Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 984168, 10 pages
http://dx.doi.org/10.1155/2013/984168

Research Article

A Simplicial Branch and Bound Duality-Bounds Algorithm to Linear Multiplicative Programming

Xue-Gang Zhou[1] and Bing-Yuan Cao[2]

[1] *Department of Applied Mathematics, Guangdong University of Finance, Guangzhou, Guangdong 510521, China*
[2] *School of Mathematics and Information Science, Key Laboratory of Mathematics and Interdisciplinary Sciences of Guangdong, Higher Education Institutes, Guangzhou University, Guangzhou, Guangdong 510006, China*

Correspondence should be addressed to Bing-Yuan Cao; caobingy@163.com

Received 12 September 2012; Revised 26 January 2013; Accepted 17 February 2013

Academic Editor: Frank Werner

A simplicial branch and bound duality-bounds algorithm is presented to globally solving the linear multiplicative programming (LMP). We firstly convert the problem (LMP) into an equivalent programming one by introducing p auxiliary variables. During the branch and bound search, the required lower bounds are computed by solving ordinary linear programming problems derived by using a Lagrangian duality theory. The proposed algorithm proves that it is convergent to a global minimum through the solutions to a series of linear programming problems. Some examples are given to illustrate the feasibility of the present algorithm.

1. Introduction

1.1. Problem and Applications. In this paper, linear multiplicative programming problems are given by:

$$\begin{aligned} (\text{LMP})\ v = \min \quad & h(x) = \sum_{i=1}^{p} \left(c_i^T x + c_{i0}\right)\left(d_i^T x + d_{i0}\right), \\ \text{s.t.} \quad & x \in X = \left\{x \in R^n \mid Ax \leqslant b,\ x \geqslant 0\right\}, \end{aligned} \tag{1}$$

where $p \geqslant 2$, $c_i^T = (c_{i1}, c_{i2}, \ldots, c_{in})$, $d_i^T = (d_{i1}, d_{i2}, \ldots, d_{in}) \in \mathbb{R}^n$, $i = 1, 2, \ldots, p$, $b^T = (b_1, b_2, \ldots, b_m) \in \mathbb{R}^m$, $c_{i0}, d_{i0} \in \mathbb{R}$, $i = 1, 2, \ldots, p$, A is an $m \times n$ matrix, $X \subseteq R^n$ is a nonempty, compact convex set. We assume also that for each $i = 1, 2, \ldots, p$, $d_i^T x + d_{i0} > 0$. Generally, the problem (LMP) is a special case of non-convex programming problem, known to be an NP-hard even at $p = 1$ [1].

Problem (LMP) has many important applications. Since it subsumes quadratic programming, bilinear programming and linear zero-one programming as special cases, the applications appear quite numerous. Readers may refer to Benson [2] for the following analysis. Now the quadratic programming problem is given as follows:

$$\begin{aligned} \min \quad & \frac{1}{2} y^T Q y + d^T y + c, \\ \text{s.t.} \quad & y \in Y = \left\{y \in R^n \mid Ay \leqslant b,\ y \geqslant 0\right\}, \end{aligned} \tag{2}$$

where Q is an $n \times n$ symmetric matrix of rank $p, d \in R^n$ and $c \in R$. From Tuy [3], there exist linearly independent sets of n-dimensional vectors, $\{v_1, v_2, \ldots, v_p\}$ and $\{w_1, w_2, \ldots, w_p\}$, such that, for all $y \in R^n$,

$$\frac{1}{2} y^T Q y = \frac{1}{2} \sum_{i=1}^{p} \left(v_i^T y\right)\left(w_i^T y\right). \tag{3}$$

Thus, problem (LMP) encompasses the general quadratic programming problem as a special case, and the applications of problem (LMP) include all of the applications of general quadratic programming. Among the latter, for example, quadratic assignment problems [4], problems in economies of scale [5], the constrained linear regression problem [6], VLSI chip design problems [7], the linear complementarity problem [5], and portfolio analysis problems [6].

The bilinear programming problem can be converted into the LMP and it may be written by

$$\begin{aligned} \min \quad & \frac{1}{2} b^T y + d^T z + y^T Q z, \\ \text{s.t.} \quad & y \in Y = \{ y \in R^m \mid C y \leqslant c,\ y \geqslant 0 \}, \\ & z \in Z = \{ z \in R^n \mid E z \leqslant e,\ z \geqslant 0 \}, \end{aligned} \tag{4}$$

where $b \in R^m$, $d \in R^n$, $c \in R^q$, $e \in R^r$, Q is an $m \times n$ matrix of rank p, C and E are $q \times m$ matrix and $r \times n$ matrix, respectively. From Konno and Yajima [8], by using a constructive procedure, it can be written in the form

$$\begin{aligned} \min \quad & \frac{1}{2} b^T y + d^T z + \sum_{i=1}^{p} \left(v_i^T y \right) \left(w_i^T z \right), \\ \text{s.t.} \quad & y \in Y, \quad z \in Z, \end{aligned} \tag{5}$$

where $v_1, v_2, \ldots, v_p \in R^m$ and $w_1, w_2, \ldots, w_p \in R^n$. The latter is a special case of the LMP with $x^T = [y^T, z^T]$, $h(x) = b^T y + d^T z + \sum_{i=1}^{p} (\alpha_i^T x)(\beta_i^T x)$, $\alpha_i^T = [v_i^T, 0^T]$ and $\beta_i^T = [0^T, w_i^T]$, $\alpha_i, \beta_i \in R^{m+n}$, $i = 1, 2, \ldots, p$, and $X = \{(y, z) \in R^{m+n} \mid Cy \leqslant c, Ez \leqslant e, y, z \geqslant 0\}$. Therefore, among the applications problem (P) are all of the applications of bilinear programming, including, for example, location-allocation problems [9], constrained bimatrix games [10], the three-dimensional assignment problem [11], certain linear max-min problems [12], and many problems in engineering design, economic management and operations research.

A linear zero-one programming problem may be written as

$$\begin{aligned} (\text{Q}) \min \quad & \frac{1}{2} b^T y, \\ \text{s.t.} \quad & y \in Y = \{ y \in R^m \mid C y \leqslant c,\ y \geqslant 0 \}, \\ & y_i \in \{0, 1\}, \quad i = 1, 2, \ldots, n, \end{aligned} \tag{6}$$

where $b \in R^m$, $c \in R^p$, C is $p \times m$ matrix. From Raghavachari [13], for $M > 0$ sufficiently large, y^* is an optimal solution to problem (Q) if and only if y^* is an optimal solution to the problem

$$\begin{aligned} (\text{QC}) \min \quad & \frac{1}{2} b^T y + M \sum_{i=1}^{n} y_i \left(1 - y_i \right), \\ \text{s.t.} \quad & y \in Y = \{ y \in R^m \mid C y \leqslant c,\ y \geqslant 0 \}, \\ & 0 \leqslant y_i \leqslant 1, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{7}$$

Since problem (QC) is a special case of the LMP, it follows that all of the numerous applications of linear zero-one programming are embodied among the applications of the LMP. For an overview of some of these applications, see Nemhauser and Wolsey [14].

1.2. Purpose and Content. The LMP is a global optimization problem. In the past 20 years, many algorithms have been proposed globally to a multiplicative programming one. The methods can be classified as parameterization based methods [2, 15–17], outer-approximation and branch-and-bound methods [18–25], vertex enumeration methods [26, 27], a method based on image space analysis [28], a primal and dual simplex method [29], an outcome-space cutting plane method [30], heuristic methods [31, 32], and decomposition method [33].

