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

Xue-Gang Zhou and Bing-Yuan Cao

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:

\[
\text{(LMP)} \quad \nu = \min \ h(x) = \sum_{i=1}^{p} (c_i^T x + c_i^0) (d_i^T x + d_i^0),
\]

\[\text{s.t. } \ x \in X = \{x \in \mathbb{R}^n | Ax \leq b, \ x \geq 0\}, \]

where \( p \geq 2 \), \( c_i = (c_{i1}, c_{i2}, \ldots, c_{in}) \), \( d_i = (d_{i1}, d_{i2}, \ldots, d_{in}) \in \mathbb{R}^n \), \( i = 1, 2, \ldots, p \), \( b^T = (b_1, b_2, \ldots, b_m) \in \mathbb{R}^m \), \( c_0, d_0 \in \mathbb{R} \), \( i = 1, 2, \ldots, p \). \( A \) is an \( m \times n \) matrix, \( X \subseteq \mathbb{R}^n \) is a nonempty, compact convex set. We assume also that for each \( i = 1, 2, \ldots, p \), \( d_i^T x + d_i^0 > 0 \). 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:

\[
\min \quad \frac{1}{2} y^T Q y + d^T y + c,
\]

\[\text{s.t. } \ y \in Y = \{y \in \mathbb{R}^n | Ay \leq b, \ y \geq 0\}, \]

where \( Q \) is an \( n \times n \) symmetric matrix of rank \( p \), \( d \in \mathbb{R}^n \) and \( c \in \mathbb{R} \). 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 \( y \in \mathbb{R}^n \),

\[
\frac{1}{2} y^T Q y = \frac{1}{2} \sum_{i=1}^{p} (v_i^T y) (w_i^T y).
\]

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

$$\min \frac{1}{2}b^Ty + d^Tz + y^TQz,$$

s.t. \( y \in Y = \{y \in \mathbb{R}^m \mid Cy \leq c, \ y \geq 0\}, \)

\( z \in Z = \{z \in \mathbb{R}^n \mid Ez \leq e, \ z \geq 0\}, \)

where \( b \in \mathbb{R}^m, d \in \mathbb{R}^n, c \in \mathbb{R}^q, e \in \mathbb{R}^r, Q \) 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

$$\min \frac{1}{2}b^Ty + d^Tz + \sum_{i=1}^{p} (v_i^Ty)(w_i^Tz),$$

s.t. \( y \in Y, \ z \in Z, \)

where \( v_1, v_2, \ldots, v_p \in \mathbb{R}^m \) and \( w_1, w_2, \ldots, w_p \in \mathbb{R}^r \). The latter is a special case of the LMP with \( x^T = [y^T, z^T], h(x) = b^Ty + d^Tz + \sum_{i=1}^{p} (\alpha_i^Tx)(\beta_i^Tz), \alpha_i^T = [v_i^T, 0^T] \) and \( \beta_i^T = [0^T, w_i^T], \alpha_i, \beta_i \in \mathbb{R}^{m+n}, i = 1, 2, \ldots, p, \) and \( X = \{(y, z) \in \mathbb{R}^{m+n} \mid Cy \leq c, Ez \leq e, y, z \geq 0\}. \)

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\((S')\)) 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 \( \ell_i = \min_{x \in X} (d_i^T x + d_{i0}) \), and \( y_i = \max_{x \in X} (d_i^T x + d_{i0}) \). By introducing additional variable...
vector \( y = (y_1, y_2, \ldots, y_p) \in \mathbb{R}^p \), we construct a set \( Y \) as follows:

\[
Y = \{ y \in \mathbb{R}^p \mid d_i^T x + d_0 - y_i \leq 0, \quad i = 1, 2, \ldots, p, y \in D, \text{ for some } x \in X \},
\]

(8)

where \( D \triangleq \{ y \in \mathbb{R}^p \mid l_i \leq y_i \leq u_i, \quad i = 1, 2, \ldots, p \} \).

