IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 27, 2020, accepted April 20, 2020, date of publication April 27, 2020, date of current version May 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990677

An Efficient Outer Space Algorithm for Generalized Linear Multiplicative Programming Problem

HONGWEI JIAO[1,2], WENJIE WANG[2], RONGJIANG CHEN[2], YOULIN SHANG[1], AND JINGBEN YIN[2]

[1]Postdoctoral Research Station of Control Science and Engineering, Henan University of Science and Technology, Luoyang 471023, China
[2]Postdoctoral Research Base, Henan Institute of Science and Technology, Xinxiang 453003, China

Corresponding authors: Hongwei Jiao (jiaohongwei@hist.edu.cn), Rongjiang Chen (chenrj0324@126.com), and Youlin Shang (mathshang@sina.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 11871196, in part by the China Postdoctoral Science Foundation under Grant 2017M622340, in part by the Key Scientific and Technological Research Projects of Henan Province under Grant 202102210147 and Grant 192102210114, and in part by the Science and Technology Climbing Program of Henan Institute of Science and Technology under Grant 2018JY01.

ABSTRACT Base on the outer space search and the branch-and-bound framework, this paper presents an efficient outer space branch-and-bound algorithm for globally solving generalized linear multiplicative programming problem. First of all, we convert the problem into an equivalent problem. Then, by utilizing a direct relaxation method, we establish the linear relaxed problem to compute the lower bound of the global optimal value of the equivalent problem. By subsequently subdividing the initial outer space rectangle and solving a series of linear relaxed problems, the proposed algorithm is convergent to the global optimal solution of the primal problem. Finally, compared with some known algorithms, numerical experiments are given to demonstrate the feasibility and effectiveness of the proposed algorithm.

INDEX TERMS Generalized linear multiplicative programming problem, global optimization, linear relaxed problem, branch-and-bound algorithm.

I. INTRODUCTION

We investigate the following generalized linear multiplicative programming problem:

$$(\mathrm{MP}): \begin{cases} \min f(x) = \sum_{i=1}^{p}(\sum_{j=1}^{n} c_{ij}x_j + e_i)(\sum_{j=1}^{n} d_{ij}x_j + g_i) \\ s.t.\ x \in X = \{x \mid Ax \le b,\ x \ge 0\}, \end{cases}$$

where $p \ge 2$, $\sum_{j=1}^{n} c_{ij}x_j + e_i$ and $\sum_{j=1}^{n} d_{ij}x_j + g_i$, $i = 1, \ldots, p$, are all bounded linear (affine) functions defined on X, $A \in R^{m\times n}$, $b \in R^m$, X is a bounded polyhedron set. Because the problem (MP) have a wide of applications in microeconomics [1], data mining/pattern recognition [2], plant layout design [3], system reliability analysis and optimization [4]–[10], optimal resource allocation for Power-Efficient MC-NOMA [11] and other fields [12]–[14], and it has potential theoretical and computational difficulty. So that, it has attracted the attentions of many researchers and practitioners.

In the past few decades, a large number of algorithms have been proposed for the problem (MP) and its special form. Most of these algorithms are based on the branch and bound framework, for examples, by utilizing convex envelope and concave envelope of logarithmic function to construct linear relaxation, Shen and Jiao [15] proposed a branch-and-bound algorithm for linear multiplicative programming problem with exponents; by utilizing two-level linear relaxation to construct linear relaxed problem, Jiao [16] proposed a branch-and-bound algorithm for generalized linear multiplicative programming problem of generalized polynomial forms; based on partitioning and searching outcome space region, Gao *et al.* [17] presented an outcome space branch-and-bound algorithm for the special form of the problem (MP); based on utilizing the convex envelope of bilinear function, and by successively subdividing and

The associate editor coordinating the review of this manuscript and approving it for publication was Khalid Aamir.

searching the outcome space rectangle of the affine functions $\sum_{j=1}^{n} d_{ij}x_j + g_i, i = 1, \ldots, p$, Yin *et al.* [18] proposed a new outcome space branch-and-bound algorithm for solving the general form of problem (MP); by using new linearizing technique, Wang *et al.* [19] presented a novel algorithm for the problem (MP); based on simplex partition and search, wang *et al.* [20] proposed a simplex branch-and-bound algorithm for the problem (MP). In addition to the branch and bound algorithm mentioned above, some other types of algorithms are also proposed to solve the problem (MP), such as, Phuong and Tuy [21] and Chen and Jiao [22] proposed two monotonic optimization methods for generalized linear fractional programming problem and generalized linear multiplicative programming problems; by utilizing the concept of level set, Liu and Zhao [10] gave a level set method for solving the linear multiplicative programming problem with exponents. Recently, Shen *et al.* [23]–[25] proposed three polynomial time approximation algorithms for generalized linear fractional multiplicative programming problem; Zhao and Zhao [12] presented a convex relaxation method for generalized linear multiplicative programming problem with generalized linear multiplicative constraints; based on geometric programming solver, Zhao and Yang [13] also proposed an inner approximation algorithm for generalized linear multiplicative programming problem with generalized linear multiplicative constraints; Jiao *et al.* [5] formulated an outcome space method for generalized linear multiplicative programming problem with linear constraints; these methods can be also used to solve the problem (MP) considered in this paper.

But as far as we know, apart from [18], most of the algorithms mentioned above require that $\sum_{j=1}^{n} c_{ij}x_j + e_i \geq 0$ and $\sum_{j=1}^{n} d_{ij}x_j + g_i \geq 0$, it is to say, the algorithm for the problem (MP) with that $\sum_{j=1}^{n} c_{ij}x_j + e_i$ and $\sum_{j=1}^{n} d_{ij}x_j + g_i$ can acquire arbitrary values has been still little studied in the literatures. Thus, it is still very necessary to propose a new practical algorithm for the general form of the problem (MP).