In this paper, a simplicial branch and bound duality-bounds algorithm is presented to the problem (LMP) by solving a sequence of linear programming one over partitioned subsets. The algorithm implements a simplicial branch and bound search, finding a global optimal solution to the problem, equivalent to the problem (LMP). Branching takes place in a space of only dimension p in the algorithm, where p is the number of terms in the objective function of problem (LMP). During the search, the required lower bounds are computed by solving ordinary linear programming problems. When the algorithm is infinite, any accumulation point of this sequence of feasible solutions is guaranteed to globally solve the problem. The proposed branch and bound algorithm is summarized as follows. Firstly, the branch and bound search takes place in a space of only dimension p, where p is the number of terms in the objective function of problem (LMP), rather than in the decision space $\mathbb{R}^n$. Secondly, the subproblems that must be solved during the search are all linear programming problems that can be solved very efficiently, for example, by a simplex method. The algorithms in this article are motivated by the seminal works of [34], the sum of linear ratios problem, and Horst and Tuy [35] by using branch and bound for global optimization.

The organization and content of this article can be summarized as follows. In Section 2, some preliminary results and operations are presented to implement the simplicial branch and bound duality-bounds algorithm. The simplicial branch and bound duality-bounds algorithm is given in Section 3. In Section 4, the convergence of the algorithm is established. In Section 5 some examples are solved to demonstrate that the proposed algorithm is effective. Some concluding remarks are given in Section 6.

2. Preliminaries

In this section, we firstly show how to convert the LMP into an equivalent nonconvex programming (LMP(S^0)) by introducing a p-dimension vector y for finding a simplex. Then, for each p-dimensional simplex S created by the branching process, the lower bound LB(S) can be found by solving an ordinary linear program by using the Lagrangian weak duality theorem of nonlinear programming.

2.1. Initial Simplex. To globally solve the LMP, the branch and bound algorithm to be developed will be used for searching values of $d_i^T x + d_{i0}$, $i = 1, 2, \ldots, p$, at optimality. For each $i = 1, 2, \ldots, p$, let $l_i = \min_{x \in X} \{d_i^T x + d_{i0}\}$, and $\gamma_i = \max_{x \in X} \{d_i^T x + d_{i0}\}$. By introducing additional variable vector $y = (y_1, y_2, \ldots, y_p) \in \mathbb{R}^p$, we construct a set Y as follows:

$$Y = \{y \in \mathbb{R}^p \mid d_i^T x + d_{i0} - y_i \leqslant 0, \quad i = 1, 2, \ldots, p, y \in D, \text{ for some } x \in X\}, \tag{8}$$

where $D \triangleq \{y \in \mathbb{R}^p \mid l_i \leqslant y_i \leqslant \gamma_i,\ i = 1, 2, \ldots, p\}$.

In order to construct simplex S^0, denoting $\gamma = \min_{x\in X} \sum_{i=1}^{p} (d_i^T x + d_{i0})$. Let $S^0 \in R^p$ be the convex hull of $v^0, v^1, \ldots, v^p$, where $v_j^0 = \gamma_j$, $j = 1, 2, \ldots, n$, and, for each $j = 1, 2, \ldots, n$,

$$v_j^i = \begin{cases} \gamma_j & \text{if } j \neq i, \\ \gamma - \sum_{j\neq i} \gamma_j & \text{if } j = i. \end{cases} \tag{9}$$

Notice that either $\gamma = \sum_{i=1}^{p} \gamma_i$ or $\gamma < \sum_{i=1}^{p} \gamma_i$, and for any $j = 1, \ldots, p$, $(v^j - v^0)^T = (0, 0, \ldots, \gamma - \sum_{i=1}^{p} \gamma_i, \ldots, 0)$, where $\gamma - \sum_{i=1}^{p} \gamma_i$ is the jth component of $v^j - v^0$. We can easily show the following theorem (see Benson [34]).

Theorem 1. *S^0 is either a single point or S^0 is a p-dimensional simplex. In either case, $S^0 \supseteq Y$.*

Remark 2. From Theorem 1, S^0 is either a single point or S^0 is a p-dimensional simplex. Notice that if S^0 is a single one, that is, $S^0 = \{y^*\}$ is a single one, then $X = \{x^*\}$ is a single point set and x^* is a global optimal solution to the LMP, where x^* is any optimal solution to the linear program obtained by setting x equal to x^* in problem (LMP(S^0)). Therefore, we will assume in the remainder of this article that S^0 is a p-dimensional simplex.

2.2. Equivalent Problem. For any simplex $S \subset \mathbb{R}^p$, define the problem

$$\begin{aligned} (\text{LMP}(S)) \quad v(S) = \min \quad & \overline{h}(x, y) = \sum_{i=1}^{p} \left(c_i^T x + c_{i0}\right) y_i, \\ \text{s.t.} \quad & d_i^T x + d_{i0} - y_i \leqslant 0, \\ & i = 1, 2, \ldots, p, \\ & Ax - b \leqslant 0, \\ & x \geqslant 0, \quad y \in S. \end{aligned} \tag{10}$$

In order to solve the LMP, the branch and bound algorithm is used to solves problem (LMP(S^0)) instead. The validity of solving problem (LMP(S^0)), in order to solve the LMP, follows from the next result.

Theorem 3. *If (x^*, y^*) is a global optimal solution for problem (LMP(S^0)), then x^* is a global optimal solution for problem (LMP). If x^* is a global optimal solution for problem (LMP), then (x^*, y^*) is a global optimal solution for problem (LMP(S^0)), where $y_i^* = d_i^T x^* + d_{i0}$, $i = 1, 2, \ldots, p$. The global optimal values v and $v(S^0)$ of problems (LMP) and (LMP(S^0)), respectively, are equal.*

Proof. By using the fact that $S^0 \supseteq Y$, the proof of this theorem follows easily from the defintions of problem (LMP(S^0)). □

2.3. Duality Bound. For each p-dimensional simplex S created by the branching process, the algorithm computes a lower bound LB(S) for the optimal value $v(S)$ of problem (LMP(S)). The next theorem shows that, by using the Lagrangian weak duality theorem of nonlinear programming, the lower bound LB(S) can be found by solving an ordinary linear programming.

Theorem 4. *Let $S \subseteq \mathbb{R}^p$ be a p-dimensional simplex with vertices $\overline{y}^0, \overline{y}^1, \ldots, \overline{y}^p$, and let $J = \{0, 1, 2, \ldots, p\}$. Then $LB(S) \leqslant v(S)$, where $LB(S)$ is the optimal value of the linear programming problem*

$$\begin{aligned} (LP(S))\ LB(S) = \max \quad & \sum_{i=1}^{p} \theta_i d_{i0} - \lambda^T b + t \\ s.t. \quad & -\sum_{i=1}^{p} \theta_i d_i - A^T \lambda \leqslant C, \\ & \sum_{i=1}^{p} \overline{y}_i^j \theta_i + t \leqslant \sum_{i=1}^{p} \overline{y}_i^j c_{i0}, \\ & j = 0, 1, 2, \ldots, p, \\ & \theta \geqslant 0, \quad \lambda \geqslant 0, \ t \ \text{free}, \end{aligned} \tag{11}$$

where $C \in \mathbb{R}^n$ and $C = (\min_{j\in J}(\sum_{i=1}^{p} \overline{y}_i^j c_{i1}), \min_{j\in J}(\sum_{i=1}^{p} \overline{y}_i^j c_{i2}), \ldots, \min_{j\in J}(\sum_{i=1}^{p} \overline{y}_i^j c_{in}))^T$.

