 linearization relaxation technique is presented for generating the relaxation linear programming. In Section 3 the proposed branch-and-bound algorithm in which the relaxed sub-problems are embedded is described, and the convergence of the algorithm is established. Numerical results of some problems in the area of application are considered in Section 4 and Section 5 provides a summary.

#### **II. LINEAR RELAXATION PROGRAMMING**

In order to saving the problem (P), the principal structure in the development of a solution procedure for solving problem (P) is the construction of lower bounds for this problem, as well as for its partitioned sub-problems. A lower bound on the solution of problem (P) and its partitioned sub-problems can be obtained by solving a linear programming relaxation problem.

Firstly, we solve the following 2n linear programming problems:

and

$$l_i = \begin{cases} \min & x_i \\ s.t. & Ax \le b, \end{cases}$$

$$u_i = \begin{cases} \max x_i \\ s.t. \quad Ax \le b, \end{cases}$$

where i = 1, ..., n. Then we can get the initial interval vector

$$X^{0} = \{x \in \mathbb{R}^{n} | l \le x \le u\},\$$
$$l = (l_{1}, l_{2}, \dots, l_{n})^{T},\$$
$$u = (u_{1}, u_{2}, \dots, u_{n})^{T}.$$

Next, the problem can be rewritten as follows:

(P): 
$$\begin{cases} \min f(x) = \sum_{j=1}^{p} (\sum_{i=1}^{n} c_{ji} x + d_{j}) (\sum_{i=1}^{n} e_{ji} x_{i} + f_{j}) \\ s.t. \quad Ax \le b, \\ x \in X^{0} \end{cases}$$

The linear relaxation of the problem (P) can be realized by underestimating every objective function with a linear function. All the details of this linearization technique for generating relaxations will be given as follows.

Let

$$X^{k} = \{ x \in \mathbb{R}^{n} : l^{k} \le x \le u^{k} \} \subseteq X^{0},$$
$$l^{k} = (l_{1}^{k}, l_{2}^{k}, \cdots, l_{n}^{k})^{T}, \ u = (u_{1}^{k}, u_{2}^{k}, \cdots, u_{n}^{k})^{T},$$

then

$$f(x) = \sum_{j=1}^{p} (\sum_{i=1}^{n} c_{ji} x_{i} + d_{j}) (\sum_{i=1}^{n} e_{ji} x_{i} + f_{j})$$

$$= \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} x_{i} x_{k} + \sum_{j=1}^{p} (d_{j} \sum_{i=1}^{n} e_{ji} x_{i} + f_{j} \sum_{i=1}^{n} c_{ji} x_{i})$$

$$+ \sum_{j=1}^{p} d_{j} f_{j}$$

$$= \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} \frac{1}{2} c_{ji} e_{jk} \left( (x_{i} + x_{k})^{2} - x_{i}^{2} - x_{k}^{2} \right)$$

$$+ \sum_{j=1}^{p} (d_{j} \sum_{i=1}^{n} e_{ji} x_{i} + f_{j} \sum_{i=1}^{n} c_{ji} x_{i}) + \sum_{j=1}^{p} d_{j} f_{j}$$

$$= \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} (x_{i} + x_{k})^{2}$$

$$- \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} x_{k}^{2}$$

$$+ \sum_{i=1}^{p} (d_{j} \sum_{i=1}^{n} e_{ji} x_{i} + f_{j} \sum_{i=1}^{n} c_{ji} x_{i})$$

$$\sum_{j=1}^{n} (u_j \sum_{i=1}^{n} e_{ji} x_i + J_j \sum_{i=1}^{n} e_{ii} x_i + J$$

 $+\sum_{j=1}^{p}d_{j}f_{j}$ 

In the following, we proved that for  $\forall x_i \in [l_i, u_i]$ , we have

 $(l_i + u_i)x_i - \frac{(l_i + u_i)^2}{4} \le x_i^2 \le (l_i + u_i)x_i - l_iu_i,$ 

And

$$\max_{l \le x \le u} \left( x_i^2 - [(l_i + u_i)x_i - \frac{(l_i + u_i)^2}{4}] \right)$$
  
= 
$$\max_{l \le x \le u} \left( (l_i + u_i)x_i - l_iu_i - x_i^2 \right)$$
  
= 
$$\frac{(u_i - l_i)}{4}$$

In fact, according to the geometric property of  $x_i^2$  in the region  $[l_i, u_i]$ , the conclusion is very clear.