This paper will present a new outer space branch-and-bound algorithm for globally solving generalized linear multiplicative programming problem (MP). First of all, we convert the problem (MP) into an equivalent nonconvex programming problem (EQ). Next, by utilizing the direct linear relaxed technique, we establish the linear relaxed problem of the equivalent problem (EQ). By subsequently partitioning the outer space rectangle of the images of $\sum_{j=1}^{n} d_{ij}x_j + g_i, i = 1, \ldots, p$, and solving a series of linear relaxed problems, the proposed algorithm is convergent to the global minimum of the primal problem (MP). Compared with the existent algorithms, the proposed algorithm have the following some potential practical and computational advantages:

(i) The mathematical modelling of the considered problem has a more general form, which does not impose any special signal restrictions on each linear function $\sum_{j=1}^{n} c_{ij}x_j + e_i$ and $\sum_{j=1}^{n} d_{ij}x_j + g_i, i = 1, \ldots, p$, in the objective function of the problem (MP).

(ii) The branching search take place in the outer space rectangle of the images of $\sum_{j=1}^{n} d_{ij}x_j + g_i, i = 1, \ldots, p$, rather than the rectangle of variable region, which will greatly reduce the dimension of the searched space and decrease the required computational efforts since n usually far exceeds p in many practical problems.

(iii) During the process of the branch-and-bound search, all subproblems that need to be solved in each iteration are all linear programming problems, and the scale of these linear programming problems which can be solved by the known LP solver remains unchanged.

(iv) Numerical experiments show that the proposed algorithm has the stronger robustness, and which can be used to solve all test problems to get their global optimal solutions and global optimal values with the higher computational efficiency.

The remaining section of the paper is organized as follows. Firstly, we convert the problem (MP) into an equivalent problem (EQ) in Section 2, and by utilizing the direct relaxation technique, we establish its linear relaxed problem. Secondly, we propose a new outer space branch-and-bound algorithm for the problem (EQ) in Section 3, and derive its global convergence. Thirdly, some numerical experiments are reported in Section 4, and numerical results demonstrate the stronger robustness and the higher efficiency of the algorithm. Finally, we present some concluding remarks in Section 5 and give some future works.

II. EQUIVALENT PROBLEM AND ITS LINEAR RELAXED PROGRAMMING PROBLEM

In this section, we firstly will convert the problem (MP) into an equivalent problem. Next, we will drive the linear relaxed problem of the equivalent problem.

Without losing generality, for each $j = 1, \ldots, n$, we assume that $l_j^0 = \min_{x\in X} x_j$ and $u_j^0 = \max_{x\in X} x_j$. And for each $i = 1, \ldots, p$, we let $L_i^0 = \min_{x\in X} \sum_{j=1}^{n} d_{ij}x_j + g_i$ and $U_i^0 = \max_{x\in X} \sum_{j=1}^{n} d_{ij}x_j + g_i$, which are easily computed by solving $2p$ linear programming problems. Let $F^0 = \{\alpha \in R^p \mid L_i^0 \leq \alpha_i \leq U_i^0,\ i = 1, \ldots, p\}$, we consider the following problem:

$$
(\mathrm{EQ}):\begin{cases} \min & G(x,\alpha) = \sum_{i=1}^{p} \alpha_i (\sum_{j=1}^{n} c_{ij}x_j + e_i) \\ s.t. & \sum_{j=1}^{n} d_{ij}x_j + g_i - \alpha_i = 0, i = 1, \ldots, p, \\ & x \in X,\ \alpha \in F^0. \end{cases}
$$

Obviously, the problems (MP) and (EQ) have the same global optimal solutions and global optimal values. In the following, we will present a direct linear relaxation method for constructing the linear relaxed problem of the problem (EQ), which is given by the following Theorem 1.

Theorem 1: Let $F = \{\alpha \in R^p \mid L_i \leq \alpha_i \leq U_i, i = 1, \ldots, p\} \subseteq F^0$, for any $x \in X$ and $\alpha \in F$, define the functions $G(x,\alpha)$, $G_{RC}(x,\alpha)$ and $G^{RC}(x,\alpha)$ as follows:

$$\begin{aligned} G(x,\alpha) &= \sum_{i=1}^{p}\alpha_i(\sum_{j=1}^{n} c_{ij}x_j + e_i),\\ G_{RC}(x,\alpha) &= \sum_{i=1}^{p}(\sum_{j=1,c_{ij}>0}^{n} c_{ij}L_ix_j + \sum_{j=1,c_{ij}<0}^{n} c_{ij}U_ix_j\\ &\quad + e_i\alpha_i),\\ G^{RC}(x,\alpha) &= \sum_{i=1}^{p}(\sum_{j=1,c_{ij}>0}^{n} c_{ij}U_ix_j + \sum_{j=1,c_{ij}<0}^{n} c_{ij}L_ix_j\\ &\quad + e_i\alpha_i). \end{aligned}$$

Then, we have the following conclusions:

(i) For any $x \in X$ and $\alpha \in F$, we have

$$G_{RC}(x,\alpha) \leq G(x,\alpha) \leq G^{RC}(x,\alpha);$$

(ii)

$$\begin{aligned} \lim_{\|U-L\|\to 0} G_{RC}(x,\alpha) &= \lim_{\|U-L\|\to 0} G(x,\alpha)\\ &= \lim_{\|U-L\|\to 0} G^{RC}(x,\alpha). \end{aligned}$$

Proof: (i) Since

$$\begin{aligned} &G(x,\alpha) - G_{RC}(x,\alpha)\\ &= \sum_{i=1}^{p}\alpha_i(\sum_{j=1}^{n} c_{ij}x_j + e_i)\\ &\quad - \sum_{i=1}^{p}(\sum_{j=1,c_{ij}>0}^{n} c_{ij}L_ix_j + \sum_{j=1,c_{ij}<0}^{n} c_{ij}U_ix_j + e_i\alpha_i)\\ &= \sum_{i=1}^{p}(\sum_{j=1}^{n} c_{ij}\alpha_ix_j - \sum_{j=1,c_{ij}>0}^{n} c_{ij}L_ix_j - \sum_{j=1,c_{ij}<0}^{n} c_{ij}U_ix_j)\\ &= \sum_{i=1}^{p}[\sum_{j=1,c_{ij}>0}^{n} c_{ij}(\alpha_i - L_i)x_j + \sum_{j=1,c_{ij}<0}^{n} c_{ij}(\alpha_i - U_i)x_j]\\ &\geq 0 \end{aligned}$$