To construct simplex \( S^0 \), denote \( y = \min_{x \in \mathcal{X}} \sum_{i=1}^{p} (d_i^T x + d_0) \). Let \( S^0 \in \mathbb{R}^p \) be the convex hull of \( v_0^0, v_1^0, \ldots, v_p^0 \), where \( v_j^0 = y_j, \quad j = 1, 2, \ldots, n \), and, for each \( j = 1, 2, \ldots, n \),

\[
v_j^0 = \begin{cases}
y_j & \text{if } j \neq i, \\
y - \sum_{j \neq i} y_j & \text{if } j = i.
\end{cases}
\]

(9)

Notice that either \( y = \sum_{i=1}^{p} y_i \) or \( y < \sum_{i=1}^{p} y_i \), and for any \( j = 1, \ldots, p \), \((v^0 - v_j^0)^T = (0, 0, \ldots, y - \sum_{i=1, i \neq j}^{p} y_i, \ldots, 0)\), where \( y - \sum_{i=1, i \neq j}^{p} y_i \) is the \( j \)th component of \( v^0 - v_j^0 \). 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

\[
\text{(LMP (S))} \quad \nu(S) = \min_{\theta \geq 0, \lambda \geq 0} \{ \sum_{i=1}^{p} (c_i^T x + c_0) \} y_i,
\]

\[
\text{s.t.} \quad d_i^T x + d_0 - y_i \leq 0, \quad i = 1, 2, \ldots, p,
\]

\[
A x - b \leq 0,
\]

\[
x \geq 0, \quad y \in S.
\]

(10)

In order to solve the LMP, the branch and bound algorithm is used to solve 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 \( y_i^* = d_i^T x^* + d_0, \quad i = 1, 2, \ldots, p \). The global optimal values \( \nu \) and \( \nu(S^0) \) 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 definitions 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 \( \text{LB}(S) \) for the optimal value \( \nu(S) \) of problem (LMP(S)). The next theorem shows that, by using the Lagrangian weak duality theorem of nonlinear programming, the lower bound \( \text{LB}(S) \) can be found by solving an ordinary linear programming.

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

\[
\text{(LP (S))} \quad \nu(S) = \max_{\theta \geq 0} \left\{ \sum_{i=1}^{p} (c_i^T x + c_0) \right\} y_i,
\]

\[
s.t. \quad -d_i^T x - d_0 - y_i \leq 0, \quad i = 1, 2, \ldots, p,
\]

\[
A x - b \leq 0,
\]

\[
x \geq 0, \quad y \in S.
\]

(11)

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

\[
\text{LB}(S) = \max_{\theta \geq 0} \left\{ \sum_{i=1}^{p} (c_i^T x + c_0) \right\} y_i
\]

\[
+ \sum_{i=1}^{p} \theta_i [d_i^T x + d_0 - y_i]
\]

\[
+ \lambda^T A x - \lambda^T b
\]

\[
+ \min_{\theta \geq 0} \left\{ \sum_{i=1}^{p} \left( \lambda y_i^* + \theta_i d_i^T x^* \right) + \lambda^T A x \right\}
\]

\[
+ \sum_{i=1}^{p} y_i (c_0 - \theta_i)
\]

(12)
Since
\[
\min_{y \in S} \left\langle \sum_{i=1}^{p} (y_i c_i^T + \theta_i d_i^T) + \lambda^T A, x \right\rangle
\]
\[
= \begin{cases} 
0, & \sum_{i=1}^{p} (y_i c_i^T + \theta_i d_i^T) + \lambda^T A \geq 0, \quad \forall x \geq 0, \\
-\infty, & \text{otherwise,}
\end{cases}
\]

it follows that,
\[
LB(S) = \max \left\{ \sum_{i=1}^{p} \theta_i d_i - \lambda^T b + \min_{y \in S} \left\{ \sum_{i=1}^{p} y_i (c_i - \theta_i) \right\} \right\} \\
s.t. \sum_{i=1}^{p} (y_i c_i^T + \theta_i d_i^T) + \lambda^T A \geq 0, \quad \forall y \in S, \\
\lambda \geq 0, \quad \theta \geq 0.
\]

Since $S$ is a compact polyhedron with extreme points $\bar{y}_j, j = 0, 1, 2, \ldots, p$, for each $\theta \in \mathbb{R}^p$ and $\lambda \geq 0$, \[ \sum_{i=1}^{p} (y_i c_i^T + \theta_i d_i^T) + \lambda^T A \geq 0 \] holds for all $y \in S$ if and only if it holds for all $y \in \{\bar{y}_0, \bar{y}_1, \ldots, \bar{y}_p\}$. So, for all $j \in J$, we can get \[ \sum_{i=1}^{p} (y_i c_i^T + \theta_i d_i^T) + \lambda^T A \geq 0, \quad \forall j \in J. \]