Proof. By the definition of $v(S)$ and the weak duality theorem of Lagrangian duality, $v(S) \geqslant \text{LB}(S)$, where

$$\begin{aligned} \text{LB}(S) &= \max_{\substack{\theta\geqslant 0\\ \lambda\geqslant 0}} \left\{ \min_{\substack{y\in S\\ x\geqslant 0}} \left[\sum_{i=1}^{p} \left(c_i^T x + c_{i0}\right) y_i + \sum_{i=1}^{p} \theta_i \left[d_i^T x + d_{i0} - y_i\right] + \lambda^T A x - \lambda^T b \right] \right\} \\ &= \max_{\substack{\theta\geqslant 0\\ \lambda\geqslant 0}} \left\{ \sum_{i=1}^{p} \theta_i d_{i0} - \lambda^T b + \min_{\substack{y\in S\\ x\geqslant 0}} \left[\left\langle \sum_{i=1}^{p} \left(y_i c_i^T + \theta_i d_i^T\right) + \lambda^T A, x \right\rangle + \sum_{i=1}^{p} y_i \left(c_{i0} - \theta_i\right) \right] \right\}. \end{aligned} \tag{12}$$

Since

$$\min_{y\in S}\left\langle \sum_{i=1}^{p}\left(y_i c_i^T+\theta_i d_i^T\right)+\lambda^T A, x\right\rangle = \begin{cases}0, & \sum_{i=1}^{p}\left(y_i c_i^T+\theta_i d_i^T\right)+\lambda^T A \geqslant 0, \quad \forall x\geqslant 0,\\ -\infty, & \text{otherwise},\end{cases} \tag{13}$$

it follows that,

$$\begin{aligned}\mathrm{LB}(S)=\max\quad & \left\{\sum_{i=1}^{p}\theta_i d_{i0}-\lambda^T b+\min_{y\in S}\left[\sum_{i=1}^{p}y_i\left(c_{i0}-\theta_i\right)\right]\right\}\\ \text{s.t.}\quad & \sum_{i=1}^{p}\left(y_i c_i^T+\theta_i d_i^T\right)+\lambda^T A\geqslant 0,\quad \forall y\in S,\\ & \lambda\geqslant 0,\quad \theta\geqslant 0.\end{aligned} \tag{14}$$

Since S is a compact polyhedron with extreme points $\overline{y}^j$, $j=0,1,2,\ldots,p$, for each $\theta\in\mathbb{R}^p$ and $\lambda\geqslant 0$, $\sum_{i=1}^{p}(y_i c_i^T+\theta_i d_i^T)+\lambda^T A\geqslant 0$ holds for all $y\in S$ if and only if it holds for all $y\in\{\overline{y}^0,\overline{y}^1,\ldots,\overline{y}^p\}$. So, for all $j\in J$, we can get $\sum_{i=1}^{p}(\overline{y}_i^j c_i^T+\theta_i d_i^T)+\lambda^T A\geqslant 0$, that is,

$$-\sum_{i=1}^{p}\theta_i d_i^T-\lambda^T A\leqslant \sum_{i=1}^{p}\overline{y}_i^j c_i^T,\quad \forall j\in J. \tag{15}$$

Notice that for all $j\in J$, the left-hand-side of (15) is the same linear function of θ and λ, then (15) is equivalent to $-\sum_{i=1}^{p}\theta_i d_i - A^T\lambda \leqslant C$, where $C\in\mathbb{R}^n$ and $C=(\min_{j\in J}(\sum_{i=1}^{p}\overline{y}_i^j c_{i1}),\ldots,\min_{j\in J}(\sum_{i=1}^{p}\overline{y}_i^j c_{in}))^T$.

Therefore,

$$\begin{aligned}(\mathrm{LB}(S))=\max\quad & \left[\sum_{i=1}^{p}\theta_i d_{i0}-\lambda^T b +\min_{y\in S}\left(\sum_{i=1}^{p}y_i\left(c_{i0}-\theta_i\right)\right)\right]\\ \text{s.t.}\quad & -\sum_{i=1}^{p}\theta_i d_i - A^T\lambda\leqslant C,\\ & \theta\geqslant 0,\quad \lambda\geqslant 0.\end{aligned} \tag{16}$$

That is,

$$\begin{aligned}(\mathrm{LB}(S))=\max\quad & \sum_{i=1}^{p}\theta_i d_{i0}-\lambda^T b+t\\ \text{s.t.}\quad & -\sum_{i=1}^{p}\theta_i d_i - A^T\lambda\leqslant C,\\ & \sum_{i=1}^{p}y_i\left(c_{i0}-\theta_i\right)\geqslant t,\\ & \theta\geqslant 0,\quad \lambda\geqslant 0.\end{aligned} \tag{17}$$

For any $\theta\in\mathbb{R}^p$, $y_i(c_{i0}-\theta_i)$ is a linear function. Because simplex S is a compact polyhedron with extreme points $\overline{y}^0,\overline{y}^1,\ldots,\overline{y}^p$, this implies for any $\theta\in\mathbb{R}^p$, $\sum_{i=1}^{p}y_i(c_{i0}-\theta_i)\geqslant t$ holds if and only if

$$\sum_{i=1}^{p}\overline{y}_i^j\left(c_{i0}-\theta_i\right)-t\geqslant 0,\quad j=0,1,2,\ldots,p. \tag{18}$$

The proof is complete. □

Proposition 5. *Let $S^1,S^2\subseteq\mathbb{R}^p$ be a p-dimensional subsimplices of S formed by the branching process such that $S^1\subseteq S^2\subseteq S^0$. Then*

(i) $LB(S^1)\geqslant LB(S^2)$.

(ii) Let $S\subseteq\mathbb{R}^p$ be a p-dimensional simplex with vertices $\overline{y}^0,\overline{y}^1,\ldots,\overline{y}^p$. Then $LB>-\infty$.

Proof. The proof is similar to [34, Proposition 3], it is omitted here. □

Remark 6. From part (ii) of Proposition 5, for any p-dimensional simplex S created by the algorithm during the branch and bound search, the duality bounds-based lower bound $\mathrm{LB}(S)$ for the optimal value $v(S)$ of problem (LMP(S)) is either finite or equal to $+\infty$. When $\mathrm{LB}(S)=+\infty$, problem (LMP(S)) is infeasible and, as we shall see, S will be eliminated from further consideration by the deletion by bounding process of the algorithm. The monotonicity property in part (i) of Proposition 5 will be used to help to show the convergence of the algorithm.

Now, we show how to determine an upper bound of the global optimal value for (LMP(S)). For each p-dimensional simplex S generated by the algorithm such that $\mathrm{LB}(S)$ is finite, the algorithm generates a feasible solution w to problem (LMP). As the algorithm finds more and more feasible solutions to it an upper bound for the optimal value v of it improves iteratively. These feasible solutions are found from dual optimal ones to the lower bounding problems (LP(S)) that are solved by the algorithm, as given in the following result.