On a one hand, for  $\forall x_i \in [l_i, u_i]$ , since

$$x_i^2 - [(l_i + u_i)x_i - \frac{(l_i + u_i)^2}{4}]$$

is a convex function, so the maximum value can be got at  $l_i$  or  $u_i$ , so we can know

$$\max_{\substack{l \le x \le u}} \left( x_i^2 - \left[ (l_i + u_i) x_i - \frac{(l_i + u_i)^2}{4} \right] \right)$$
$$= \frac{(u_i - l_i)}{4}$$

On the other hand, for  $\forall x_i \in [l_i, u_i]$ , since

$$(l_i + u_i)x_i - l_iu_i - x_i^2$$

is a concave function, so its maximum value can be got at  $\frac{l_i + u_i}{2}$ , so we can get

$$\max_{l \le x \le u} \left( (l_i + u_i) x_i - l_i u_i - x_i^2 \right) = \frac{(u_i - l_i)}{4}.$$
  
In the following, we consider

 $2x_i x_k = (x_i + x_k)^2 - x_i^2 - x_k^2.$ 

Let

$$\begin{split} f_{ijk}\left(x_{i}\right) = \begin{cases} (l_{i}+u_{i})x_{i}-l_{i}u_{i}, & c_{ji}e_{jk} > 0\\ (l_{i}+u_{i})x_{i}-\frac{(l_{i}+u_{i})^{2}}{4}, c_{ji}e_{jk} < 0 \end{cases},\\ f_{ijk}\left(x_{i},x_{k}\right) = \begin{cases} (l_{i}+l_{k}+u_{k}+u_{i})(x_{i}+x_{k}) - \\ (l_{i}+l_{k})(u_{i}+u_{k}), & c_{ji}e_{jk} < 0;\\ (l_{i}+l_{k}+u_{k}+u_{i})(x_{i}+x_{k}) - \\ -\frac{(l_{i}+l_{k}+u_{k}+u_{i})^{2}}{4}, & c_{ji}e_{jk} > 0. \end{cases} \end{split}$$

Let

$$\varphi(\mathbf{x}) = \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} \left( f_{ijk}(x_i, x_k) - f_{ijk}(x_i) - f_{ijk}(x_k) \right) \\ + \sum_{j=1}^{p} \left( d_j \left( \sum_{i=1}^{n} e_{ji} x_i + f_j \right) + f_j \left( \sum_{i=1}^{n} c_{ji} x_i + d_j \right) \right) \\ - \sum_{j=1}^{p} d_j f_j$$

we can get a linear relaxation programming of problem (P) in  $X^k$ :

$$RLP(S^{k}):\begin{cases} \min & \varphi(x) \\ s.t. & Ax \le b \\ & x \in X^{k} = \{x: l^{k} \le x \le u^{k}\} \end{cases}$$

**Theorem 1. For**  $\forall x_i \in [l_i, u_i]$ , we have

(i) 
$$\varphi(x) \le f(x)$$
;  
(ii)  $\lim_{\|u^k \to t^k\| \to 0} [f(x) - \varphi(x)] = 0, x \in X^k$ .  
where

$$\varphi(x) = \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} \left( f_{ijk}(x_i, x_k) - f_{ijk}(x_i) - f_{ijk}(x_k) \right) \\ + \sum_{j=1}^{p} \left( d_j \left( \sum_{i=1}^{n} e_{ji} x_i + f_j \right) + f_j \left( \sum_{i=1}^{n} c_{ji} x_i + d_j \right) \right) - \sum_{j=1}^{p} d_j f_j$$

and

$$f(x) = \sum_{j=1}^{p} (\sum_{i=1}^{n} c_{ji} x_{i} + d_{j}) (\sum_{i=1}^{n} e_{ji} x_{i} + f_{j})$$
  
$$= \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} (x_{i} + x_{k})^{2}$$
  
$$- \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} x_{i}^{2} - \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} x_{k}^{2}$$
  
$$+ \sum_{j=1}^{p} (d_{j} \sum_{i=1}^{n} e_{ji} x_{i} + f_{j} \sum_{i=1}^{n} c_{ji} x_{i})$$
  