Therefore,
\[
(LB(S)) = \max \left[ \sum_{i=1}^{p} \theta_i d_i - \lambda^T b + \min_{y \in S} \left( \sum_{i=1}^{p} y_i (c_i - \theta_i) \right) \right] \\
s.t. \sum_{i=1}^{p} \theta_i d_i - A^T \lambda \leq C, \\
\theta \geq 0, \quad \lambda \geq 0.
\]

That is,
\[
(LB(S)) = \max \sum_{i=1}^{p} \theta_i d_i - \lambda^T b + t \\
s.t. \sum_{i=1}^{p} \theta_i d_i - A^T \lambda \leq C, \\
\sum_{i=1}^{p} y_i (c_i - \theta_i) \geq t, \\
\theta \geq 0, \quad \lambda \geq 0.
\]

For any $\theta \in \mathbb{R}^p$, $y(c_0 - \theta_i)$ is a linear function. Because simplex $S$ is a compact polyhedron with extreme points $\bar{y}_0, \bar{y}_1, \ldots, \bar{y}_p$, this implies for any $\theta \in \mathbb{R}^p$, \[ \sum_{i=1}^{p} y_i (c_i - \theta_i) \geq t \] holds if and only if
\[
\sum_{i=1}^{p} \bar{y}_i (c_i - \theta_i) - t \geq 0, \quad j = 0, 1, 2, \ldots, p. \quad \text{(18)}
\]

The proof is complete.

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

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

(ii) Let $S \subseteq \mathbb{R}^p$ be a $p$-dimensional simplex with vertices $\bar{y}_0, \bar{y}_1, \ldots, \bar{y}_p$. Then $LB > -\infty$.

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

**Remark.** 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 $LB(S)$ for the optimal value $\nu(S)$ of the problem (LMP(S)) is either finite or equal to $+\infty$. When $LB(S) = +\infty$, the 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 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 $LB(S)$ 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 $\nu$ of it improves iteratively. These feasible solutions are found from dual optimal ones to the lower bounding problems (LMP(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 $\bar{y}_0, \bar{y}_1, \ldots, \bar{y}_p$, and suppose that $LB(S) \neq +\infty$. Let $w \in \mathbb{R}^p$ 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 (LMP(S)) is
\[
DLP(S) \quad LB(S) = \min C^T w + \sum_{j=0}^{p} q_j \sum_{i=1}^{p} \bar{y}_i c_{i0}
\]
The constraints of problem (DLP(S)) imply that \( Aw \leq b, w \geq 0. \)

### 3. Global Optimizing Algorithm

To globally solve problem \((\text{LMP}(S))\), 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 subsimplex. These subsimplices are obtained by the branching process, which helps the branch and bound procedure identify a location in the feasible region of problem \((\text{LMP}(S^0))\) 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 \((\text{LMP}(S))\). The initial partition \( Q_0 \) 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, \( k \geq 1 \), 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 \( S_{k+1}^1, S_{k+1}^2 \). 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 \([v^i, v^j]\) of \( S \). Then \( \{S^1, S^2\} \) is called a simplicial bisection of \( S \), where the vertex set of \( S^1 \) is \( \{v^0, v^1, \ldots, v^{j-1}, w, v^j+1, \ldots, v^p\} \) and the vertex set of \( S^2 \) is \( \{v^0, v^1, \ldots, v^{j-1}, w, v^j+1, \ldots, v^p\} \).

#### 3.2. Lower Bound and Upper Bound

The second fundamental process of the algorithm is the lower bounding process. For each simplex \( S \subseteq S^0 \) created by the branching process, this process gives a lower bound \( \text{LB}(S) \) for the optimal value \( v(S) \) of the following problem \((\text{LMP}(S))\),

\[
\begin{align*}
\text{LB}(S) &= \max \sum_{i=1}^{p} \theta_i d_i - \lambda^T b + t^* \\
\text{s.t.} & \quad -\sum_{i=1}^{p} \theta_i d_i - A^T \lambda \leq C, \\
& \quad \sum_{i=1}^{p} \theta_i = t, \theta \geq 0, \lambda \geq 0, f \text{ free,}
\end{align*}
\]

where \( \theta^1, \theta^2, \ldots, \theta^p \) denote the vertices of the \( p \)-dimensional simplex \( S \).

During each iteration \( k \geq 0 \), the lower bounding process computes a lower bound \( \text{LB}_k \) for the optimal value \( v(S^0) \) of problem \((\text{LMP}(S^0))\). For each \( k \geq 0 \), this lower bound \( \text{LB}_k \) is given by

\[
\text{LB}_k = \min \{ \text{LB}(S) \mid S \in Q_k \}
\]

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 \((\text{LMP}(S))\). Let \( w \in \mathbb{R}^n \) be optimal dual variables corresponding to the first \( n \) constraints of linear program \((\text{LP}(S))\), and set \( x^* = w^* \). Then, from definition of problem \((\text{DLP}(S))\), we have that \( Ax^* \leq b, x^* > 0 \). This implies that \( x^* \) is a feasible solution to \((\text{LMP}(S))\). Therefore, an upper bound \( \text{UB}(S) \) of \((\text{LMP}(S))\) is \( h(x^*) \). In each iteration of the algorithm, this process finds an upper bound \( \text{UB}_k \) for \( v \). For each \( k \geq 0 \), let \( w \in \mathbb{R}^n \) be optimal dual variables corresponding to the first \( n \) constraints of linear program \((\text{LP}(S))\), then this upper bound \( \text{UB}_k \) is given by

\[
\text{UB}_k = h(x),
\]

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 occurring, either by deletion by bounding or by deletion by infeasibility.

During any iteration \( k, k \geq 1 \), let \( \text{UB}_k \) 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

\[
\text{LB}(S) \geq \text{UB}_k \tag{24}
\]

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

\[
\min \{ \overline{y}_i, \overline{y}_1^1, \ldots, \overline{y}_p^1 \} > L_i, \tag{25}
\]
or

\[
\max \{ \overline{y}_0, \overline{y}_1^1, \ldots, \overline{y}_p^p \} < l_i, \tag{26}
\]

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 \( \text{UB}_0 = +\infty \).

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

**Main Step (at iteration \( k \))**

**Step 2 (Termination).** If \( \text{UB}_{k-1} - \text{LB}_{k-1} \leq \varepsilon \), where \( \varepsilon > 0 \) is some accuracy tolerance, then stop, and \( x^{k-1} \) is a global \( \varepsilon \)-optimal solution for problem (LMP) and \( \nu = \text{LB}_{k-1} \).

Otherwise, set \( x^k = x^{k-1} \), \( \text{LB}_k = \text{LB}_{k-1} \), \( \text{UB}_k = \text{UB}_{k-1} \). Go to Step 3.

**Step 3 (Branching).** Let \( S^k \in Q_{k-1} \) satisfy \( S^k \in \arg\min\{\text{LB}(S) \mid S \in Q_{k-1}\} \). Use simplicial bisection to divide \( S^k \) into \( S^k_1 \) and \( S^k_2 \). Let \( \tilde{R} = \{ S^k_1, S^k_2 \} \).

**Step 4 (Infeasibility Test).** Delete from \( \tilde{R} \) each simplex that passes the deletion by infeasibility test. Let \( S \) represent the subset of \( \tilde{R} \) thereby obtained.

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

**Step 6 (Updating upper Bound).** If \( h(\omega) < h(x^k) \), set \( x^k = \omega \), \( \text{UB}_k = h(x^k) \).

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

**Step 8 (Deletion).** \( Q_k = Q_k \setminus \{ S : \text{LB}(S) - \text{UB}_k \geq \varepsilon \} \).

**Step 9 (Convergence).** If \( Q_k = \emptyset \), then stop. \( \text{UB}_k \) is an optimal value of the LMP, and \( x^k \) is a global \( \varepsilon \)-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 \( v(S^k) \geq \text{UB}_k + \varepsilon = h(x^k) + \varepsilon \).

Since, by Proposition 5, \( v = v(S^0) \) and since \( x^k \in X \), this implies that \( v \geq h(x^k) + \varepsilon \) and \( x^k \) is a global \( \varepsilon \)-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 \( \omega^* \) denote any accumulation point of \( \{ \omega^r \}_{r=0}^{\infty} \) where, for each \( r \), \( \omega^r \in \mathbb{R}^n \) denotes any optimal dual variables corresponding to the first \( n \) constraints of linear program (LP(\( S^r \))). Then \( \omega^* \) 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], \( \bigcap_{r=0}^\infty S^r = \{ y^* \} \) for some point \( y^* \in \mathbb{R}^p \).