Proposition 7. *Let $S\subseteq\mathbb{R}^p$ be a p-dimensional simplex with vertices $\overline{y}^0,\overline{y}^1,\ldots,\overline{y}^p$, and suppose that $LB(S)\neq+\infty$. Let $w\in\mathbb{R}^n$ be optimal dual variables corresponding to the first n constraints of linear program LP(S). Then w is a feasible solution for problem (LMP).*

Proof. The dual linear program to problem (LP(S)) is

$$\mathrm{DLP}(S)\ \mathrm{LB}(S) = \min\quad C^T w+\sum_{j=0}^{p}q_j\sum_{i=1}^{p}\overline{y}_i^j c_{i0}$$

$$\text{s.t.} \quad -d_i^T w + \sum_{j=0}^{p} \overline{y}_i^j q_j \geqslant d_{i0}, \quad i = 1, 2, \ldots, p,$$
$$-Aw \geqslant -b, \ \sum_{j=0}^{p} q_j = 1, \ w \geqslant 0. \tag{19}$$

The constraints of problem (DLP(S)) imply that $Aw \leqslant b$, $w \geqslant 0$. □

3. Global Optimizing Algorithm

To globally solve problem (LMP(S)), the algorithm to be presented uses a branch and bound approach. There are three fundamental processes in the algorithm, a branching process, a lower bounding one, and an upper bounding one.

3.1. Branching Rule. The branch and bound approach is based on partitioning the p-dimensional simplex S^0 into smaller subsimplices that are also of dimension p, each concerned with a node of the branch and bound tree, and each node is associated with a linear subproblem on each subsimplicie. These subsimplices are obtained by the branching process, which helps the branch and bound procedure identify a location in the feasible region of problem (LMP(S^0)) that contains a global optimal solution to the problem.

During each iteration of the algorithm, the branching process creates a more refined partition of a portion in $S = S^0$ that cannot yet be excluded from consideration in the search for a global optimal solution for problem (LMP(S)). The initial partition Q_1 consists simply of S, since at the beginning of the branch and bound procedure, no portion of S can as yet be excluded from consideration.

During iteration k of the algorithm, $k \geqslant 1$, the branching process is used to help create a new partition Q_{k+1}. First, a screening procedure is used to remove any rectangle from Q_k that can, at this point of the search, be excluded from further consideration, and Q_{k+1} is temporarily set equal to the set of simplices that remain. Later in iteration k, a rectangle S^k in Q_{k+1} is identified for further examination. The branching process is then evoked to subdivide S^k into two subsimplices S_1^k, S_2^k. This subdivision is accomplished by a process called simplicial bisection.

Definition 8 (see [35]). Let S be a p-dimensional simplex with vertex set $\{v^0, v^1, \ldots, v^p\}$. Let w be the midpoint of any of the longest edges $[v^r, v^t]$ of S. Then $\{S^1, S^2\}$ is called a *simplicialbisection* of S, where the vertex set of S^1 is $\{v^0, v^1, \ldots, v^{r-1}, w, v^{r+1}, \ldots, v^p\}$ and the vertex set of S^2 is $\{v^0, v^1, \ldots, v^{t-1}, w, v^{t+1}, \ldots, v^p\}$.

3.2. Lower Bound and Upper Bound. The second fundamental process of the algorithm is the lower bounding one. For each simplex $S \subseteq S^0$ created by the branching process, this process gives a lower bound LB(S) for the optimal value $v(S)$ of the following problem LMP(S),

$$v(S) = \min \quad h(x) = \sum_{i=1}^{p} \left(c_i^T x + c_{i0}\right) y_i,$$
$$\text{s.t.} \quad d_i^T x + d_{i0} - y_i \leqslant 0 \tag{20}$$
$$Ax - b \leqslant 0,$$
$$x \geqslant 0, \quad y \in S.$$

For each simplex S created by the branching process, LB(S) is found by solving a single linear programming LP(S) as follows,

$$(\text{LB}(S)) = \max \quad \sum_{i=1}^{p} \theta_i d_{i0} - \lambda^T b + t^*$$
$$\text{s.t.} \quad -\sum_{i=1}^{p} \theta_i d_i - A^T \lambda \leqslant C,$$
$$\sum_{i=1}^{p} \overline{y}_i^j \theta_i + t \leqslant \sum_{i=1}^{p} \overline{y}_i^j c_{i0}, \tag{21}$$
$$j = 0, 1, 2, \ldots, p,$$
$$\theta \geqslant 0, \quad \lambda \geqslant 0, \ t \text{ free},$$

where $\overline{y}^0, \overline{y}^1, \ldots, \overline{y}^p$ denote the vertices of the p-dimensional simplex S.

During each iteration $k \geqslant 0$, the lower bounding process computes a lower bound LB_k for the optimal value $v(S^0)$ of problem (LMP(S^0)). For each $k \geqslant 0$, this lower process bound LB_k is given by

$$\text{LB}_k = \min \left\{ \text{LB}(S) \mid S \in Q_k \right\}. \tag{22}$$

The upper bounding process is the third fundamental one of the branch and bound algorithm. For each p-dimensional simplex S created by the branching process, this process finds an upper bound for (LMP(S)). Let $w \in \mathbb{R}^n$ be optimal dual variables corresponding to the first n constraints of linear program LP(S), and set $x^* = w^*$. Then, from definition of problem (DLP(S)), we have that $Ax^* \leqslant b$, $x^* \geqslant 0$. This implies that x^* is a feasible solution to (LMP(S)). Therefore, an upper bound UB(S) of (LMP(S)) is $h(x^*)$. In each iteration of the algorithm, this process finds an upper bound for v. For each $k \geqslant 0$, let $w \in \mathbb{R}^n$ be optimal dual variables corresponding to the first n constraints of linear program LP(S), then this upper bound UB_k is given by

$$\text{UB}_k = h(x), \tag{23}$$

where x is the incumbent feasible solution to the problem.

3.3. Deleting Technique. As the branch and bound search proceeds, certain p-dimensional simplices created by the algorithm are eliminated from further consideration. There are two ways occuring, either by deletion by bounding or by deletion by infeasibility.

During any iteration k, $k \geqslant 1$, let UB_k be the smallest objective function value achieved in problem (LMP) by the feasible solutions to problem (LMP(S)) thus far generated by the algorithm. A simplex $S \subseteq S^0$ is deleted by bounding when

$$\mathrm{LB}(S) \geqslant \mathrm{UB}_k \tag{24}$$

holds. When (30) holds, searching simplex S further will not improve upon the best feasible solution found thus far for problem (LMP).

As soon as each p-dimensional simplex S is created by simplicial bisection in the algorithm, it is subjected to the deletion by infeasibility test. Let $\overline{y}^0, \overline{y}^1, \ldots, \overline{y}^p$ denote the vertices of such a simplex S. If for some $i \in \{1, 2, \ldots, p\}$, either

$$\min\left\{\overline{y}_i^0, \overline{y}_i^1, \ldots, \overline{y}_i^p\right\} > L_i, \tag{25}$$

or

$$\max\left\{\overline{y}_i^0, \overline{y}_i^1, \ldots, \overline{y}_i^p\right\} < l_i, \tag{26}$$

then simplex S is said to pass the deletion by infeasibility test and it is eliminated by the algorithm from further consideration. If for each $i \in \{1, 2, \ldots, p\}$, both (25) and (26) fail to hold, then simplex S fails the deletion by infeasibility test and it is retained for further scrutiny by the algorithm. The validity of the deletion by infeasibility test follows from the fact that if (25) or (26) holds for some i, then for each $y \in S$, there is no $x \in X$ such that

$$d_i^T x + d_{i0} - y_i = 0. \tag{27}$$

This implies problem (LMP(S)) infeasible.