$$+ \sum_{j=1}^{p} d_{j} f_{j}$$

**Proof.** (i) by the definition of  $\varphi(x)$ , we have

$$\begin{split} \varphi(\mathbf{x}) &= \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} \left( f_{ijk}(x_{i}, x_{k}) - f_{ijk}(x_{i}) - f_{ijk}(x_{k}) \right) \\ &+ \sum_{j=1}^{p} \left( d_{j} \left( \sum_{i=1}^{n} e_{ji} x_{i} + f_{j} \right) + f_{j} \left( \sum_{i=1}^{n} c_{ji} x_{i} + d_{j} \right) \right) - \sum_{j=1}^{p} d_{j} f_{j} \\ &\leq \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} (x_{i} + x_{k})^{2} \\ &- \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} x_{i}^{2} - \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} x_{k}^{2} \\ &+ \sum_{j=1}^{p} \left( d_{j} \left( \sum_{i=1}^{n} e_{ji} x_{i} + f_{j} \right) + f_{j} \left( \sum_{i=1}^{n} c_{ji} x_{i} + d_{j} \right) \right) - \sum_{j=1}^{p} d_{j} f_{j} \\ &= \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} \frac{1}{2} c_{ji} e_{jk} \left( (x_{i} + x_{k})^{2} - x_{i}^{2} - x_{k}^{2} \right) \\ &+ \sum_{j=1}^{p} \left( d_{j} \left( \sum_{i=1}^{n} e_{ji} x_{i} + f_{j} \right) + f_{j} \left( \sum_{i=1}^{n} c_{ji} x_{i} + d_{j} \right) \right) - \sum_{j=1}^{p} d_{j} f_{j} \\ &= \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} x_{i} x_{k} + \sum_{j=1}^{p} \left( d_{j} \left( \sum_{i=1}^{n} e_{ji} x_{i} + f_{j} \right) + f_{j} \left( \sum_{i=1}^{n} c_{ji} x_{i} + f_{j} \right) \right) \\ &+ f_{j} \left( \sum_{i=1}^{n} c_{ji} x_{i} + d_{j} \right) \right) - \sum_{j=1}^{p} d_{j} f_{j} \\ &= \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{i=1}^{n} c_{ji} x_{i} + d_{j} \right) \left( \sum_{i=1}^{p} e_{ji} x_{i} + f_{j} \right) \\ &= f(x) \end{split}$$

Therefore, we have

$$\varphi(x) \le f(x)$$
 for  $\forall x_i \in [l_i, u_i]$ .

(ii) for  $\forall x_i \in [l_i, u_i]$ , and the definition of  $\varphi(x)$  and f(x), we can get

$$\begin{split} |f(x) - \varphi(x)| \\ \leq \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} \left| \frac{(u_{k} - l_{k} + u_{i} - l_{i})^{2}}{4} \right| \\ + \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} \left| \frac{(u_{i} - l_{i})^{2}}{4} \right| + \left| \frac{(u_{k} - l_{k})^{2}}{4} \right| \\ = \left| \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} (x_{i} + x_{k})^{2} - \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} x_{i}^{2} - \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} x_{i}^{2} - \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} x_{k}^{2} \\ + \sum_{j=1}^{p} \left( d_{j} (\sum_{i=1}^{n} e_{ji} x_{i} + f_{j}) + f_{j} (\sum_{i=1}^{n} c_{ji} x_{i} + d_{j}) \right) \\ - \sum_{j=1}^{p} d_{j} f_{j} \\ - \frac{1}{2} \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} \left( f_{ijk} (x_{i}, x_{k}) - f_{ijk} (x_{i}) - f_{ijk} (x_{k}) \right) \\ - \sum_{j=1}^{p} \left( d_{j} (\sum_{i=1}^{n} e_{ji} x_{i} + f_{j}) + f_{j} (\sum_{i=1}^{n} c_{ji} x_{i} + d_{j}) \right) + \sum_{j=1}^{p} d_{j} f_{j} \right|$$

$$= \frac{1}{2} \left| \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} \left( \frac{\left[ (x_{i} + x_{k})^{2} - f_{ijk}(x_{i}, x_{k}) \right]}{-\left[ x_{i}^{2} - f_{ijk}(x_{i}) \right] - \left[ x_{k}^{2} - f_{ijk}(x_{k}) \right]} \right) \right|$$
  
Since

 $(x_i + x_k)^2 - f_{iik}(x_i, x_k)$ 

$$=(x_{i}+x_{k})^{2} - \begin{cases} (l_{i}+l_{k}+u_{k}+u_{i})(x_{i}+x_{k}) - \\ (l_{i}+l_{k})(u_{i}+u_{k}), & c_{ji}e_{jk} < 0; \\ (l_{i}+l_{k}+u_{k}+u_{i})(x_{i}+x_{k}) \\ -\frac{(l_{i}+l_{k}+u_{k}+u_{i})^{2}}{4}, & c_{ji}e_{jk} > 0. \end{cases}$$

$$= \begin{cases} (x_i + x_k)^2 - [(l_i + l_k + u_k + u_i)(x_i + x_k) - (l_i + l_k)(u_i + u_k)], & c_{ji}e_{jk} < 0; \\ (x_i + x_k)^2 - [(l_i + l_k + u_k + u_i)(x_i + x_k) - (l_i + l_k + u_k + u_i)^2] \\ - \frac{(l_i + l_k + u_k + u_i)^2}{4} ], & c_{ji}e_{jk} > 0. \end{cases}$$