For each simplex \( S^r \), denote its vertices by \( \overline{y}_j \), \( j = 0, 1, \ldots, p \), denote \( \omega^r, q^r_0, q^r_1, \ldots, q^r_p \) as an optimal dual solution to linear program (LP(\( S^r \))). Set \( U = \{ q \in \mathbb{R}^{p+1} \mid \sum_{j=0}^p q_j = 1, q_j \geq 0, j = 0, 1, \ldots, p \} \), then \( U \) is compact and for each \( r \), \( q^r = (q^r_0, q^r_1, \ldots, q^r_p) \in U \).

By Proposition 7, for each \( r, \omega^r \in X \). Since \( X \) is bounded, this implies that \( \{ \omega^r \} \) has at least one convergent subsequence.
Let \( \{w^r\}_{r \in R} \) denote any such subsequence, and let \( w^* = \lim_{r \to R} w^r \). 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 \( j = 0, 1, 2, \ldots, p \), \( \lim_{r \to R'} q^r_j = q^* \). We have \( q^* \in U \), since \( U \) is close. Notice that since \( \lim_{r \to R} w^r \) = \( w^* \), \( \lim_{r \to R'} w^r \) = \( w^* \). Also, since \( \bigcap_b S^r = \{y^*\} \), \( \lim_{r \to R'} y^r_j = y^* \) for each \( j = 0, 1, 2, \ldots, p \).

Set \( (C^r)^T = (\min_{j \in J}(\sum_{i=1}^{p} y^r_{ij} c_1), \min_{j \in J}(\sum_{i=1}^{p} y^r_{ij} c_2), \ldots, \min_{j \in J}(\sum_{i=1}^{p} y^r_{ij} c_n)) \). Then, we have

\[
\lim_{r \to R'} (C^r)^T = C^*^T = \left( \min_{j \in J}(\sum_{i=1}^{p} y^r_{ij} c_1), \min_{j \in J}(\sum_{i=1}^{p} y^r_{ij} c_2), \ldots, \min_{j \in J}(\sum_{i=1}^{p} y^r_{ij} c_n) \right).
\]

From Theorems 3–4, Proposition 5, and the algorithm, for each \( r, r' \in R' \) such that \( r' \geq r \), \( LB(S^{r'}) \leq LB(S^r) \leq v \). Therefore, for some infinite subsequence \( R'' \) of \( R' \), \( \lim_{r \to R''} LB(S^r) \leq v \).

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

\[
LB(S^r) = (C^r)^T w^r + \sum_{j=0}^{p} q^r_j \sum_{i=1}^{p} y^r_{ij} c_0,
\]

\[
\forall r \in R''.
\]

Taking limits over \( r \in R'' \) in this equation yields

\[
\lim_{r \to R''} LB(S^r) = (C^*)^T w^* + \sum_{j=0}^{p} q^*_j \sum_{i=1}^{p} y^*_r c_0,
\]

\[
= (C^*)^T w^* + \sum_{i=1}^{p} (c_0^T w^* + c_0) y^*_r \leq v,
\]

where the first equation follows from \( \lim_{r \to R''} w^r = w^* \), \( \lim_{r \to R''} (C^r)^T = C^*^T \), \( \lim_{r \to R''} y^r_{ij} = y^*_r \), \( j = 0, 1, \ldots, p \), and \( \lim_{r \to R''} q^r_j = q^*_j \), the second equation holds because \( q^*_j \in U \), the third equation follows from \( (C^*)^T = (y^*_r c_1, y^*_r c_2, \ldots, y^*_r c_n) \), and the inequality holds because \( \lim_{r \to R''} LB(S^r) \leq v \).

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 \( w^* \in X \), \( Aw^* \leq b \) and \( w^* > 0 \). Let \( i \in \{1, 2, \ldots, p\} \). For each \( r \in R'' \), from constraint \( i \) of problem (DLP(S^r)), \( -d_i^r w^r + \sum_{j=0}^{p} y_{ij}^r q_j^r = d_0^r \). Taking limits over \( r \in R'' \) in the above equation, we obtain

\[
d_0 = -d_i^* w^* + \sum_{j=0}^{p} y_{ij}^* q_j^* \leq d_0.
\]

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 \( \{x^k\}_{k=1}^{\infty} \) 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.**

\[
\min \ G(x) = (x_1 + 2x_2 - 2) (-2x_1 - x_2 + 3) + (3x_1 - 2x_2 + 3)(x_1 - x_2 - 1)
\]

s.t. \( -2x_1 + 3x_2 \leq 6 \),

\( 4x_1 - 5x_2 \leq 8 \),
5x_1 + 3x_2 \leq 15, \\
-4x_1 - 3x_2 \leq -12, \\
x_1 \geq 0, \quad x_2 \geq 0.

(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 $v^0 = \bar{y}^0 = (-1.6667, 1.1351)$, $v^1 = \bar{y}^1 = (3.8649, 1.1351)$, $\bar{y}^2 = \bar{y}^2 = (-1.6667, 6.6667)$. Also, for each $i = 1, 2$, we compute the minimum $l_i$ and the maximum $L_i$ of $\langle d, x \rangle + d_i$ over $X$ for use in the deletion by infeasibility 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 $w^0 = (1.2857, 2.8571)$,

\[
\begin{align*}
\max & \quad 3\theta_1 - \theta_2 - 6\lambda_1 - 8\lambda_2 - 15\lambda_3 + 12\lambda_4 + t \\
\text{s.t.} & \quad -2\theta_1 + \theta_2 - 2\lambda_1 + 4\lambda_2 + 5\lambda_3 - 4\lambda_4 \geq -1.7386, \\
& \quad -\theta_1 - \theta_2 + 3\lambda_1 - 5\lambda_2 + 3\lambda_3 - 3\lambda_4 \geq 16.6668, \\
& \quad -1.6667\theta_1 + 1.1351\theta_2 + t \leq 0.35165, \\
& \quad -1.6667\theta_1 + 6.6667\theta_2 + t \leq 46.6669, \\
\theta & \geq 0, \quad \lambda \geq 0, \quad t \text{ free.}
\end{align*}
\]

We set $x^0 = w^*$, $UB_0 = -16.224$, $LB_0 = -49.7086$, $G_0 = \{ S^0 \}$, and $k = 1$. Select the convergence tolerance to be equal to $\epsilon = 10^{-2}$.

Iteration 1. Since $UB_0 - LB_0 < \epsilon$, $S^0$ is split by simplicial bisection into $S^1_1$ and $S^1_2$ where the vertices of $S^1_1$ are $(-1.6667, 1.1351), (1.0991, 3.9009), (-1.6667, 6.6667)$ and the vertices of $S^1_2$ are $(-1.6667, 1.1351), (3.8649, 1.1351), (1.0991, 3.9009)$. Neither $S^1_1$ nor $S^1_2$ 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 $UB_1 = UB_0$. By solving problem ($LP(S^1_2)$), we obtain the lower bound $LB(S^1_2) = -18.09944$ and the dual variable $w = (1.2857, 2.8571)$. With problem ($LP(S^1_2)$), the dual to problem ($LP(S^1_1)$) does not lead to an update of $y^1$ and $UB_1$. We have $Q^1 = \{ S^1_1, S^1_2 \}$. neither $S^1_1$ nor $S^1_2$ is deleted by Step 8 from $Q^1$. At the end of Iteration 1, $x^1 = (1.2857, 2.8571)$, $UB_1 = -16.224$, $LB_1 = -38.6454$, $G^1 = \{ S^1_1, S^1_2 \}$.

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

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

Example 13.

\[
\begin{align*}
\min & \quad G(x) = (-x_1 + 2x_2 - 0.5)(-2x_1 + x_2 + 6) \\
& \quad + (3x_1 - 2x_2 + 6.5)(x_1 + x_2 - 1)
\end{align*}
\]

s.t. $-5x_1 + 8x_2 \leq 24,$

$5x_1 + 8x_2 \leq 44,$

$6x_1 - 3x_2 \leq 15,$

$-4x_1 - 5x_2 \leq -10,$

$x_1 \geq 0, \quad x_2 \geq 0.$

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 $v^0 = \bar{y}^0 = (9, 6)$, $v^1 = \bar{y}^1 = (-6.5, 6)$, $v^2 = \bar{y}^2 = (9, 3.5)$. Also, for each $i = 1, 2$, we compute the minimum $l_i$ and the maximum $L_i$ of $\langle d, x \rangle + d_i$ over $X$ for use in the deletion by infeasibility 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 $\epsilon$-optimal solution $x^* = (1.5549, 0.7561)$ and $\epsilon$-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 $\epsilon = 10^{-5}$. And then all elements of $A, c, d, c_0$ and $d_0$ 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


Submit your manuscripts at http://www.hindawi.com