and

$$\begin{aligned} &G^{RC}(x,\alpha) - G(x,\alpha)\\ &= \sum_{i=1}^{p}(\sum_{j=1,c_{ij}>0}^{n} c_{ij}U_ix_j + \sum_{j=1,c_{ij}<0}^{n} c_{ij}L_ix_j + e_i\alpha_i)\\ &\quad - \sum_{i=1}^{p}\alpha_i(\sum_{j=1}^{n} c_{ij}x_j + e_i)\\ &= \sum_{i=1}^{p}(\sum_{j=1,c_{ij}>0}^{n} c_{ij}U_ix_j + \sum_{j=1,c_{ij}<0}^{n} c_{ij}L_ix_j - \sum_{j=1}^{n} c_{ij}\alpha_ix_j)\\ &= \sum_{i=1}^{p}[\sum_{j=1,c_{ij}>0}^{n} c_{ij}(U_i - \alpha_i)x_j + \sum_{j=1,c_{ij}<0}^{n} c_{ij}(L_i - \alpha_i)x_j]\\ &\geq 0, \end{aligned}$$

from the above inequalities, we can get that, for any $x \in X$ and $\alpha \in F$,

$$G_{RC}(x,\alpha) \leq G(x,\alpha) \leq G^{RC}(x,\alpha).$$

(ii) since

$$\begin{aligned} &G(x,\alpha) - G_{RC}(x,\alpha)\\ &= \sum_{i=1}^{p}[\sum_{j=1,c_{ij}>0}^{n} c_{ij}(\alpha_i - L_i)x_j + \sum_{j=1,c_{ij}<0}^{n} c_{ij}(\alpha_i - U_i)x_j]\\ &\leq \sum_{i=1}^{p}[\sum_{j=1,c_{ij}>0}^{n} c_{ij}(U_i - L_i)x_j - \sum_{j=1,c_{ij}<0}^{n} c_{ij}(U_i - L_i)x_j]\\ &= \sum_{i=1}^{p}(U_i - L_i)(\sum_{j=1,c_{ij}>0}^{n} c_{ij}x_j - \sum_{j=1,c_{ij}<0}^{n} c_{ij}x_j)\\ &\leq \sum_{i=1}^{p}(U_i - L_i)(\sum_{j=1,c_{ij}>0}^{n} c_{ij}u_j - \sum_{j=1,c_{ij}<0}^{n} c_{ij}l_j) \end{aligned}$$

and

$$\begin{aligned} &G^{RC}(x,\alpha) - G(x,\alpha)\\ &= \sum_{i=1}^{p}[\sum_{j=1,c_{ij}>0}^{n} c_{ij}(U_i - \alpha_i)x_j + \sum_{j=1,c_{ij}<0}^{n} c_{ij}(L_i - \alpha_i)x_j]\\ &\leq \sum_{i=1}^{p}[\sum_{j=1,c_{ij}>0}^{n} c_{ij}(U_i - L_i)x_j - \sum_{j=1,c_{ij}<0}^{n} c_{ij}(U_i - L_i)x_j]\\ &= \sum_{i=1}^{p}(U_i - L_i)(\sum_{j=1,c_{ij}>0}^{n} c_{ij}x_j - \sum_{j=1,c_{ij}<0}^{n} c_{ij}x_j)\\ &\leq \sum_{i=1}^{p}(U_i - L_i)(\sum_{j=1,c_{ij}>0}^{n} c_{ij}u_j - \sum_{j=1,c_{ij}<0}^{n} c_{ij}l_j), \end{aligned}$$

this implies that

$$\begin{aligned} \lim_{\|U-L\|\to 0} G_{RC}(x,\alpha) &= \lim_{\|U-L\|\to 0} G(x,\alpha)\\ &= \lim_{\|U-L\|\to 0} G^{RC}(x,\alpha). \end{aligned}$$

and the proof is completed. ◇

Let $F = \{\alpha \in R^p \mid L_i \le \alpha_i \le U_i, i = 1, \ldots, p\}$ be F^0 or its a sub-rectangle, by Theorem 1, we can establish the linear relaxed problem (LP_F) of the problem (EQ) over F as follows:

$$(LP_F): \begin{cases} \min \ G_{RC}(x,\alpha) = \sum_{i=1}^{p} (\sum_{j=1, c_{ij}>0}^{n} c_{ij} L_i x_j \\ \quad + \sum_{j=1, c_{ij}<0}^{n} c_{ij} U_i x_j + e_i \alpha_i) \\ s.t. \ \sum_{j=1}^{n} d_{ij} x_j + g_i - \alpha_i = 0, \quad i = 1, \ldots, p, \\ \quad x \in X, \ \alpha \in F. \end{cases}$$

By the constructing method for the linear relaxed problem, over the same rectangle F, it is obvious that the feasible region of the problem (LP_F) contain the feasible region of the problem (EQ), and the optimal value of the problem (LP_F) is less than or equal to that of the problem (EQ). That is to say, the problem (LP_F) can provide a valid lower bound for the problem (EQ) over the same rectangle F.

III. OUTER SPACE ALGORITHM AND ITS CONVERGENCE

In this section, we firstly introduce a rectangular bisection method, then based on the branch-and-bound framework, combine the rectangular bisection method and the former linear relaxed problem together, a new outer space branch-and-bound algorithm is proposed for globally solving the problem (MP).

A. RECTANGULAR BISECTION METHOD

In the algorithm, the branching process take place in the outer space rectangle of the affine functions $\sum_{j=1}^{n} d_{ij} x_j + g_i, i = 1, \ldots, p$. Without loss of generality, let

$$F^{k-1} = \{\alpha \in R^p | L_i \le \alpha_i \le U_i, i = 1, \ldots, p\}$$

be F^0 or a sub-rectangle of F^0, which is selected for the next segmentation, the proposed rectangular bisection method is described as follows. Let

$$q = \arg\max\{U_i - L_i | i = 1, 2, \ldots, p\}$$

be subscript of maximum edge of rectangle F^{k-1}, partition F^{k-1} into two new sub-rectangles

$$F^{k,1} = \{\alpha \in R^p | L_i \le \alpha_i \le \frac{L_i + U_i}{2}, i = q; \\ L_i \le \alpha_i \le U_i, \quad i = 1, 2, \ldots, p, i \ne q\}$$

and

$$F^{k,2} = \{\alpha \in R^p | \frac{L_i + U_i}{2} \le \alpha_i \le U_i, i = q; \\ L_i \le \alpha_i \le U_i, \quad i = 1, 2, \ldots, p, i \ne q\}.$$

Without loss of generality, assume $\{F^k\}$ be a nested rectangular subsequence which be formed by the rectangular partitioning process. Obviously, by the exhaustiveness of the rectangular bisection method, we can get that $\lim_{k\to\infty} F^k = \alpha^*$.