3.4. Branch and Bound Algorithm. Based on the results and algorithmic process discussed in this section, the branch and bound algorithm for globally solving the LMP can be stated as follows.

Step 1 (Initialization).

(0.1) Initialize the iteration counter $k := 0$; the set of all active nodes $Q_0 := \{S^0\}$; the upper bound $\mathrm{UB}_0 = +\infty$.

(0.2) Solve linear program (LP(S^0)) for its finite optimal value LB(S^0). Let $w \in \mathbb{R}^n$ be optimal dual variables corresponding to the first n constraints of linear program LP(S^0). Set $x^0 = w$, $\mathrm{UB}_0 = h(x^0)$, $\mathrm{LB}_0 = \mathrm{LB}(S^0)$. Set $k = 1$, and go to iteration k.

Main Step (at iteration k)

Step 2 (Termination). If $\mathrm{UB}_{k-1} - \mathrm{LB}_{k-1} \leqslant \varepsilon$, where $\varepsilon > 0$ is some accuracy tolerance, then stop, and x^{k-1} is a global ε-optimal solution for problem (LMP) and $v = \mathrm{LB}_{k-1}$. Otherwise, set $x^k = x^{k-1}$, $\mathrm{LB}_k = \mathrm{LB}_{k-1}$, $\mathrm{UB}_k = \mathrm{UB}_{k-1}$. Go to Step 3.

Step 3 (Branching). Let $S^k \in Q_{k-1}$ satisfy $S^k \in \arg\min\{\mathrm{LB}(S) \mid S \in Q_{k-1}\}$. Use simplicial bisection to divide S^k into S_1^k and S_2^k. Let $\widehat{R} = \{S_1^k, S_2^k\}$.

Step 4 (Infeasiblity Test). Delete from $\widehat{R}$ each simplex that passes the deletion by infeasiblity test. Let R represent the subset of $\widehat{R}$ thereby obtained.

Step 5 (Fathoming). For each new sub-simplex $S \in R$, compute optimal value LB(S) of linear program (LP(S^0)). If LB(S) is finite, let $w \in \mathbb{R}^n$ be optimal dual variables corresponding to the first $(p+1)n$ constraints of linear program (LP(S)).

Step 6 (Updating upper Bound). If $h(w) < h(x^k)$, set $x^k = w$, $\mathrm{UB}_k = h(x^k)$.

Step 7 (New Partition). Let $Q_k = \{Q_{k-1} \setminus \{S^k\}\} \bigcup R$.

Step 8 (Deletion). $Q_k = Q_k \setminus \{S : \mathrm{LB}(S) - \mathrm{UB}_k \geqslant \varepsilon\}$.

Step 9 (Convergence). If $Q_k = \emptyset$, then stop. UB_k is an optimal value of the LMP, and x^k is a global ε-optimal solution for problem (LMP). Otherwise, set $k+1$ and go to Step 2.

4. Convergence of the Algorithm

In this section we give a global convergence of algorithm above. By the construction of the algorithm, when the algorithm is finite, then $Q_k = \emptyset$, so that $v(S^0) \geqslant \mathrm{UB}_k + \varepsilon = h(x^k) + \varepsilon$. Since, by Proposition 5, $v = v(S^0)$ and since $x^k \in X$, this implies that $v \geqslant h(x^k) + \varepsilon$ and x^k is a global ε-optimal solution to the LMP. Thus, when the algorithm is finite, it globally solves the LMP as desired.

If the algorithm does not terminate after finitely many iterations, then it is easy to show that it generates at least one infinite nested subsequence $\{S^r\}$ of simplices, that is, where $S^{r+1} \subseteq S^r$ for all r. In this case, the following result is a key to convergence in the algorithm.

Theorem 9. *Suppose that the Branch and Bound Algorithm is infinite, and that $\{S^r\}$ is an infinite nested subsequence of simplices generated by the algorithm. Let w^* denote any accumulation point of $\{w^r\}_{r=0}^{\infty}$ where, for each r, $w^r \in \mathbb{R}^n$ denotes any optimal dual variables corresponding to the first n constraints of linear program (LP(S^r)). Then w^* is a global optimal solution for problem (LMP).*

Proof. Suppose that the algorithm is infinite, and let $\{S^r\}$ be chosen as in the theorem. Then, from Horst and Tuy [35], $\bigcap_r S^r = \{y^*\}$ for some point $y^* \in \mathbb{R}^p$.

For each simplex S^r, denote its vertices by $\overline{y}^{r,j}$, $j = 0, 1, \ldots, p$, denote $(w^r, q_0^r, q_1^r, \ldots, q_p^r)$ as an optimal dual solution to linear program (LP(S^r)). Set $U = \{q \in \mathbb{R}^{p+1} \mid \sum_{j=0}^{p} q_j = 1,\ q_j \geqslant 0,\ j = 0, 1, \ldots, p\}$, then U is compact and for each r, $q^r \triangleq (q_0^r, q_1^r, \ldots, q_p^r) \in U$.

By Proposition 7, for each r, $w^r \in X$. Since X is bounded, this implies that $\{w^r\}$ has at least one convergent subsequence.

Let $\{w^r\}_{r\in R}$ denote any such subsequence, and let $w^* = \lim_{r\in R} w^r$. Then, since X is closed, $w^* \in X$. Now, we show that w^* is a global optimal solution for problem (LMP).

Since U is bounded, then there exists an infinite subsequence R' of R such that for each $j = 0,1,2,\ldots,p$, $\lim_{r\in R'} q^r = q^*$. We have $q^* \in U$, since U is close. Notice that since $\lim_{r\in R} w^r = w^*$, $\lim_{r\in R'} w^r = w^*$. Also, since $\bigcap_r S^r = \{y^*\}$, $\lim_{r\in R'} \overline{y}^{r,j} = y^*$ for each $j = 0,1,2,\ldots,p$. Set $(C^r)^T = (\min_{j\in J}(\sum_{i=1}^p \overline{y}_i^{r,j} c_{i1}), \min_{j\in J}(\sum_{i=1}^p \overline{y}_i^{r,j} c_{i2}),\ldots, \min_{j\in J}(\sum_{i=1}^p \overline{y}_i^{r,j} c_{in}))$. Then, we have

$$\begin{aligned}\lim_{r\in R'} (C^r)^T &= C^{*T}\\ &= \left(\min_{j\in J}\left(\sum_{i=1}^p \overline{y}_i^* c_{i1}\right), \min_{j\in J}\left(\sum_{i=1}^p \overline{y}_i^* c_{i2}\right),\right.\\ &\qquad \left.\ldots, \min_{j\in J}\left(\sum_{i=1}^p \overline{y}_i^* c_{in}\right)\right)\\ &= \left(\overline{y}_i^* c_{i1}, \overline{y}_i^* c_{i2}, \ldots, \overline{y}_i^* c_{in}\right).\end{aligned} \tag{28}$$

From Theorems 3–4, Proposition 5, and the algorithm, for each $r, r' \in R'$ such that $r' > r$, $\mathrm{LB}(S^r) \leqslant \mathrm{LB}(S^{r'}) \leqslant v$. Therefore, for some infinite subsequence R'' of R', $\lim_{r\in R''} \mathrm{LB}(S^r)$ exists and satisfies $\lim_{r\in R''} \mathrm{LB}(S^r) \leqslant v$.