By the Theorem 1, we have

$$(x_i + x_k)^2 - [(l_i + l_k + u_k + u_i)(x_i + x_k) - (l_i + l_k)(u_i + u_k)] \to 0$$

as  $\|u_k - l_k\| \to 0$ , and

$$(x_{i} + x_{k})^{2} - [(l_{i} + l_{k} + u_{k} + u_{i})(x_{i} + x_{k}) - \frac{(l_{i} + l_{k} + u_{k} + u_{i})^{2}}{4}] \rightarrow 0$$

as  $||u_k - l_k|| \to 0$ . So that we have

$$(x_i + x_k)^2 - f_{ijk}(x_i, x_k) \to 0 \text{ as } ||u_k - l_k|| \to 0.$$

Since

$$x_{i}^{2} - f_{ijk}(x_{i})$$

$$= x_{i}^{2} - \begin{cases} (l_{i} + u_{i})x_{i} - l_{i}u_{i}, & c_{ji}e_{jk} > 0\\ (l_{i} + u_{i})x_{i} - \frac{(l_{i} + u_{i})^{2}}{4}, c_{ji}e_{jk} < 0 \end{cases}$$

$$= \begin{cases} x_{i}^{2} - [(l_{i} + u_{i})x_{i} - l_{i}u_{i}], & c_{ji}e_{jk} > 0\\ x_{i}^{2} - [(l_{i} + u_{i})x_{i} - \frac{(l_{i} + u_{i})^{2}}{4}], c_{ji}e_{jk} < 0 \end{cases}$$

By the Theorem 1, we have

 $x_k^2 - f_{iik}(x_k)$ 

$$(x_i)^2 - f_{ijk}(x_i) \to 0 \text{ as } ||u_k - l_k|| \to 0.$$
  
Since

Since

$$= x_k^2 - \begin{cases} (l_i + u_i)x_k - l_iu_i, & c_{ji}e_{jk} > 0\\ (l_i + u_i)x_k - \frac{(l_i + u_i)^2}{4}, c_{ji}e_{jk} < 0 \end{cases}$$
$$= \begin{cases} x_k^2 - [(l_i + u_i)x_k - l_iu_i], & c_{ji}e_{jk} > 0\\ x_k^2 - [(l_i + u_i)x_k - \frac{(l_i + u_i)^2}{4}], c_{ji}e_{jk} < 0 \end{cases}$$

By the Theorem 1, we have

$$(x_k)^2 - f_{ijk}(x_k) \to 0 \text{ as } ||u_k - l_k|| \to 0,$$

Therefore, we have  $\lim_{\|u^k - l^k\| \to 0} [f(x) - \varphi(x)]$ 

$$= \lim_{\|u^{k} - t^{k}\| \to 0} \frac{1}{2} \left| \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} \begin{pmatrix} [(x_{i} + x_{k})^{2} - f_{ijk}(x_{i}, x_{k})] \\ -[x_{i}^{2} - f_{ijk}(x_{i})] - [x_{k}^{2} - f_{ijk}(x_{k})] \end{pmatrix} \right|$$
  
$$= \frac{1}{2} \lim_{\|u^{k} - t^{k}\| \to 0} \left| \sum_{j=1}^{p} \sum_{i=1}^{n} \sum_{k=1}^{n} c_{ji} e_{jk} \begin{pmatrix} [(x_{i} + x_{k})^{2} - f_{ijk}(x_{i}, x_{k})] \\ -[x_{i}^{2} - f_{ijk}(x_{i})] - [x_{k}^{2} - f_{ijk}(x_{k})] \end{pmatrix} \right|$$
  
$$\to 0, x \in X^{k}.$$

Based on the above linear under-estimators, every feasible point of the (P) in sub-domain  $X^k$  is feasible in (RLP); and the value of the objective function for the (RLP) is less than or equal to that of the (P) for all points in  $X^k$ . Thus, the (RLP) provides a valid lower bound for the solution of the (P) over the partition set  $X^k$ . It should be noted that problem (RLP) contains only the necessary constraints to guarantee convergence of the algorithm.

#### **III ACCELERATING TECHNIQUE**

In this following, we propose an accelerating method for global optimization algorithm of problem (P) using a suitable deleting technique. This technique offers a possibility to cut away a large part of the currently investigated region in which the globally optimal solution of the (P) does not exist, and can be seen as an accelerating device for global optimization algorithm of problem (P). We can give the following accelerating theorem. Since  $\varphi(x)$  and  $\varphi^{u}(x)$  are all linear functions over  $X = [\underline{x}_{i}, \overline{x}_{i}]_{n \times 1} \subseteq X^{0}$ , in the following, for convenience of expression, without loss of generality, we let

$$\begin{split} \varphi(x) &= \sum_{i=1}^{n} \underline{c}_{i} x_{i} + \underline{c}_{0}, \\ \varphi^{ll} &= \sum_{i=1}^{n} \min\{\underline{c}_{i} \overline{x}_{i}, \underline{c}_{i} \underline{x}_{i}\} + \underline{c}_{0}, \\ s_{i} &= \frac{UB - \varphi^{ll} + \min\{\underline{c}_{i} \overline{x}_{i}, \underline{c}_{i} \underline{x}_{i}\}}{\underline{c}_{i}}, \quad \text{with } \underline{c}_{i} \neq 0, \end{split}$$

where i = 1, ..., n.

**Theorem 2. (Ref. [23]).** For any  $X = [\underline{x}_i, \overline{x}_i]_{n \times 1} \subseteq X^0$ , the following conclusions holds:

(i) If  $\varphi^{ll} > UB$ , then there exists no optimal solution of problem (P) over X;

(ii) If  $\varphi^{ll} \leq UB$ , then:

If there exists some index  $v \in \{1,...,n\}$  satisfying  $\underline{c}_{v} > 0$  and  $s_{v} < \overline{x}_{v}$ , then there is no optimal solution of problem (P) over  $X_{1}$ ;

Conversely, if  $\underline{c}_{v} < 0$  and  $s_{v} > \underline{x}_{v}$  for some index  $v \in \{1, ..., n\}$ , then there does exit optimal solution of problem (P) over  $X_{2}$ ;

where

with

$$[\underline{x}_{1i}, \overline{x}_{1i}] = \begin{cases} [\underline{x}_i, \overline{x}_i], & \text{if } i \neq v, \\ (s_v, \overline{x}_v] \cap [\underline{x}_v, \overline{x}_v], & \text{if } i = v, \end{cases}$$
$$X_2 = [\underline{x}_{2i}, \overline{x}_{2i}]_{n \times 1} \subseteq X.$$

 $X_1 = [\underline{x}_{1i}, \overline{x}_{1i}]_{n \times 1} \subseteq X$ 

with

$$[\underline{x}_{2i}, \overline{x}_{2i}] = \begin{cases} [\underline{x}_i, \overline{x}_i], & \text{if } i \neq v, \\ [\underline{x}_v, s_v) \cap [\underline{x}_v, \overline{x}_v], & \text{if } i = v, \end{cases}$$

Proof. The proof of the theorem is omitted.

#### IV ALGORITHM AND ITS CONVERGENCE

In this section a branch and bound algorithm is proposed to globally solve the (P) based on the former linear relaxation method. This algorithm needs to solve a sequence of relaxation linear programming over the initial rectangle  $X^0$  or partitioned sub-rectangle  $X^k$  in order to find a global optimum solution.

The branch and bound approach is based on partitioning the set  $X^0$  into sub-hyper-rectangles, each concerned with a node of the branch and bound tree, and each node is associated with a relaxation linear sub-problem in each sub-hyper-rectangle. Hence, at any stage k of the algorithm, suppose that we have a collection of active nodes denoted by  $\Omega_k$ , say, each associated with a hyper-rectangle

$$X \subseteq X^0, \quad \forall X \in \Omega_k$$

For each such node X, we will have computed a lower bound of the optimal value of the (P) via the solution LB(X) of the (RLP), so that the lower bound of optimal value of the (P) on the whole initial box region  $X^0$  at stage k is given by

$$LB_{k} = \min\{LB(X), \forall X \in \Omega_{k}\}.$$

Whenever the lower bounding solution for any node subproblem, i.e., the solution of the relaxation linear programming (RLP) turns out to be feasible to the (P), we update the upper bound of incumbent solution *UB* if necessary. Then, the active nodes collection  $\Omega_k$  will satisfy

$$LB(X) < UB, \forall X \in \Omega_k,$$

for each stage k. We now select an active node to partition its associated hyper-rectangle into two subhyper-rectangles as described below, computing the lower bounds for each new node as before. Upon fathoming any non-improving nodes, we obtain a collection of active nodes for the next stage, and this process is repeated until convergence is obtained.

Let  $LB(X^k)$  refer to the optimal objective function value of (RLP) for the sub-hyper-rectangles  $X^k$  and  $x^k = x(X^k)$  refer to an element of corresponding argmin. The basic steps of the proposed algorithm are summarized as follows.

### Algorithm statement Step 0. (Initialization)

Initialize the iteration counter k := 0; convergence tolerance  $\varepsilon > 0$ ; the set of all active node  $\Omega_0 = \{X^0\}$ ; the upper bound  $UB = \infty$ , and the set of feasible points  $F := \emptyset$ .

Solve the (RLP) for  $X = X^0$ , obtaining  $LB_0 := LB(X)$ 

and

If  $x^0$  is feasible to the (RLP) update F and UB, if necessary. If

 $x^0 \coloneqq x(X)$ .

$$UB \leq LB_0 + \varepsilon$$
,

then stop with  $x^0$  as the prescribed solution to the (P). Otherwise, proceed to Step 1.

Step 1. (Updating upper bound)

Select the midpoint  $x^m$  of  $X^k$ , if  $x^m$  is feasible to the (P) then

$$F \coloneqq F \bigcup \{x^m\}$$

Define the upper bound

$$UB := \min_{x \in F} \varphi(x)$$
.

If  $F \neq \emptyset$ , the best known feasible point is denoted by  $h = \arg \min_{x \in Y} - g(x)$ 

$$v := \arg \min_{x \in F} \varphi(x)$$

For the investigated sub-rectangle  $X^k$ , we using deleting technique to deleting a part of  $X^k$ , denote the remaining as  $X^k$ .

#### Step 2. (Accelerating)

For the investigated sub-rectangle  $X^k$ , we can use accelerating technique to delete a part of  $X^k$ , denote the remaining as  $X^k$ .

#### Step 3. (Branching)

Choose a branching variable  $x_q$  to partition  $X^k$  to get two new sub-hyper-rectangles according to the above selected branching rule. Call the set of new partition rectangles as  $\overline{X}^k$ .

For each  $X \in \overline{X}^k$ , calculate the lower bound  $\varphi^L$  of  $\varphi(x)$  over the rectangle X, i.e.,

 $\varphi^L \coloneqq \min_{x \in X} \varphi(x).$ 

If

$$\varphi^L > UB$$
,

then remove the corresponding sub-rectangle X from  $\overline{X}^k$ , i.e.

$$\overline{X}^k \coloneqq \overline{X}^k \setminus X$$

and skip to next element of  $\overline{X}^k$ .

If  $\overline{X}^k \neq \emptyset$ , solve the (RLP) to obtain LB(X) and x(X) for each  $X \in \overline{X}^k$ . If

$$LB(X) > UB,$$

set

$$\overline{X}^k := \overline{X}^k \setminus X$$
;

otherwise, update the best available solution UB, F and b if possible, as in Step 1.

Step 4. (Updating lower bound)

The partition set remaining is now

$$\Omega_{\iota} \coloneqq (\Omega_{\iota} \setminus X^{k}) \bigcup \overline{X}$$

giving a new lower bound

$$LB_k := \inf_{X \in \Omega_k} LB(X)$$

Step 5. (Fathoming)

Fathom any non-improving nodes by setting

$$\Omega_{k+1} = \Omega_k \setminus \{ X : UB - LB(X) \le \varepsilon, X \in \Omega_k \},\$$

If

$$\Omega_{k+1} = \emptyset$$
,

then stop with *UB* is an optimal solution. Otherwise, k := k + 1, and select an active node  $X^k$  such that

$$X^{k} := \arg\min_{X \in \Omega_{k}} LB(X), x^{k} := x(X^{k}),$$

and return to Step 1.

**Theorem 3.** (convergence result). The above algorithm either terminates finitely with the incumbent solution being optimal to the (P), or generates an infinite sequence of iteration such that along any infinite branch of the branch and bound tree, any accumulation point of the sequence  $\{x^k\}$  will be the global solution of the problem (P), i.e.

Proof.

$$LB = \lim_{k \to \infty} LB_k = \min_{x \in D} f(x).$$

A sufficient condition for a global optimization to be convergent to the global minimum, stated in Horst and Tuy [27] requires that the bounding operation must be consistent and the selection operation bound improving.

A bounding operation is called consistent if at every step any unfathomed partition can be further refined, and if any infinitely decreasing sequence of successively refined partition elements satisfies:

## $\lim_{k \to \infty} (UB - LB_k) = 0.$

where LB (s) is a computed lower bound in stage s and UB is the best upper bound at iteration s not necessarily occurring inside the same sub-rectangle with LB (s). In the following we will demonstrate the above formulation holds.

Since the employed subdivision process is the bisection, the process is exhaustive. Consequently, from Lemma 1 and Theorem 2 and the discussion in [27] the formulation holds, and then it means that the employed bounding operation is consistent.

A selection operation is called bound improving if at least one partition element where the actual lower bound is attained is selected for further partition after a finite number of refinements. Clearly, the employed selection operation is bound improving because the partition element where the actual lower bound is attained is selected for further partition in the immediately following iteration.

In summary, we have shown that the bounding operation is consistent and that the selection operation is bound improving, therefore according to Theorem.

#### V. NUMERICAL EXPERIMENT

To verify performance of the proposed algorithm, the algorithm is coded in C++ language on Pentium IV (433 MHZ) microcomputer and each linear programming is solved by simplex method, and the convergence tolerance  $\varepsilon$  is set to  $10^{-8}$  in our experiment.

Example 1.

$$\begin{array}{ll} \min & G_0(x) = (x_1 + x_2 + x_3)(2x_1 + x_2 + x_3) \\ s.t. & 1 \le x_1 \le 3, \\ & 1 \le x_2 \le 3.5, \\ & 1 \le x_3 \le 3. \end{array}$$

Using the above proposed algorithm we can globally solve the example 1 on microcomputer, the result is given as follows. Numerical results of the example 1 are optimal value v = 12.

Example 2.  $(min - C_1(x) - (x + x + 1.5x)(2x + 2x + x))$ 

$$\begin{array}{l} \min \quad G_0(x) = (x_1 + x_2 + 1.5x_3)(2x_1 + 2x_2 + x_3) \\ s.t. \quad 1 \le x_1 \le 3, \\ 1 \le x_2 \le 3.5, \\ 1 \le x_2 \le 3. \end{array}$$

Using the above proposed algorithm we can globally solve the example 2 on microcomputer. Numerical results of the example 2 are optimal value v = 17.5.

Example 3.

$$\begin{cases} \min & G_0(x) = (2x_1 + x_2 + x_3)(2x_1 + 2x_2 + x_3) \\ s.t. & 1 \le x_1 \le 2.5, \\ & 1 \le x_2 \le 3.5, \\ & 1 \le x_2 \le 3.5. \end{cases}$$

Using the above proposed algorithm we can globally solve the Example 3 on microcomputer. Numerical results of the example 3 are optimal value v = 20.

Example 4.

$$\begin{cases} \min & G_0(x) = (2x_1 + 1.5x_2 + x_3)(2x_1 + 2x_2 + x_3) \\ st. & 1 \le x_1 \le 2, \\ & 1 \le x_2 \le 2, \\ & 1 \le x_3 \le 2. \end{cases}$$

Using the above proposed algorithm we can globally solve the Example 4 on microcomputer. Numerical results of the example 4 are optimal value v = 27.5.

From numerical experiments, it is seen that our algorithm can globally solve the problem (P) effectively.

### VI. CONCLUDING REMARKS

In this paper, a global optimization algorithm is presented for a class of linear multiplicative programming with linear constraints. By problems utilizing linearization technique, a linear relaxation programming of the (P) is then obtained based on the linear lower bounding of the objective function. The algorithm was shown to attain finite  $\varepsilon$  convergence to the global minimum through the successive refinement of a linear relaxation of the feasible region and the subsequent solution of a series of linear programming problems. The proposed approach was applied to several test problems. In all cases, convergence to the global minimum was achieved. The numerical results are given to illustrate the feasibility and the robust stability of the present algorithm.

#### ACKNOWLEDGMENT

This paper is supported by the National Natural Science Foundation of Henan Province of China, Natural Science Research Foundation of Henan Institute of Science and Technology.

The work was also supported by Foundation for University Key Teacher by the Ministry of Education of Henan Province and the Natural Science Foundation of Henan Educational Committee (2010B110010).

#### REFERENCES

- G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions", *Phil. Trans. Roy. Soc. London*, vol. A247, pp. 529–551, April 1955.
- [2] I. Quesada and I.E. Grossmann, Alternative bounding approximations for the global optimization of various engineering design problems. In I.E. Grossmann, (ed.), Global Optimization in Engineering Design, Vol. 9 Nonconvex Optimization and Its Applications, Kluwer Academic Publishers, Norwell, MA, pp. 309-331, 1996.
- [3] J.M. Mulvey, R.J. Vanderbei and S.A. Zenios, Robust optimization of large-scale systems. Operations Research, 43, pp. 264-281, 1995.
- [4] T. Kuno, Y. Yajima and H. Konno, An outer approximation method for minimizing the product of several convex functions on a convex set. Journal of Global optimization, 3 (3), pp. 325-335, 1993.
- [5] H.P. Benson, Decomposition branch and bound based algorithm for linear programs with additional multiplicative constraints. Journal of Optimization Theory and Applications, 126 (1), pp. 41-46, 2005.

- [6] T. Kuno, A finite branch and bound algorithm for linear multiplicative programming. Computational Optimization and Application, 20, pp. 119-135, 2001.
- [7] H.S. Ryoo and N. V. Sahinidis, Global Optimization of Multiplicative Programs. Journal of Global Optimization, 26, pp. 387-418, 2003.
- [8] S. Schaible and C. Sodini, Finite algorithm for generalized linear multiplicative programming. Journal of Optimization Theory and Applications, 87 (2), pp. 441-455, 1995.
- [9] H.P. Benson and G.M. Boger, Outcome-space cuttingplane algorithm for linear multiplicative programming. Journal of Optimization Theory and Applications, 104 (2), pp. 301-322, 2000.
- [10] X.J. Liu, T. Umegaki, and Y. Yamamoto, Heuristic methods for linear multiplicative programming. Journal of Global Optimization, 4 (15), pp. 433-447, 1999.
- [11] H.-M. Li, K.-C. Zhang, A decomposition algorithm for solving large-scale quadratic programming problems, Applied Mathematics and Computation, 173 (1), pp. 394-403, 2006.
- [12] H. Wu, K. Zhang, A new accelerating method for global non-convex quadratic optimization with non-convex quadratic constraints, Applied Mathematics and Computation, 197 (2), pp. 810-818, 2008.
- [13] S.-T. Liu, R.-T. Wang, A numerical solution method to interval quadratic programming, Applied Mathematics and Computation, 189 (2), pp. 1274-1281, 2007.
- [14] P. Shen, M. Gu, A duality-bounds algorithm for nonconvex quadratic programs with additional multiplicative constraints, Applied Mathematics and Computation, 198 (1), pp. 1-11, 2008.
- [15] P. Shen, Y. Duan, Y. Ma, A robust solution approach for nonconvex quadratic programs with additional multiplicative constraints, Applied Mathematics and Computation, 201 (1-2), pp. 514-526, 2008.
- [16] C. Wang, et al. An Acelerating Algorithm for a Class of Multiplicative Programming, Mathematica Applicata, 24 (4), pp. 13-15, 2011.
- [17] C. Xue, H. Jiao, et al., An approximate algorithm for solving generalized linear multiplicative programming, Journal of Henan Normal University (Natural Science), 36 (5), pp. 13-15, 2008.
- [18] Y. Gao, C. Xu, Yongjian Yang, An outcome-space finite algorithm for solving linear multiplicative programming, Applied Mathematics and Computation, 179 (2), 494-505, 2006.
- [19] R. M. Oliveira, and P. A. V. Ferreira, An Outcome Space Approach for Generalized Convex Multiplicative Programs, Journal of Global Optimization, 47, pp. 107-118, 2010.
- [20] A. M. Ashtiani, P. A. V. Ferreira, Global Maximization of a Generalized Concave Multiplicative Problem in the Outcome Space, Anais do CNMAC, 3, pp. 377-383, 2010.
- [21] Y. Chen, H. Jiao, A nonisolated optimal solution of general linear multiplicative programming problems, Computers & Operations Research, 36, pp. 2573-2579, 2009.
- [22] W. Chun-Feng, L. San-Yang, S. Pei-Ping, Global minimization of a generalized linear multiplicative programming, Appl. Math. Modelling, 36 (6), pp. 2446-2451, 2012.
- [23] P. Shen, H. Jiao, Linearization method for a class of multiplicative programming with exponent, Applied Mathematics and Computation, 183 (1), pp. 328-336, 2006.
- [24] C.-F. Wang, S.-Y. Liu, A new linearization method for generalized linear multiplicative programming, Computers & Operations Research, 38, pp. 1008-1013, 2011.
- [25] H. Jiao, Y.R. Guo, P. Shen, Global optimization of generalized linear fractional programming with nonlinear

constraints, Applied Mathematics and Computation, 183 (2), pp.717-728, 2006.

- [26] H. Jiao, A branch and bound algorithm for globally solving a class of nonconvex programming problems, Nonlinear Analysis: Theory, Methods \& Applications, 70, pp. 1113-1123, 2009.
- [27] P. Shen, X. Bai, W. Li, A new accelerating method for globally solving a class of nonconvex programming problems, Nonlinear Analysis: Theory, Methods & Applications, 71 (7-8), pp. 2866-2876, 2009.
- [28] H. P. Benson, Global Maximization of a Generalized Concave Multiplicative Function, Journal of Optimization Theory and Application, 137, pp. 105-120, 2008.
- [29] H. Mao, Q. Feng, A kind of computational method for solving a class of fractional problems using deleting technique, Journal of Computational Information Systems, 6 (4), pp. 1243-1250, 2010.

**Jingben Yin**, (1970-), male, is a associate professor at Department of Mathematics, Henan Institute of Science and Technology, China. He received the Master Degree from ZhengZhou University in 2009.

His research interests include software engineering, computer application, optimization algorithm design, product design, manufacturing information systems, optimization algorithm, nonlinear system, optimal control theory. He has published over 30 research monographs. Corresponding Email: jingbenyin@163.com

**Yutang Liu**, is a teacher at a teacher of Henan Mechanical and Electrical Engineering College.

His research interests include computational algorithm, numerical algorithm, software engineering, computer application, optimization algorithm design, product design, manufacturing information systems, optimization algorithm, nonlinear system, optimal control theory.

**Baolin Ma**, male, MSc, Lecturer, Was born in February 1978, School of Mathematical Sciences, Henan Institute of Science and Technology, Work at optimization theory, graph theory.

His research interests include global optimization algorithm, software engineering, computer application, optimization design, product design, manufacturing information systems, optimization algorithm, nonlinear system control theory.

**Dongwei Shi**, male, MSc, Lecturer, Was born in 1976, School of Mathematical Sciences, Henan Institute of Science and Technology, Work at optimization theory, graph theory.

His research interests include numerical computation, global optimization algorithm, software engineering, computer application, optimization design, product design, manufacturing information systems, optimization algorithm, nonlinear system control theory.