B. OUTER SPACE BRANCH-AND-BOUND ALGORITHM

Combine the above rectangular bisection method and the former linear relaxed problem together, a new outer space branch-and-bound algorithm is proposed for solving the problem (MP) as follows.

Algorithm Steps

Step 1. Let the initial iteration number $k = 0$ and the termination error condition $\varepsilon \ge 0$, by solving linear programming problems, for each $i = 1, 2, \ldots, p$, compute

$$L_i^0 = \min_{x \in X} \sum_{j=1}^{n} d_{ij} x_j + g_i$$

and

$$U_i^0 = \max_{x \in X} \sum_{j=1}^{n} d_{ij} x_j + g_i,$$

let the initial rectangle

$$F^0 = \{\alpha \in R^p \mid L_i^0 \le \alpha_i \le U_i^0, \ i = 1, \ldots, p\},$$

and let the initial active node set $\Upsilon_0 = \{F^0\}$.
Solve the problem (LP_{F^0}) to obtain its optimal solution (x^0, α^0) and optimal value $LB(F^0)$, respectively. Let the lower bound $LB_0 = LB(F^0)$ and the upper bound $UB_0 = G(x^0, \alpha^0)$.
If $UB_0 - LB_0 \le \varepsilon$, then the algorithm terminates, and obtain the optimal solution (x^0, α^0) of the (EQ) and the optimal solution x^0 of the (MP) over F^0, respectively.
Otherwise, let the deleted rectangle set $\Omega = \emptyset$, $k = 1$, and go on Step 2.

Step 2. Let $UB_k = UB_{k-1}$, partition F^{k-1} into two new sub-rectangles $F^{k,1}$ and $F^{k,2}$, let $\bar{F}^k$ be the new set of the partitioned sub-rectangles, and let $\Omega = \Omega \cup \{F^{k-1}\}$ be the set of the deleted rectangle.
For each rectangle $F^{k,t} \in \bar{F}^k$, solve the problem $(LP_{F^{k,t}})$ to obtain its optimal solution $(x^{k,t}, \alpha^{k,t})$ and optimal value $LB(F^{k,t})$. If $LB(F^{k,t}) > UB_k$, then let $\bar{F}^k := \bar{F}^k \setminus F^{k,t}$ and $\Omega = \Omega \cup \{F^{k,t}\}$. Otherwise, let $UB_k = \min\{UB_k, G(x^{k,t}, \alpha^{k,t})\}$ be the new upper bound.

Step 3. Let $\Upsilon_k = (\Upsilon_{k-1} \setminus F^{k-1}) \cup \{\bar{F}^k\}$ be the remaining partitioned rectangle set, and let $LB_k = \max_{F \in \Upsilon_k} LB(F)$ be the new lower bound.

Step 4. If $UB_k - LB_k \le \varepsilon$, then the algorithm stops, and we get the optimal solution (x^k, α^k) of the (EQ) and the optimal solution x^k of the (MP), respectively. Otherwise, let $k = k + 1$ and return to Step 2.

C. CONVERGENCE OF OUTER SPACE ALGORITHM

In the subsection, the global convergence of the outer space branch-and-bound algorithm is given as follows.

If the algorithm terminates going through k iterations, then, at the k_{th} iteration, we can get the feasible solution x^k of the problem (MP) and the feasible solution (x^k, α^k) of the problem (EQ) by solving the (LP_{F^k}), where

$$\alpha_i^k = \sum_{j=1}^{n} d_{ij}x_j^k + g_i, \quad i = 1, 2, \ldots, p.$$

Let v be the global optimal value of the primal problem (MP). From the convergent terminating condition, the updating upper bound methods, the updating lower bound methods, the equivalence of the problem (MP) and (EQ) and Theorem 1, we can obtain that

$$G(x^k, \alpha^k) \leq LB_k + \varepsilon, \ LB_k \leq v, \ v \leq G(x^k, \alpha^k)$$

and

$$f(x^k) = G(x^k, \alpha^k).$$

Combining the several inequalities together, we can follow that

$$v \leq f(x^k) \leq v + \epsilon.$$

Therefore, if the proposed algorithm terminates after going through k iterations, then x^k is an $\epsilon-$global optimal solution of the problem (MP).

Theorem 2: If the proposed algorithm generates an infinite sequence $\{x^k\}$ of solution, then the limitation x^* of $\{x^k\}$ will be a global optimal solution of the problem (MP).

Proof: If the proposed algorithm generates an infinite sequence of solution, then, by solving the linear relaxed programming problem (LP_{F^k}), we can get a feasible solution sequence $\{x^k\}$ of the problem (MP) and a feasible solution sequence $\{(x^k, \alpha^k)\}$ of the problem (EQ), where

$$\alpha_i^k = \sum_{j=1}^{n} d_{ij}x_j^k + g_i, \quad i = 1, 2, \ldots, p.$$

From the continuity of the linear function $\alpha_i = \sum_{j=1}^{n} d_{ij}x_j + g_i$, $\alpha_i^k = (\sum_{j=1}^{n} d_{ij}x_j^k + g_i) \in [L_i^k, U_i^k]$ and the exhaustiveness of the rectangular bisection method, for any $i \in \{1, 2, \ldots, p\}$, we have that

$$\begin{aligned}\sum_{j=1}^{n} d_{ij}x_j^* + g_i &= \lim_{k\to\infty} \sum_{j=1}^{n} d_{ij}x_j^k + g_i \\ &= \lim_{k\to\infty} \bigcap_k [L_i^k, U_i^k] = \alpha_i^*.\end{aligned}$$

So that, (x^*, α^*) is a feasible solution of the problem (EQ), by the Theorem 1, and also since $\{LB_k\}$ is an increasing lower bound sequence with that $LB_k \leq v$, we can get that

$$G(x^*, \alpha^*) \geq v \geq \lim_{k\to\infty} LB_k = \lim_{k\to\infty} G_{RC}(x^k, \alpha^k) = G(x^*, \alpha^*).$$