Form an objective function of problem (DLP(S^r)), we obtain the equation

$$\mathrm{LB}(S^r) = (C^r)^T w^r + \sum_{j=0}^p q_j^r \sum_{i=1}^p \overline{y}_i^{r,j} c_{i0}, \qquad \forall r \in R''. \tag{29}$$

Taking limits over $r \in R''$ in this equation yields

$$\begin{aligned}\lim_{r\in R''} \mathrm{LB}(S^r) &= (C^*)^T w^* + \sum_{j=0}^p q_j^* \sum_{i=1}^p \overline{y}_i^* c_{i0}\\ &= (C^*)^T w^* + \sum_{i=1}^p \overline{y}_i^* c_{i0}\\ &= \sum_{i=1}^p \left(c_i^T w^* + c_{i0}\right) y_i^* \leqslant v,\end{aligned} \tag{30}$$

where the first equation follows from $\lim_{r\in R''} w^r = w^*$, $\lim_{r\in R'} (C^r)^T = C^{*T}$, $\lim_{r\in R''} \overline{y}^{r,j} = y^*$, $j = 0,1,\ldots,p$, and $\lim_{r\in R''} q^r = q^*$, the second equation holds because $q^* \in U$, the third equation follows form $C^{*T} = (\overline{y}_i^* c_{i1}, \overline{y}_i^* c_{i2}, \ldots, \overline{y}_i^* c_{in})$, and the inequality holds because $\lim_{r\in R''} \mathrm{LB}(S^r) \leqslant v$.

We will now show that $(x, y) = (w^*, y^*)$ is a feasible solution to problem LMP(S^0). First, notice that since S^0 is closed, $y^* \in S^0$, and, since $w^* \in X$, $Aw^* \leqslant b$ and $w^* \geqslant 0$. Let $i \in \{1,2,\ldots,p\}$, For each $r \in R''$, from constraint i of problem (DLP(S^r)), $-d_i^T w^r + \sum_{j=0}^p \overline{y}_1^{r,j} q_j^r = d_{i0}$. Taking limits over $r \in R''$ in the above equation, we obtain

$$\begin{aligned}d_{i0} &= -d_i^T w^* + \sum_{j=0}^p \overline{y}_i^* q_j^*\\ &= -d_i^T w^* + \overline{y}_i^* \sum_{j=0}^p q_j^*\\ &= -d_i^T w^* + \overline{y}_i^*,\end{aligned} \tag{31}$$

where the first equation follows from $\lim_{r\in R''} w^r = w^*$, $\lim_{r\in R''} \overline{y}^{r,j} = y^*$, $j = 0,1,\ldots,p$, and $\lim_{r\in R''} q^r = q^*$, the last equation is due to the fact that $q^* \in U$. By the choice of i, since $y^* \in S^0$, $Aw^* \leqslant b$ and $w^* \geqslant 0$, (31) implies that $(x, y) = (w^*, y^*)$ is a feasible solution for problem LMP(S^0). Therefore, by Theorem 3,

$$\sum_{i=1}^p \left(c_i^T w^* + c_{i0}\right) y_i^* \geqslant v. \tag{32}$$

From (30) and Theorem 3, (32) implies that the feasible solution (w^*, y^*) to problem LMP(S^0) is a global optimal solution to problem LMP(S^0). Therefore, by Theorem 3, w^* is a global optimal solution to the LMP. □

With Theorem 9, we can easily show two fundamental convergence properties of the algorithm as follows.

Corollary 10. *Suppose that the Branch and Bound Algorithm is infinite. Then each accumulation point of* $\{x^k\}_{k=0}^{\infty}$ *is a global optimal solution for problem.*

Proof. The proof is similar to in [34, Corollary 1], it is omitted here. □

Corollary 11. *Suppose that the Branch and Bound Algorithm is infinite. Then* $\lim_{k\to\infty} LB_k = \lim_{k\to\infty} LB_k = v$.

Proof. The proof is similar to [34, Corollary 2], it is omitted here. □

5. Numerical Examples

Now we give numerical experiments for the proposed global optimization algorithm to illustrate its efficiency.

Example 12.

$$\begin{aligned}\min \quad & G(x) = (x_1 + 2x_2 - 2)(-2x_1 - x_2 + 3)\\ & \qquad + (3x_1 - 2x_2 + 3)(x_1 - x2 - 1)\\ \text{s.t.} \quad & -2x_1 + 3x_2 \leqslant 6,\\ & 4x_1 - 5x_2 \leqslant 8,\end{aligned}$$

$$\begin{aligned} &5x_1 + 3x_2 \leqslant 15, \\ &-4x_1 - 3x_2 \leqslant -12, \\ &x_1 \geqslant 0, \quad x_2 \geqslant 0. \end{aligned} \tag{33}$$

Prior to initiating the algorithm, we use Theorem 1 to find a simplex S^0 containing X. By solving three linear programs, we determine that the vertices S^0 are given by $v^0 = \overline{y}^0 = (-1.6667, 1.1351)$, $v^1 = \overline{y}^1 = (3.8649, 1.1351)$, $v^2 = \overline{y}^2 = (-1.6667, 6.6667)$. Also, for each $i = 1, 2$, we compute the minimum l_i and the maximum L_i of $\langle d_i, x\rangle + d_{i0}$ over X for use in the deletion by infeasiblity test (25) and (26). This entails solving four linear programs and get $l_1 = -2.8919$, $l_2 = -2.6667$, $L_1 = -1.6667$, $L_2 = 1.1351$.

Initialization. By solving the following linear programming (LP(S^0)), we get LB(S^0) $= -49.7086$ and the dual variables $w^0 = (1.2857, 2.8571)$,

$$\begin{aligned} \max \quad & 3\theta_1 - \theta_2 - 6\lambda_1 - 8\lambda_2 - 15\lambda_3 + 12\lambda_4 + t \\ \text{s.t.} \quad & -2\theta_1 + \theta_2 - 2\lambda_1 + 4\lambda_2 + 5\lambda_3 - 4\lambda_4 \geqslant -1.7386, \\ & -\theta_1 - \theta_2 + 3\lambda_1 - 5\lambda_2 + 3\lambda_3 - 3\lambda_4 \geqslant 16.6668, \\ & -1.66667\theta_1 + 1.1351\theta_2 + t \leqslant 10.7115 \\ & 3.86649\theta_1 + 1.1351\theta_2 + t \leqslant -0.35165, \\ & -1.66667\theta_1 + 6.66667\theta_2 + t \leqslant 46.6669, \\ & \theta \geqslant 0, \quad \lambda \geqslant 0, \ t \text{ free.} \end{aligned} \tag{34}$$

We set $x^0 = w^*$, $\text{UB}_0 = -16.224$, $\text{LB}_0 = -49.7086$, $G^0 = \{S^0\}$, and $k = 1$. Select the convergence tolerance to be equal to $\varepsilon = 10^{-2}$.

Iteration 1. Since $\text{UB}_0 - \text{LB}_0 < \varepsilon$, S^0 is split by simplicial bisection into S_1^1 and S_2^1 where the vertices of S_1^1 are (−1.6667, 1.1351), (1.0991, 3.9009), (−1.6667, 6.6667) and the vertices of S_2^1 are (−1.6667, 1.1351), (3.8649, 1.1351), (1.0991, 3.9009). Neither S_1^1 nor S_2^1 is deleted by the deletion by infeasibility test. By solving problem (LP(S_1^1)), we obtain the lower bound LB(S_1^1) $= -38.6454$ and the dual variable $w = (1.2857, 2.8571)$. Since $h(w) = -16.224 = h(x^1)$, where $x^1 = x^0$, we do not update y^1 and $\text{UB}_1 = \text{UB}_0$. By solving problem (LP(S_2^1)), we obtain the lower bound LB(S_2^1) $= -18.09944$ and the dual variable $w = (1.2857, 2.8571)$. With problem (LP(S_1^1)), the dual to problem (LP(S_2^1)) does not lead to an update of y^1 and UB_1. We have $Q^1 = \{S_1^1, S_2^1\}$. and neither S_1^1 nor S_2^1 is deleted by Step 8 from Q^1. At the end of Iteration 1, $x^1 = (1.2857, 2.8571)$, $\text{UB}_1 = -16.224$, $\text{LB}_1 = -38.6454$, $G^1 = \{S_1^1, S_2^1\}$.

The algorithm finds a global ε-optimal value −16.2837 after 7 iterations at the global ε-optimal solution $x^* = (1.547, 2.421)$.

Table 1: Computational results of test problems.

Example	p	(m, n)	Iter	Time
1	4	(10, 10)	39.8	28.7
2	4	(10, 20)	44.2	30.3
3	4	(20, 20)	69.1	38.5
4	5	(10, 10)	43.6	29.9
5	5	(10, 20)	50.7	35.0
6	5	(20, 20)	82.8	45.7
7	6	(10, 20)	56.2	30.4
8	7	(10, 20)	67.0	34.5
9	8	(10, 20)	85.6	49.9
10	9	(10, 20)	116.7	78.0

Example 13.

$$\begin{aligned} \min \quad & G(x) = (-x_1 + 2x_2 - 0.5)(-2x_1 + x_2 + 6) \\ & \qquad + (3x_1 - 2x_2 + 6.5)(x_1 + x_2 - 1) \\ \text{s.t.} \quad & -5x_1 + 8x_2 \leqslant 24, \\ & 5x_1 + 8x_2 \leqslant 44, \\ & 6x_1 - 3x_2 \leqslant 15, \\ & -4x_1 - 5x_2 \leqslant -10, \\ & x_1 \geqslant 0, \quad x_2 \geqslant 0. \end{aligned} \tag{35}$$

Prior to initiating the algorithm, we use Theorem 1 to find a simplex S^0 containing X. By solving three linear programs, we determine that the vertices S^0 are given by $v^0 = \overline{y}^0 = (9, 6)$, $v^1 = \overline{y}^1 = (-6.5, 6)$, $v^2 = \overline{y}^2 = (9, 3.5)$. Also, for each $i = 1, 2$, we compute the minimum l_i and the maximum L_i of $\langle d_i, x\rangle + d_{i0}$ over X for use in the deletion by infeasiblity test (25) and (26). This entails solving four linear programs and get $l_1 = 1$, $l_2 = 1$, $L_1 = 9$, $L_2 = 6$.

The algorithm terminates at the beginning of Iteration 29 with the global ε-optimal solution $x^* = (1.5549, 0.7561)$ and ε-optimal value 10.6756 for problem (35).

Example 14. In this example, we solve 10 different random instances for various sizes and objective function structures. These test problems are generated by fixing $\varepsilon = 10^{-5}$. And then all elements of A, c_i, d_i, c_{i0} and d_{i0} are randomly generated, whose ranges are [1, 10]. Since the test problems are coded in C++ and the experiments are conducted on a Pentium IV (3.06 GHZ) microcomputer, the computational results of five problems are summarized in Table 1. The following indices characterize the performance in algorithm: Iter: the average number of iterations; Time: the average execution time in seconds.

6. Conclusion

We have presented and validated a new simplicial branch and bound algorithm globally to the linear multiplicative programming. The algorithm implements a simplicial branch and bound search, finding a global optimal solution to the problem that is equivalent to the LMP. We believe that the new algorithm has advantage in several potentially practical and computational cases. Besides, numerical examples show that the proposed algorithm is feasible.

Acknowledgment

Supported by National Natural Science Foundation of China (no. 70771030, no. 70271047 and no. 11261010) and Project Science Foundation of Guangdong University of Finance (no. 2012RCYJ005).

References

[1] T. Matsui, "NP-hardness of linear multiplicative programming and related problems," *Journal of Global Optimization*, vol. 9, no. 2, pp. 113–119, 1996.

[2] H. P. Benson, "Global maximization of a generalized concave multiplicative function," *Journal of Optimization Theory and Applications*, vol. 137, no. 1, pp. 105–120, 2008.

[3] H. Tuy, *Analysis and Global Optimization*, vol. 22, Kluwer Academic, Dordrecht, The Netherlands, 1998.

[4] P. M. Pardalos and J. B. Rosen, *Constrained Global Optimization: Algorithms and Applications*, vol. 268, Springer, Berlin, Germany, 1987.

[5] C. A. Floudas and V. Visweswaran, "Quadratic optimization," in *Handbook of Global Optimization*, R. Horst and P. M. Pardalos, Eds., pp. 217–269, Kluwer Academic, Dordrecht, The Netherlands, 1995.

[6] R. Horst, P. M. Pardalos, and N. V. Thoai, *Introduction to Global Optimization*, vol. 3, Kluwer Academic, Dordrecht, The Netherlands, 1995.

[7] H. Watanabe, *IC layout generation and compaction using mathematical programming [Ph.D. thesis]*, University of Rochester, Rochester, NY, USA, 1984.

[8] H. Konno and Y. Yajima, "Solving rank two bilinear programs by parametric simplex algorithms," Instituteof Human and Social Sciences Working Paper IHSS 90-17, Tokyo Institute of Technology, Tokyo, Japan, 1990.

[9] H. Konno, P. T. Thach, and H. Tuy, *Optimization on Low Rank Nonconvex Structures*, vol. 15, Kluwer Academic, Dordrecht, The Netherlands, 1997.

[10] O. L. Mangasarian, "Equilibrium points of bimatrix games," *Journal of the Society for Industrial and Applied Mathematics*, vol. 12, pp. 778–780, 1964.

[11] A. M. Frieze, "A bilinear programming formulation of the 3-dimensional assignment problem," *Mathematical Programming*, vol. 7, no. 1, pp. 376–379, 1979.

[12] J. E. Falk, "A linear max-min problem," *Mathematical Programming*, vol. 5, pp. 169–188, 1973.

[13] M. Raghavachari, "On connections between zero-one integer programming and concave programming under linear constraints," *Operations Research*, vol. 17, pp. 680–684, 1969.

[14] G. L. Nemhauser and L. A. Wolsey, *Combinatorial Optimization*, Wiley, New York, NY, USA, 1998.

[15] H. Konno, T. Kuno, and Y. Yajima, "Global minimization of a generalized convex multiplicative function," *Journal of Global Optimization*, vol. 4, no. 1, pp. 47–62, 1994.

[16] H. Konno, Y. Yajima, and T. Matsui, "Parametric simplex algorithms for solving a special class of nonconvex minimization problems," *Journal of Global Optimization*, vol. 1, no. 1, pp. 65–81, 1991.