So, by the continuity of the function $f(x)$ and the above inequalities, we can get that

$$\lim_{k\to\infty} LB_k = v = G(x^*, \alpha^*) = f(x^*) = \lim_{k\to\infty} f(x^k).$$

Therefore, the limitation x^* of $\{x^k\}$ is a global optimal solution for the problem (MP), the proof is completed. ◇

D. THE COMPLEXITY OF THE PROPOSED ALGORITHM

Definition 1: A nonempty compact hyper-rectangle with sides parallel to the axes is denoted by

$$F = [L_1, U_1] \times \ldots \times [L_p, U_p] \subset R^p.$$

The diameter of a hyper-rectangle $F \subset R^p$ is

$$\begin{aligned}\delta(F) &= \max\{\|\alpha - \alpha'\|_2 : \alpha, \alpha' \in F\} \\ &= \sqrt{(U_1 - L_1)^2 + \ldots + (U_p - L_p)^2}.\end{aligned}$$

Theorem 3: For the proposed branch-and-bound algorithm, assumed that, for a given feasible hyper-rectangle F^0, there exists a fixed positive number C and an accuracy ϵ. In addition, we also assume that the branching process will eventually subdivide the hyper-rectangle into $s = 2^p$ smaller sub-hyper-rectangles. Then, in the worst case, the number of iterations of the proposed algorithm by dividing the hyper-rectangle F^0 can be given by the following expression:

$$\sum_{\upsilon=0}^{z} 2^{p.v},$$

where

$$z = \lceil \log_2 \frac{C.\delta(F)}{\epsilon} \rceil, \ \delta(F) = \max\{\delta(F^l) : l \in \{1, 2, \ldots, s\}\}.$$

Proof: The proof method is similar to the proof Theorem 5 in [14].

It can be seen that the number of iterations of the proposed algorithm is exponential growth, but when the size of p satisfies $p \ll n$, our algorithm has an advantage in solving large-scale problem (MP), and the reader can refer to [14].

IV. NUMERICAL EXPERIMENT

To test and verify the computational performance and robustness of the algorithm, with the given convergent error $\epsilon = 1.0e-6$, some numerical problems are coded by Matlab 2014a in a microcomputer with Win10 system, Intel(R) Xeon(R) CPU E5-2620 v4 @2.10GHz processor and 16.0GB RAM memory, these test problems and their numerical results are listed as follows.

Problem 1 (Yin et al. [18]; Wang et al. [20]); Shen and Huang [25]):

$$\begin{cases} \min\ 3x_1 - 4x_2 + (x_1 + 2x_2 - 1.5)(2x_1 - x_2 + 4) \\ \qquad +(x_1 - 2x_2 + 8.5)(2x_1 + x_2 - 1) \\ s.t.\ \ 5x_1 - 8x_2 \geq -24, \\ \qquad 5x_1 + 8x_2 \leq 44, \\ \qquad 6x_1 - 3x_2 \leq 15, \\ \qquad 4x_1 + 5x_2 \geq 10, \\ \qquad x_1 \geq 0. \end{cases}$$

Problem 2 (Yin et al. [18]; Wang et al. [19]); Wang et al. [20]; Shen and Huang [25]):

$$\begin{cases} \min\ (0.813396x_1 + 0.6744x_2 + 0.305038x_3 \\ \quad +0.129742x_4 + 0.217796) \\ \quad \times(0.224508x_1 + 0.063458x_2 + 0.93223x_3 \\ \quad +0.528736x_4 + 0.091947) \\ s.t.\ \ 0.488509x_1 + 0.063458x_2 + 0.945686x_3 \\ \quad +0.210704x_4 \le 3.562809, \\ \quad -0.324014x_1 - 0.501754x_2 - 0.719204x_3 \\ \quad +0.099562x_4 \le -0.052215, \\ \quad 0.445225x_1 - 0.346896x_2 + 0.637939x_3 \\ \quad -0.257623x_4 \le 0.42792, \\ \quad -0.202821x_1 + 0.647361x_2 + 0.920135x_3 \\ \quad -0.983091x_4 \le 0.84095, \\ \quad -0.886420x_1 - 0.802444x_2 - 0.305441x_3 \\ \quad -0.180123x_4 \le -1.353686, \\ \quad -0.515399x_1 - 0.424820x_2 + 0.897498x_3 \\ \quad +0.187268x_4 \le 2.137251, \\ \quad -0.591515x_1 + 0.060581x_2 - 0.427365x_3 \\ \quad +0.579388x_4 \le -0.290987, \\ \quad 0.423524x_1 + 0.940496x_2 - 0.437944x_3 \\ \quad -0.742941x_4 \le 0.37362, \\ \quad x_1, x_2, x_3, x_4 \ge 0. \end{cases}$$

Problem 3 (Gao et al. [17]; Yin et al. [18]; Wang et al. [19]; Wang et al. [20]; Shen and Huang [25]):

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

Problem 4 (Yin et al. [18]; Wang et al. [19]; Wang et al. [20]; Shen and Huang [25]):

$$\begin{cases} \min\ x_1 + (2x_1 - 3x_2 + 13)(x_1 + x_2 - 1) \\ s.t.\ \ -x_1 + 2x_2 \le 8, \\ \quad -x_2 \le -3, \\ \quad x_1 + 2x_2 \le 12, \\ \quad x_1 - 2x_2 \le -5,\ \ x_1, x_2 \ge 0. \end{cases}$$

Problem 5 (Yin et al. [18]; Wang et al. [20]; Shen and Huang [25]):

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

Problem 6 (Yin et al. [18]; Wang et al. [20]); Shen and Huang [25]):

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

Problem 7 (Yin et al. [18]; Wang et al. [19]; Shen and Huang [25]):

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

Problem 8 (Yin et al. [18]; Wang et al. [19]; Shen and Huang [25]):

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

Problem 9 (Yin et al. [18]; Wang et al. [19]; Shen and Huang [25]):

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

Problem 10 (Yin et al. [18]; Wang et al. [19]; Shen and Huang [25]):

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

Problem 11 (Yin et al. [18]):

$$\begin{cases} \min\ \sum_{i=1}^{p}(\sum_{j=1}^{n} c_{ij}x_j + e_i)(\sum_{j=1}^{n} d_{ij}x_j + g_i) \\ s.t.\ \ Ax \le b,\ x \ge 0. \end{cases}$$

where, for each $i = 1, \ldots, p, j = 1, \ldots, n$, c_{ij} and d_{ij} are randomly generated in the interval [0, 1]; for each $i = 1, \ldots, p$, e_i and g_i are randomly generated in the interval [0, 100]; each element of A is randomly generated in the interval [0, 1]; each

TABLE 1. Comparison of numerical results for Problems 1-10.

Problems	Refs.	Optimal solution	Optimal value	Iterations	Time(s)
1	This paper	(0.0000, 3.0000)	-2.50000	182	7.07510
	Yin et al. [18]	(0.0000, 3.0000)	-2.50000	30	1.54421
	Wang et al. [20]	(0, 3)	-2.5	1	0.063
	Shen and Huang [26]	(0, 3)	-2.5	1	0.047
2	This paper	(1.3148,0.1396,0.0000,0.4233)	0.89019	22	0.51458
	Yin et al. [18]	(1.3148,0.1396,0.0000,0.4233)	0.89019	0	0.05622
	Wang et al. [19]	(1.3148, 0.1396, 0, 0.4233)	0.8902	1	0.061
	Wang et al. [20]	(1.3148, 0.1396, 0, 0.4233)	0.8902	1	0.188
	Shen and Huang [26]	(1.3148, 0.1396, 0, 0.4233)	0.8902	1	0.093
3	This paper	(2.0000, 8.0000)	10.00000	27	0.70909
	Gao et al. [17]	(2, 8)	10	53	0.3
	Yin et al. [18]	(2.0000, 8.0000)	10.00000	0	0.04965
	Wang et al. [19]	(2, 8)	10	1	0.046
	Wang et al. [20]	(2, 8)	10	48	5.078
	Shen and Huang [26]	(2, 8)	10	1	0.046
4	This paper	(0.0000, 4.0000)	3.00000	172	3.53071
	Yin et al. [18]	(0.0000, 4.0000)	3.00000	0	0.05613
	Wang et al. [19]	(0, 4)	3	1	0.0727
	Wang et al. [20]	(0, 4)	3	2	0.203
	Shen and Huang [26]	(0, 4)	3	1	0.046
5	This paper	(0.0000, 5.0000)	-233.00000	83	2.14846
	Yin et al. [18]	(0.0000, 5.0000)	-233.00000	1	0.14337
	Wang et al. [20]	(0, 5)	-233	4	0.3232
	Shen and Huang [26]	(0, 5)	-233	1	0.047
6	This paper	(0.0000, 0.0000)	4.00000	15	0.35513
	Yin et al. [18]	(0.0000, 0.0000)	4.00000	9	0.36751
	Wang et al. [20]	(0, 0)	4	7	0.6336
	Shen and Huang [26]	(0, 0)	4	1	0.062
7	This paper	(0.0000, 4.0000)	3.00000	89	2.49274
	Yin et al. [18]	(0.0000, 4.0000)	3.00000	0	0.05496
	Wang et al. [19]	(0, 4)	3	1	0.0693
	Shen and Huang [26]	(0, 4)	3	1	0.062
8	This paper	(1.0000, 3.0000)	-13.00000	50	0.95348
	Yin et al. [18]	(1.0000, 3.0000)	-13.00000	0	0.05962
	Wang et al. [19]	(1, 3)	-13	1	0.0868
	Shen and Huang [20]	(1, 3)	-13	1	0.047
9	This paper	(1.0000, 4.0000)	-22.00000	52	1.02796
	Yin et al. [18]	(1.0000, 4.0000)	-22.00000	0	0.05179
	Wang et al. [19]	(1, 4)	-22	1	0.0849
	Shen and Huang [26]	(1, 4)	-22	1	0.046
10	This paper	(5.5000, 1.0000, 3.5000)	-109.75000	602	19.0745
	Yin et al. [18]	(5.5000, 1.0000, 3.5000)	-109.75000	16	0.50701
	Wang et al. [19]	(5.5556, 1.7778, 2.6667)	-112.754	5	1.1491
	Shen and Huang [26]	(5.5556, 1.7778, 2.6667)	-112.754	1	0.062

element of b is randomly generated in the interval $[0, n]$. Use the proposed algorithm in this paper to solve this problem, numerical results are given in Table 2.

In Table 2, some notations have been used in column headers as follows:

n: Number of variables;

p: Number of sums;

m: Rows of matrix A;

Avg.NT: Average iteration numbers;

Std.NT: Standard deviation of iteration numbers;

Avg.Time: Average CPU time in seconds;

Std.Time: Standard deviation of CPU time.

From numerical results of Table 1, for Problems 1-10, compared with the known algorithms in the literature, the algorithm proposed in this paper can not only obtain the same optimal solution and optimal value, but also they have almost the same higher computational efficiency.

From numerical results of Problem 11 in Table 2, by solving randomly generated large-scale generalized linear multiplicative programming problem, and with the increase of the scale of the Problem 11, our algorithm has the higher computational efficiency than that of Ref. [18].

TABLE 2. Comparison of numerical results for Problem 11.

(p, m, n)	The algorithm of Yin et al.[18] Avg (Std).NT	Avg (Std).Time	Our algorithm in this paper Avg (Std).NT	Avg (Std).Time
(5, 10, 10)	5 (6.446359869)	0.140346 (0.15469412)	0 (0.0)	0.036252 (0.058404039)
(5, 20, 20)	7.6 (8.796463936)	0.236998 (0.275757184)	0 (0.0)	0.016184 (0.009127138)
(5, 30, 30)	21.6 (22.97438187)	0.825649 (0.836289456)	0 (0.0)	0.032029 (0.02244793)
(5, 40, 40)	5.9 (11.97636561)	0.30698 (0.623155793)	0 (0.0)	0.036408 (0.017846175)
(5, 50, 50)	12.5 (16.05026826)	0.839662 (1.102545699)	0 (0.0)	0.029245 (0.026237826)
(5, 100, 100)	24.9 (21.81971586)	6.330544 (5.638671191)	0 (0.0)	0.09047 (0.068449361)
(5, 100, 200)	30.6 (34.55173255)	13.055885 (14.96527528)	0 (0.0)	0.250534 (0.125183682)
(5, 100, 300)	23.6 (36.23135536)	14.633973 (23.064317)	0 (0.0)	0.308047 (0.209039172)
(5, 100, 400)	57 (36.2981481)	45.280506 (28.92516337)	0 (0.0)	0.373368 (0.226911009)
(5, 100, 500)	47.7 (41.88887415)	48.22372 (42.7063508)	0 (0.0)	0.720032 (0.151168206)
(5, 100, 600)	72.5 (43.47476918)	87.380897 (52.07858271)	0 (0.0)	0.586001 (0.423804503)
(5, 100, 700)	62.6 (45.186527)	89.165117 (65.39505544)	0 (0.0)	0.845871 (0.42601031)
(5, 100, 800)	56.9 (39.09376193)	92.093383 (62.86954342)	0 (0.0)	0.733427 (0.604325317)
(5, 100, 900)	80.7 (40.4833299)	145.666589 (74.46654072)	0 (0.0)	1.391615 (0.73897593)
(5, 100, 1000)	54.525708 (50.4761085)	111.769038 (103.2291643)	0 (0.0)	1.268125 (0.461095115)
(6, 100, 100)	46.3 (29.07098592)	12.446211 (7.413944637)	0 (0.0)	0.141623 (0.099508606)
(7, 100, 100)	48.5 (26.19690397)	13.262261 (7.5017517)	0 (0.0)	0.159061 (0.118739121)
(8, 100, 100)	41.8 (37.66755161)	12.616741 (11.59979295)	0 (0.0)	0.166516 (0.118490103)
(9, 100, 100)	48 (40.57640256)	14.555257 (12.71466426)	0 (0.0)	0.243005 (0.183953506)
(10, 100, 100)	72.1 (35.6353445)	23.198052 (12.03062801)	0 (0.0)	0.200786 (0.110303909)
(10, 100, 200)	162.2 (35.99938271)	93.42215 (20.89542683)	0 (0.0)	0.41298 (0.168034014)
(10, 100, 300)	147 (53.5225809)	126.485167 (47.81166161)	0 (0.0)	0.561374 (0.312797274)
(10, 100, 400)	139.3 (61.18650904)	154.452357 (69.15309498)	0 (0.0)	0.582496 (0.372896875)
(10, 100, 500)	156.7 (78.99936709)	224.351016 (113.7021576)	0 (0.0)	0.865473 (0.159608338)
(10, 100, 1000)	199.6 (56.28933193)	574.328397 (152.4505564)	0 (0.0)	2.246488 (1.335188691)

On the whole, numerical results of Tables 1 and 2 show that the algorithm can globally solve all test problems 1-11 with the robustness and effectiveness.

Explanatory remarks: In Table 2, Avg (Std).NT in the proposed algorithm is 0, this is because, under the same random parameter condition, when using the method proposed in this paper to solve Problem 11, the algorithm meets the termination condition in Step 1, that is to say, when the algorithm solves the relaxation problem in Step 1, we get an approximate global optimal solution of the original problem (MP), and the algorithm is terminated. Since we set the initial iteration number $k = 0$ in Step 1, so that the output iteration number is 0. Obviously, compared with the numerical results of Problem 11 in [18], this shows that the proposed algorithm in this paper has higher computational efficiency than that of [18].

V. CONCLUSION

In this article, based on the branch-and-bound framework, we propose a novel outer space algorithm for globally solving generalized linear multiplicative programming problem (MP) with only assumption that each affine function $\sum_{j=1}^{n} c_{ij}x_j + e_i$ and $\sum_{j=1}^{n} d_{ij}x_j + g_i$ can acquire arbitrary value. By utilizing the equivalent transformation and the direct linear relaxation method, the initial generalized linear multiplicative programming problem (MP) can be converted into a sequence of linear relaxed problems. By subsequently subdividing the initial outer space rectangle of R^p, and by subsequently solving a sequence of linear relaxed problems, the proposed algorithm is convergent to the global optimal solution of the initial problem (MP). During the branch-and-bound searching process, all subproblems that require to be solved in each iteration of the algorithm are all linear programming problems, and the scale of these linear programming problems still remains be unchanged. Finally, comparing with some existent algorithms, numerical results demonstrate that the proposed algorithm has the higher computational efficiency.

In the future, based on the convex relaxation bounding technique and the branch-and-bound framework, the proposed outer space branch-and-bound method can be extended to solve globally minimizing generalized concave multiplicative programming problem.

ACKNOWLEDGMENT

The authors would like thank to editor and the anonymous referees for their valuable comments and suggestions, which have greatly improved the earlier version of this articles.

REFERENCES

[1] H. Konno and T. Kuno, "Generalized linear multiplicative and fractional programming," *Ann. Oper. Res.*, vol. 25, no. 1, pp. 147–161, Dec. 1990.
[2] H.-S. Ryoo and N. V. Sahinidis, "Global optimization of multiplicative programs," *J. Global Optim.*, vol. 26, pp. 387–418, Aug. 2003.
[3] C. Xue and H. Jiao, "An approaximate algorithm for solving generalized linear multiplicative programming," *J. Henan Normal Univ., Natural Sci.*, vol. 36, no. 5, pp. 13–15, 2008.
[4] P. Shen, B. Huang, and L. Wang, "Range division and linearization algorithm for a class of linear ratios optimization problems," *J. Comput. Appl. Math.*, vol. 350, pp. 324–342, Apr. 2019.
[5] H. W. Jiao, S. Y. Liu, and Y. Q. Chen, "Global optimization algorithm for a generalized linear multiplicative programming," *J. Appl. Math. Comput.*, vol. 40, nos. 1–2, pp. 551–568, Oct. 2012.
[6] T. Matsui, "NP-hardness of linear multiplicative programming and related problems," *J. Global Optim.*, vol. 9, no. 2, pp. 113–119, Sep. 1996.

[7] C. H. Scott and T. R. Jefferson, "On duality for a class of quasiconcave multiplicative programs," *J. Optim. Theory Appl.*, vol. 117, no. 3, pp. 575–583, Jun. 2003.

[8] E. A. Youness, "Global optimality of multiplicative programming problems," *Bull. Calcutta Math. Soc.*, vol. 94, no. 2, pp. 129–136, 2002.

[9] H. P. Benson and G. M. Boger, "Multiplicative programming problems: Analysis and efficient point search heuristic," *J. Optim. Theory Appl.*, vol. 94, no. 2, pp. 487–510, Aug. 1997.

[10] S. Liu and Y. Zhao, "An efficient algorithm for globally solving generalized linear multiplicative programming," *J. Comput. Appl. Math.*, vol. 296, pp. 840–847, Apr. 2016.

[11] Z. Wei, D. W. K. Ng, J. Yuan, and H.-M. Wang, "Optimal resource allocation for power-efficient MC-NOMA with imperfect channel state information," *IEEE Trans. Commun.*, vol. 65, no. 9, pp. 3944–3961, Sep. 2017.

[12] Y. Zhao and T. Zhao, "Global optimization for generalized linear multiplicative programming using convex relaxation," *Math. Problems Eng.*, vol. 2018, pp. 9146309.1–9146309.8, May 2018.

[13] Y. Zhao and J. Yang, "Inner approximation algorithm for generalized linear multiplicative programming problems," *J. Inequalities Appl.*, vol. 2018, no. 1, Dec. 2018, Art. no. 354, doi: 10.1186/s13660-018-1947-9.

[14] X. Liu, Y. L. Gao, B. Zhang, and F. P. Tian, "A new global optimization algorithm for a class of linear fractional programming," *Mathematics*, vol. 7, no. 9, p. 867, 2019.

[15] P. Shen and H. Jiao, "Linearization method for a class of multiplicative programming with exponent," *Appl. Math. Comput.*, vol. 183, no. 1, pp. 328–336, Dec. 2006.

[16] H. Jiao, "A branch and bound algorithm for globally solving a class of nonconvex programming problems," *Nonlinear Anal., Theory, Methods Appl.*, vol. 70, no. 2, pp. 1113–1123, Jan. 2009.

[17] Y. Gao, N. Zhang, and X. Ma, "An outcome space branch-and-bound algorithm for a class of linear multiplicative programming problems," in *Advances in Global Optimization*. Springer, 2015.

[18] J. Yin, H. Jiao, and Y. Shang, "Global algorithm for generalized affine multiplicative programming problem," *IEEE Access*, vol. 7, pp. 162245–162253, 2019.

[19] C.-F. Wang, Y.-Q. Bai, and P.-P. Shen, "A practicable branch-and-bound algorithm for globally solving linear multiplicative programming," *Optimization*, vol. 66, no. 3, pp. 397–405, Mar. 2017.

[20] C.-F. Wang, S.-Y. Liu, and P.-P. Shen, "Global minimization of a generalized linear multiplicative programming," *Appl. Math. Model.*, vol. 36, no. 6, pp. 2446–2451, Jun. 2012.

[21] N. T. H. Phuong and H. Tuy, "A unified monotonic approach to generalized linear fractional programming," *J. Global Optim.*, vol. 26, no. 3, pp. 229–259, Jul. 2003.

[22] Y. Chen and H. Jiao, "A nonisolated optimal solution of general linear multiplicative programming problems," *Comput. Oper. Res.*, vol. 36, no. 9, pp. 2573–2579, Sep. 2009.

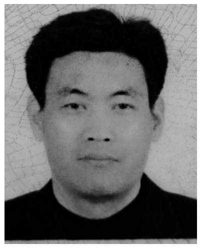
[23] P. P. Shen, T. L. Zhang, and C. F. Wang, "Solving a class of generalized fractional programming problems using the feasibility of linear programs," *J. Inequalities Appl.*, vol. 2017, Jun. 2017, Art. no. 147, doi: 10.1186/s13660-017-1420-1.

[24] P. Shen and C. Wang, "Linear decomposition approach for a class of nonconvex programming problems," *J. Inequalities Appl.*, vol. 2017, no. 1, Dec. 2017, Art. no. 74, doi: 10.1186/s13660-017-1342-y.

[25] P. Shen and B. Huang, "Global algorithm for solving linear multiplicative programming problems," *Optim. Lett.*, vol. 14, no. 3, pp. 693–710, Apr. 2020.

HONGWEI JIAO received the B.S. degree in mathematics and applied mathematics and the M.S. degree in applied mathematics from Henan Normal University, China, in 2003 and 2006, respectively, and the Ph.D. degree in applied mathematics from Xidian University. He is currently an Associate Professor with the School of Mathematical Sciences, Henan Institute of Science and Technology, China. He has published extensively and holds numerous patents in his research areas. His current research interests include optimization algorithms, system engineering, and optimal control.

WENJIE WANG received the B.S. degree in mathematics and applied mathematics from the Henan Institute of Science and Technology, Xinxiang, China, in 2019. She is currently pursuing the master's degree in system science with the School of Mathematics Science, Henan Institute of Science and Technology. She has published extensively and holds numerous patents in her research areas. Her current research interests include optimization algorithms, system science and engineering, and optimal control.



RONGJIANG CHEN received the B.S. degree in mathematics and applied mathematics from Henan University, in 1986. He is currently a Professor with the Department of Mathematics, Henan Institute of Science and Technology, China. He has published over 40 research monographs. His research interests include nonlinear programming, software engineering, computer application, optimization design, product design optimization, manufacturing information systems, global optimization algorithm, nonlinear systems, and optimal control theory.

YOULIN SHANG received the B.S. degree in mathematics and applied mathematics from Henan Normal University, China, in 1984, the M.S. degree in control science and engineering from the Henan University of Science and Technology, China, in 2002, and the Ph.D. degree in operational research and control from Shanghai University. He is currently a Full Professor with the School of Mathematics and Statistics, Henan University of Science and Technology. He has published more than 100 research monographs. He has published extensively and holds numerous patents in his research areas. His current research interests include optimization algorithms, system optimization, system engineering, and optimal control.

JINGBEN YIN received the B.S. degree in mathematics and applied mathematics from Henan University, China, in 1994, and the master's degree from Zhengzhou University, in 2009. He is currently an Associate Professor with the Department of Mathematics, Henan Institute of Science and Technology, China. He has published over 30 research monographs. His research interests include software engineering, computer application, optimization algorithm design, product design, manufacturing information systems, optimization algorithm, nonlinear systems, and optimal control theory.