[17] N. V. Thoai, "A global optimization approach for solving the convex multiplicative programming problem," *Journal of Global Optimization*, vol. 1, no. 4, pp. 341–357, 1991.

[18] H. Konno and T. Kuno, "Generalized linear multiplicative and fractional programming," *Annals of Operations Research*, vol. 25, no. 1–4, pp. 147–161, 1990.

[19] 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*, vol. 3, no. 3, pp. 325–335, 1993.

[20] H.-S. Ryoo and N. V. Sahinidis, "Global optimization of multiplicative programs," *Journal of Global Optimization*, vol. 26, no. 4, pp. 387–418, 2003.

[21] T. Kuno, "Solving a class of multiplicative programs with 0-1 knapsack constraints," *Journal of Optimization Theory and Applications*, vol. 103, no. 1, pp. 121–135, 1999.

[22] H. P. Benson, "An outcome space branch and bound-outer approximation algorithm for convex multiplicative programming," *Journal of Global Optimization*, vol. 15, no. 4, pp. 315–342, 1999.

[23] J. M. Mulvey, R. J. Vanderbei, and S. A. Zenios, "Robust optimization of large-scale systems," *Operations Research*, vol. 43, no. 2, pp. 264–281, 1995.

[24] P. Shen and H. Jiao, "Linearization method for a class of multiplicative programming with exponent," *Applied Mathematics and Computation*, vol. 183, no. 1, pp. 328–336, 2006.

[25] X.-G. Zhou and K. Wu, "A method of acceleration for a class of multiplicative programming problems with exponent," *Journal of Computational and Applied Mathematics*, vol. 223, no. 2, pp. 975–982, 2009.

[26] K. P. Bennett, "Global tree optimization: anon-greedy decision tree algorithm," *Computing Sciences AndStatistics*, vol. 26, pp. 156–160, 1994.

[27] P. M. Pardalos, "Polynomial time algorithms for some classes of constrained nonconvex quadratic problems," *Optimization*, vol. 21, no. 6, pp. 843–853, 1990.

[28] J. E. Falk and S. W. Palocsay, "Image space analysis of generalized fractional programs," *Journal of Global Optimization*, vol. 4, no. 1, pp. 63–88, 1994.

[29] S. Schaible and C. Sodini, "Finite algorithm for generalized linear multiplicative programming," *Journal of Optimization Theory and Applications*, vol. 87, no. 2, pp. 441–455, 1995.

[30] H. P. Benson and G. M. Boger, "Outcome-space cutting-plane algorithm for linear multiplicative programming," *Journal of Optimization Theory and Applications*, vol. 104, no. 2, pp. 301–322, 2000.

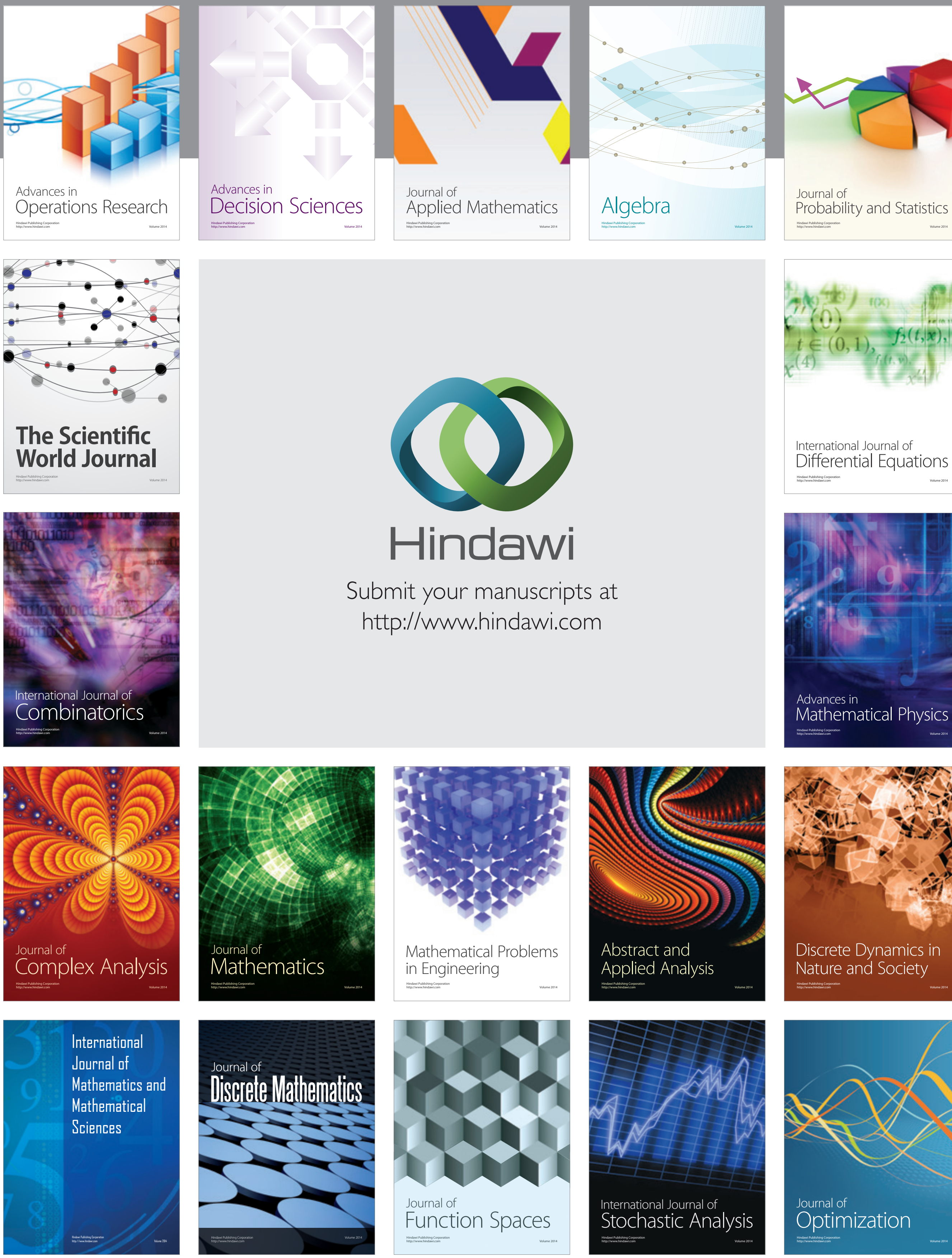
[31] H. P. Benson and G. M. Boger, "Multiplicative programming problems: analysis and efficient point search heuristic," *Journal of Optimization Theory and Applications*, vol. 94, no. 2, pp. 487–510, 1997.

[32] X. J. Liu, T. Umegaki, and Y. Yamamoto, "Heuristic methods for linear multiplicative programming," *Journal of Global Optimization*, vol. 15, no. 4, pp. 433–447, 1999.

[33] H. P. Benson, "Decomposition branch-and-bound based algorithm for linear programs with additional multiplicative constraints," *Journal of Optimization Theory and Applications*, vol. 126, no. 1, pp. 41–61, 2005.

[34] H. P. Benson, “A simplicial branch and bound duality-bounds algorithm for the linear sum-of-ratios problem,” *European Journal of Operational Research*, vol. 182, no. 2, pp. 597–611, 2007.

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



Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
Hindawi
Submit your manuscripts at
http://www.hindawi.com
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization