Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 160262, 8 pages
http://dx.doi.org/10.1155/2014/160262

Research Article

An Effective Branch and Bound Algorithm for Minimax Linear Fractional Programming

Hong-Wei Jiao,[1] Feng-Hui Wang,[2] and Yong-Qiang Chen[3]

[1] *Department of Mathematics, Henan Institute of Science and Technology, Xinxiang 453003, China*
[2] *Department of Mathematics, Luoyang Normal University, Luoyang 471022, China*
[3] *School of Mathematics, Henan Normal University, Xinxiang 453007, China*

Correspondence should be addressed to Hong-Wei Jiao; jhwxd2013@126.com

Received 15 December 2013; Accepted 30 May 2014; Published 15 June 2014

Academic Editor: Farhad Hosseinzadeh Lotfi

An effective branch and bound algorithm is proposed for globally solving minimax linear fractional programming problem (MLFP). In this algorithm, the lower bounds are computed during the branch and bound search by solving a sequence of linear relaxation programming problems (LRP) of the problem (MLFP), which can be derived by using a new linear relaxation bounding technique, and which can be effectively solved by the simplex method. The proposed branch and bound algorithm is convergent to the global optimal solution of the problem (MLFP) through the successive refinement of the feasible region and solutions of a series of the LRP. Numerical results for several test problems are reported to show the feasibility and effectiveness of the proposed algorithm.

1. Introduction

Minimax linear fractional programming problem (MLFP) has become a subject of wide interest for practitioners and scientists [1–3], which has broad applications in various disciplines, for examples, system engineering [3], electronic science [4], and management science [5], and whose mathematical modeling can be formulated as follows:

$$\begin{aligned}(\text{MLFP}):\ \min_{x}\ \max\ & \left\{\Psi_1(x), \Psi_2(x), \ldots, \Psi_p(x)\right\} \\ \text{s.t.}\quad & x \in D = \left\{x \in R^n \mid Ax \le b, x \ge 0\right\},\end{aligned} \tag{1}$$

where

$$\Psi_j(x) = \frac{\sum_{i=1}^{n} c_{ji}x_i + d_j}{\sum_{i=1}^{n} e_{ji}x_i + f_j}, \quad j = 1, 2, \ldots, p, \tag{2}$$

$\sum_{i=1}^{n} c_{ji}x_i + d_j$ and $\sum_{i=1}^{n} e_{ji}x_i + f_j$ are all affine functions such that $\sum_{i=1}^{n} c_{ji}x_i + d_j > 0$ and $\sum_{i=1}^{n} e_{ji}x_i + f_j > 0$, $A \in R^{m\times n}$, $b \in R^m$, and D is a nonempty compact set.

The minimax linear fractional programming problems (MLFP) pose significant theoretical and computational challenges. This is mainly because the problems (MLFP) possess multiple local optima that are not globally optimal. Therefore, it is necessary to put forward an effective global optimization algorithm for solving the minimax linear fractional programming problem (MLFP). During the past years, some algorithms have been proposed for solving the minimax linear fractional programming problem (MLFP), for instance, dual methods [2, 6, 7], parametric programming methods [8, 9], interior-point algorithms [10, 11], monotonic optimization approach [12], exact method [13], approximation algorithm [14], cutting plane algorithm [15], method of centers [16], inexact proximal point method [17], interval-type algorithm [18], and so on. Recently, based on the linear relaxation and branch and bound scheme, a solution algorithm has been developed for solving globally the minimax linear fractional programming problem (MLFP) [19]. Up to now, although there has been significant progress in the development of algorithms for solving the minimax linear fractional programming, to our knowledge, less work has been still done for globally solving the minimax linear fractional programming problem (MLFP).

The purpose of this paper is to present a new global optimization algorithm for solving the minimax linear fractional programming problem (MLFP), and the goals of this research are threefold. First, we present a transformation of the problem; thus, the original problem (MLFP) is reformulated as an equivalent problem (EP). Second, in order to design an effective branch and bound algorithm for the equivalent problem (EP), a new linear relaxation bounding technique is presented, and, by utilizing this technique, the nonconvex programming problem (EP) is reduced to a sequence of linear relaxation programming problems, which can provide reliable lower bounds for the optimal value of the problem (EP) and are embedded into the branch and bound framework. The main computational operation in the algorithm only involves solving a sequence of linear relaxation programming subproblems that do not grow in size from iteration to iteration. Third, compare our algorithm (using new linear relaxation bounding technique) with the known algorithms (recent literatures) with respect to robustness (finding the optimum) and efficiencies (number of function evaluations), and the numerical experimental results show that the proposed algorithm is robust and effective.

This paper is organized as follows. The next section firstly converts the problem (MLFP) into an equivalent problem (EP); a new linear relaxation bound method is presented, and then the linear relaxation programming of the problem (EP) is established. Section 3 a branch and bound algorithm is proposed for globally solving the problem (MLFP), and the convergence property of the algorithm is given. In Section 4, we report the numerical results for solving some examples with the proposed algorithm. Finally, a few concluding remarks are given in Section 5.

2. Linear Relaxation Programming

In order to globally solve the problem (MLFP), we first compute the initial lower bound $\underline{x}_i = \min_{x\in D} x_i$ and upper bound $\overline{x}_i = \max_{x\in D} x_i$ of each variable x_i and denote the initial rectangle by

$$X^0 = \{x \mid \underline{x}_i \le x_i \le \overline{x}_i, i = 1, \dots, n\}. \quad (3)$$

Next, we can convert the problem (MLFP) into the following equivalent problem (EP), which has the same global optimal solution and optimal value as the problem (MLFP):

$$\begin{aligned} \text{(EP)}: \quad & \min_{x,t} \quad t \\ & \text{s.t.} \quad \Psi_j(x) - t \le 0, \quad j = 1, \dots, p, \\ & \qquad Ax \le b, \quad x \ge 0, \\ & \qquad x \in X^0. \end{aligned} \quad (4)$$

In the following, we only consider solving the equivalent problem (EP); the important step in the construction of a solution procedure for globally solving the problem (EP) is the establishment of a linear relaxation programming for computing the lower bounds of the optimal value for this problem. Here, we only need to construct linear lower bounding function of $\Psi_j(x)$ in constraint function. For each $j \in \{1, \dots, p\}$, we can let

$$\begin{aligned} h_j(x) &= \ln\left(\sum_{i=1}^n c_{ji}x_i + d_j\right), \\ g_j(x) &= \ln\left(\sum_{i=1}^n e_{ji}x_i + f_j\right), \\ \Psi_j(x) &= \exp\left[h_j(x) - g_j(x)\right]. \end{aligned} \quad (5)$$

The detailed new linear relaxation bounding technique can be given as follows.

First, we consider the function $h_j(x)$ $(j = 1, \dots, p)$. For convenience in expression, for $\forall x \in X^k = [\underline{x}^k, \overline{x}^k] \subseteq X^0$, some notations are introduced as follows:

$$\begin{aligned} X_j &= \sum_{i=1}^n c_{ji}x_i + d_j, \\ \underline{X}_j &= \sum_{i=1}^n \min\left\{c_{ji}\underline{x}_i^k, c_{ji}\overline{x}_i^k\right\} + d_j, \\ \overline{X}_j &= \sum_{i=1}^n \max\left\{c_{ji}\underline{x}_i^k, c_{ji}\overline{x}_i^k\right\} + d_j, \\ K_j &= \frac{\ln \overline{X}_j - \ln \underline{X}_j}{\overline{X}_j - \underline{X}_j}, \\ h_j^l(x) &= \ln\left(\underline{X}_j\right) + K_j\left(\sum_{i=1}^n c_{ji}x_i + d_j - \underline{X}_j\right). \end{aligned} \quad (6)$$

Obviously, by the characteristic of the concave function, we have

$$h_j^l(x) \le h_j(x). \quad (7)$$

Second, we consider the concave function $g_j(x) = \ln(\sum_{i=1}^n e_{ji}x_i + f_j)$ $(j = 1, \dots, p)$ about the whole variable $(\sum_{i=1}^n e_{ji}x_i + f_j)$; by the property of the concave function, we have

$$g_j(x) \le g_j(x_{\text{mid}}) + \nabla g_j(x_{\text{mid}})^T (x - x_{\text{mid}}) = g_j^u(x), \quad (8)$$

where

$$\begin{aligned} x_{\text{mid}} &= \frac{1}{2}\left(\underline{x}^k + \overline{x}^k\right), \\ \nabla g_j(x) &= \begin{bmatrix} \dfrac{e_{j1}}{\sum_{i=1}^n e_{ji}x_i + f_j} \\ \vdots \\ \dfrac{e_{jp}}{\sum_{i=1}^n e_{ji}x_i + f_j} \end{bmatrix}. \end{aligned} \quad (9)$$

From (7) and (8), for all $x \in X^k$, we have

$$h_j(x) - g_j(x) \ge h_j^l(x) - g_j^u(x). \quad (10)$$

Let

$$H_j(x) = h_j^l(x) - g_j^u(x), \qquad G_j(x) = \exp\left(H_j(x)\right). \quad (11)$$

By (5), (10), and (11), we have

$$\begin{aligned} G_j(x) = & \exp\left(H_j(x)\right) = \exp\left(h_j^l(x) - g_j^u(x)\right) \\ \leq & \exp\left(h_j(x) - g_j(x)\right) = \Psi_j(x). \end{aligned} \quad (12)$$

Third, we consider the function $G_j(x)$ $(j = 1, \ldots, p)$. By the property of the convex function, we have

$$G_j(x) \geq G_j(x_{\text{mid}}) + \nabla G_j(x_{\text{mid}})^T (x - x_{\text{mid}}), \quad (13)$$

where

$$\nabla G_j(x_{\text{mid}}) = \left(K_j \begin{bmatrix} c_{j1} \\ \vdots \\ c_{jn} \end{bmatrix} - \nabla g_j(x_{\text{mid}}) \right) \exp\left(H_j(x_{\text{mid}})\right). \quad (14)$$

Let

$$G_j^l(x) = G_j(x_{\text{mid}}) + \nabla G_j(x_{\text{mid}})^T (x - x_{\text{mid}}), \quad (15)$$

and, by (12) and (13), we have

$$G_j^l(x) \leq G_j(x) \leq \Psi_j(x). \quad (16)$$

Therefore, we can follow the linear lower bounding function $G_j^l(x)$ of $\Psi_j(x)$ for each $j \in \{1, \ldots, p\}$, which underestimates the function $\Psi_j(x)$ as follows:

$$G_j^l(x) = G_j(x_{\text{mid}}) + \nabla G_j(x_{\text{mid}})^T (x - x_{\text{mid}}). \quad (17)$$

According to the above discussion, for $\forall X^k \subseteq X^0$, we can construct the linear relaxation programming (LRP) of the problem (EP) in X^k as follows:

$$\begin{aligned} \text{(LRP)}: \quad \min \quad & t \\ \text{s.t.} \quad & G_j^l(x) - t \leq 0, \quad j = 1, \ldots, p, \\ & Ax \leq b, \\ & x \in X^k. \end{aligned} \quad (18)$$

Based on the above construction method of the linear relaxation programming problem, for $\forall X^k \subseteq X^0$, LRP($X^k$) provides a valid lower bound for the optimal value of EP(X^k).

The following theorem will ensure that each $G_j^l(x)$ will approximate the corresponding function $\Psi_j(x)$ as $\|\overline{x}^k - \underline{x}^k\| \to 0$; that is, the optimal solution of the LRP(X^k) will approximate the optimal solution of EP(X^k).

Theorem 1. *For $\forall x \in X^k = [\underline{x}^k, \overline{x}^k] \subseteq X^0$, for each $j = 1, \ldots, p$, and then the error $\Theta_j = \Psi_j(x) - G_j^l(x) \to 0$ as $\|\overline{x}^k - \underline{x}^k\| \to 0$.*

Proof. Let

$$\Theta_j = \left[\Psi_j(x) - G_j(x)\right] + \left[G_j(x) - G_j^l(x)\right] = \Theta_{j1} + \Theta_{j2}, \quad (19)$$

and then it is obvious that $\Theta_{j1} \geq 0$ and $\Theta_{j2} \geq 0$.

First, we consider the difference Θ_{j1}. Let

$$\begin{aligned} \widehat{\Psi}_j(x) = & \ln\left(h_j(x)\right) - \ln\left(g_j(x)\right) \\ = & \ln\left(\sum_{i=1}^n c_{ji} x_i + d_j\right) - \ln\left(\sum_{i=1}^n e_{ji} x_i + f_j\right), \end{aligned} \quad (20)$$

and it follows that

$$\begin{aligned} \left|\Theta_{j1}\right| = & \left| \exp\left(\ln\left(\sum_{i=1}^n c_{ji} x_i + d_j\right) - \ln\left(\sum_{i=1}^n e_{ji} x_i + f_j\right) \right) \right. \\ & \left. - \exp\left(H_j(x)\right) \right| \\ = & \left| \exp\left(\widehat{\Psi}_j(x)\right) - \exp\left(H_j(x)\right) \right| \\ \leq & \left\| \widehat{\Psi}_j(x) - H_j(x) \right\| \sup_{\sigma_j \in L(\widehat{\Psi}_j(x), H_j(x))} \exp\left(\sigma_j\right), \end{aligned} \quad (21)$$

where

$$L\left(\widehat{\Psi}_j(x), H_j(x)\right) = \alpha \widehat{\Psi}_j(x) + (1 - \alpha) H_j(x), \quad \text{for } \forall \alpha \in [0, 1]. \quad (22)$$

Let

$$\begin{aligned} \Theta_{j1.1} = & \ln\left(\sum_{i=1}^n c_{ji} x_i + d_j\right) \\ & - \left[\ln\left(\underline{X}_j\right) + K_j \left(\sum_{i=1}^n c_{ji} x_i + d_j - \underline{X}_j \right) \right], \\ \Theta_{j1.2} = & g_j(x_{\text{mid}}) + \nabla g_j(x_{\text{mid}})^T (x - x_{\text{mid}}) - g_j(x), \end{aligned} \quad (23)$$

We have

$$\begin{aligned} & \widehat{\Psi}_j(x) - H_j(x) \\ & = \left\{ \ln\left(\sum_{i=1}^n c_{ji} x_i + d_j\right) - \ln\left(\sum_{i=1}^n e_{ji} x_i + f_j\right) \right\} \\ & \quad - \left\{ \ln\left(\underline{X}_j\right) + K_j \left(\sum_{i=1}^n c_{ji} x_i + d_j - \underline{X}_j \right) \right. \\ & \quad \left. - g_j(x_{\text{mid}}) - \nabla g_j(x_{\text{mid}})^T (x - x_{\text{mid}}) \right\} \\ & = \Theta_{j1.1} + \Theta_{j1.2}. \end{aligned} \quad (24)$$

Since $\Theta_{j1.1}$ is a concave function about $(\sum_{i=1}^{n} c_{ji}x_i + d_j)$, we can know that $\Theta_{j1.1}$ can attain the maximum $\Theta_{j1.1}^{\max}$ at the point $\sum_{i=1}^{n} c_{ji}x_i + d_j = 1/K_j$. Let $u_j = \overline{X}_j/\underline{X}_j$. Then, through computing, we derive

$$\Theta_{j1.1}^{\max} = \frac{\ln u_j}{u_j - 1} - 1 - \ln \frac{\ln u_j}{u_j - 1}. \tag{25}$$

Since $u_j \to 1$ as $\|\overline{x}^k - \underline{x}^k\| \to 0$, we have $\Theta_{j1.1}^{\max} \to 0$ as $\|\overline{x}^k - \underline{x}^k\| \to 0$.

One has

$$\begin{aligned} \Theta_{j1.2} &= g_j(x_{\text{mid}}) + \nabla g_j(x_{\text{mid}})^T (x - x_{\text{mid}}) - g_j(x) \\ &= \nabla g_j(x_{\text{mid}})^T (x - x_{\text{mid}}) - \nabla g_j(\eta)^T (x - x_{\text{mid}}) \\ &\le \left\|\nabla^2 g_j(\rho)\right\| \|\eta - x_{\text{mid}}\| \|x - x_{\text{mid}}\|, \end{aligned} \tag{26}$$

where η and ρ are constant vectors, which satisfy

$$\begin{aligned} g_j(x) - g_j(x_{\text{mid}}) &= \nabla g_j(\eta)^T (x - x_{\text{mid}}), \\ \nabla g_j(x_{\text{mid}}) - \nabla g_j(\eta) &= \nabla^2 g_j(\rho)(\eta - x_{\text{mid}}). \end{aligned} \tag{27}$$

Since $\nabla^2 g_j(x)$ is continuous, and X is a compact set, there exists some $M > 0$ such that

$$\left\|\nabla^2 g_j(x)\right\| \le M. \tag{28}$$

By (26), we have

$$\Theta_{j1.2} \le M\left\|\overline{x}^k - \underline{x}^k\right\|^2, \tag{29}$$

Furthermore, we have $\Theta_{j1.2} \to 0$ as $\| \overline{x}^k - \underline{x}^k \| \to 0$.

Therefore, we have

$$\widehat{\Psi}_j(x) - H_j(x) = \Theta_{j1.1} + \Theta_{j1.2} \longrightarrow 0 \quad \text{as} \quad \left\|\overline{x}^k - \underline{x}^k\right\| \longrightarrow 0. \tag{30}$$

Since $\exp(\sigma_j)$ is a continuous and bounded function about x, there exists some $\overline{M} > 0$ such that $|\exp(\sigma_j)| \le \overline{M}$. Therefore, by (30), we have

$$\Theta_{j1} \le \overline{M}\left|\widehat{\Psi}_j(x) - H_j(x)\right| \longrightarrow 0 \quad \text{as} \quad \left\|\overline{x}^k - \underline{x}^k\right\| \longrightarrow 0. \tag{31}$$

Second, we consider the difference Θ_{j2}, and it follows that

$$\begin{aligned} \Theta_{j2} &= G_j(x) - G_j^l(x) \\ &= G_j(x) - \left[G_j(x_{\text{mid}}) + \nabla G_j(x_{\text{mid}})^T (x - x_{\text{mid}})\right] \\ &= \nabla G_j(\gamma)^T (x - x_{\text{mid}}) - \nabla G_j(x_{\text{mid}})^T (x - x_{\text{mid}}) \\ &\le \left\|\nabla^2 G_j(\beta)\right\| \|\gamma - x_{\text{mid}}\| \|x - x_{\text{mid}}\|, \end{aligned} \tag{32}$$

where γ and β are constant vectors, which satisfy

$$\begin{aligned} G_j(x) - G_j(x_{\text{mid}}) &= \nabla G_j(\gamma)^T (x - x_{\text{mid}}), \\ \nabla G_j(\gamma) - \nabla G_j(x_{\text{mid}}) &= \nabla^2 G_j(\beta)(\gamma - x_{\text{mid}}). \end{aligned} \tag{33}$$

Since $\nabla^2 G_j(x)$ is a continuous function, and X is a compact set, there exists some $\widehat{M} > 0$ such that

$$\left\|\nabla^2 G_j(x)\right\| \le \widehat{M}. \tag{34}$$

By (32), we have

$$\Theta_{j2} \le \widehat{M}\left\|\overline{x}^k - \underline{x}^k\right\|^2. \tag{35}$$

Furthermore, we have

$$\Theta_{j2} \longrightarrow 0 \quad \text{as} \quad \left\|\overline{x}^k - \underline{x}^k\right\| \longrightarrow 0. \tag{36}$$

By (19), (31), and (36), we can derive that

$$\begin{aligned} \Theta_j &= \left[\Psi_j(x) - G_j(x)\right] + \left[G_j(x) - G_j^l(x)\right] \\ &= \Theta_{j1} + \Theta_{j2} \longrightarrow 0 \quad \text{as} \quad \left\|\overline{x}^k - \underline{x}^k\right\| \longrightarrow 0. \end{aligned} \tag{37}$$

By the above discussion, it is obvious that the conclusion is followed. □

3. Algorithm and Its Convergence

In this section, based on the former linear relaxation method, we will present a branch and bound algorithm for globally solving problem (EP). There are three fundamental operations in the proposed algorithm: a branching operation, an updating upper bounds operation, and an updating lower bounds operation.

The first fundamental operation iteratively subdivides the investigated rectangle X into two subrectangles. During the process of iteration of the algorithm, the branching operation produces a more refined partition that cannot yet be excluded from further consideration in finding the global optimum for the problem (EP). In this paper, we choose a simple and standard branching rule. This branching rule is enough to ensure the convergence of the algorithm since it drives the intervals shrinking to a singleton for all variables along any infinite branch of the branch and bound tree. Consider any node subproblem identified by the hyperrectangle $X = [\underline{x}, \overline{x}] \subseteq X^0$. This branching rule is as follows.

(a) Let $q = \arg\max\{\overline{x}_i - \underline{x}_i : i = 1, \ldots, n\}$.

(b) Let

$$\widehat{X}^1 = \Big\{x \in R^n \mid \underline{x}_i \le x_i \le \overline{x}_i, \quad i \ne q, \underline{x}_q \le x_q \le \frac{\underline{x}_q + \overline{x}_q}{2}\Big\},$$
$$\widehat{X}^2 = \Big\{x \in R^n \mid \underline{x}_i \le x_i \le \overline{x}_i, i \ne q, \quad \frac{\underline{x}_q + \overline{x}_q}{2} \le x_q \le \overline{x}_q\Big\}. \tag{38}$$

By this branching rule, the rectangle X is subdivided into two subrectangles $\widehat{X}^1$ and $\widehat{X}^2$.

The second fundamental operation is to update the lower bounds of the optimal value of the problem (EP). This main computation needs to solve a sequence of linear relaxation programming problems, which can be easily solved by using the simplex method.

The third fundamental operation is to update the upper bounds of the optimal value of the problem (EP). The upper bounds can be updated by computing the objective function values of the original problem (MLFP) and the equivalent problem (EP) which corresponds to optimal solution of each linear relaxation programming problem, respectively.

The set F in the algorithm is the set of fathomed subrectangles X of X^0. Let $\mathrm{LB}(X^k)$ refer to the optimal objective function value of the problem (LRP) on the sub-hyper-rectangles X^k and $x^k = x(X^k)$ refer to an element of corresponding argmin. The basic steps of the proposed algorithm are summarized as follows.

Algorithm Statement. Consider the following

Step 0. Choose $\epsilon \ge 0$. Let $X^0 = \{x \mid \underline{x}_i \le x_i \le \overline{x}_i, i = 1, \ldots, n\}, UB_0 = +\infty$. Find an optimal solution (x^0, t^0) and the optimal value $v(X^0)$ of the $\mathrm{LRP}(X^0)$. Set

$$\mathrm{LB}_0 := \mathrm{LB}\left(X^0\right) = v\left(X^0\right), \qquad \left(x^c, t^c\right) = \left(x^0, t^0\right). \tag{39}$$

For the given feasible tolerance ϵ_1, if (x^c, t^c) is feasible to the problem $\mathrm{EP}(X^0)$, update $UB_0 = t^c$, if necessary. Set

$$UB_0 = \min\left\{UB_0, \max\left\{\Psi_1\left(x^c\right), \Psi_2\left(x^c\right), \ldots, \Psi_p\left(x^c\right)\right\}\right\}. \tag{40}$$

If $UB_0 - \mathrm{LB}_0 \le \epsilon$, stop; x^c is a global ϵ-optimal solution for the problem (MLFP). Otherwise, let the set of active node $\Omega_0 = X^0$ and $F := \emptyset$, $k = 1$, and go to Step k.

Step k. $k \ge 1$.

Step k1. Set

$$UB_k = UB_{k-1}. \tag{41}$$

Subdivide X^{k-1} into two n-dimensional rectangles $X^{k,1}$, $X^{k,2}$ via the rectangular bisection process. Denote the set of new partitioned rectangles by $\overline{X}^k$. Set $F = F \cup X^{k-1}$.

Step k2. For each $j = 1, 2$, compute $v(X^{k,j})$, and if $v(X^{k,j}) \ne +\infty$, find an optimal solution $(x^{k,j}, t^{k,j})$ for the problem $\mathrm{EQ}(X)$ with $X = X^{k,j}$. For each $j = 1, 2$, set

$$\mathrm{LB}\left(X^{k,j}\right) = v\left(X^{k,j}\right). \tag{42}$$

Set $s = 0$.

Step k3. Set $s = s + 1$. If $s > 2$, go to Step $k7$. Otherwise, continue.

Step k4. If $\mathrm{LB}(X^{k,s}) \ge UB_k$, set $F = F \cup X^{k,s}$ and go to Step $k6$. Otherwise, continue.

Step k5. For the given feasible tolerance ϵ_1, if $(x^{k,s}, t^{k,s})$ is feasible to the $\mathrm{EP}(X^0)$, update $UB_k = t^{k,s}$, if necessary. Let

$$UB_k = \min\left\{UB_k, \max\left\{\Psi_1\left(x^{k,s}\right), \Psi_2\left(x^{k,s}\right), \ldots, \Psi_p\left(x^{k,s}\right)\right\}\right\}. \tag{43}$$

If

$$UB_k < \max\left\{\Psi_1\left(x^{k,s}\right), \Psi_2\left(x^{k,s}\right), \ldots, \Psi_p\left(x^{k,s}\right)\right\}, \tag{44}$$

go to Step $k6$. If

$$UB_k = \max\left\{\Psi_1\left(x^{k,s}\right), \Psi_2\left(x^{k,s}\right), \ldots, \Psi_p\left(x^{k,s}\right)\right\}, \tag{45}$$

set

$$x^c = x^{k,s}, \qquad F = F \cup \left\{X \in \Omega_{k-1} \mid \mathrm{LB}\left(X\right) \ge UB_k\right\}, \tag{46}$$

and continue.

Step k6. Go to Step $k3$.

Step k7. Set

$$\Omega_k = \left\{X \mid X \in \left(\Omega_{k-1} \cup \left\{X^{k,1}, X^{k,2}\right\}\right), X \notin F\right\}. \tag{47}$$

Step k8. Set $\mathrm{LB}_k = \min\{\mathrm{LB}(X) \mid X \in \Omega_k\}$, and let $X^k \in \Omega_k$ satisfying $\mathrm{LB}_k = \mathrm{LB}(X^k)$. If $UB_k - \mathrm{LB}_k \le \epsilon$, stop; x^c is a global ϵ-optimal solution for the problem (MLFP), and $v = \max\{\Psi_1(x^c), \Psi_2(x^c), \ldots, \Psi_p(x^c)\}$ is global optimal value for the problem $\mathrm{EP}(X^0)$. Otherwise, set $k = k + 1$ and go to Step k.

The convergence properties of the proposed algorithm are given as follows.

Theorem 2. *If the proposed algorithm terminates in finite steps, then a global optimal solution of the problem (MLFP) is obtained when the algorithm is terminated.*

Table 1: Computational results for test Examples 1–4.

Example	References	Optimal solution	Iter	$L_{\max}$	Time (s)
Example 1	ours	(1.015569395, 0.591850539, 1.401565828)	1	2	0.00332836
	[12]	(1.015695, 0.590494, 1.403675)	1	—	0.06
	[19]	(1.015678086, 0.590676676, 1.403391837)	6	5	0.01533491
Example 2	ours	(1.5, 1.5)	3	2	0.0025344
	[12]	(1.5, 1.5)	1	—	0.00
	[19]	(1.5, 1.5)	6	7	0.00579627
Example 3	ours	(1.016666667, 0.55, 1.45)	5	2	0.0110939
	[19]	(1.016666667, 0.55, 1.45)	8	8	0.02143792
Example 4	ours	(1.008333333, 0.5, 1.45)	3	2	0.00856142
	[19]	(1.008333333, 0.5, 1.45)	7	8	0.02374296

Proof. Assume that the algorithm is terminated finitely at x^k. Obviously, $\mathrm{LB}_k = UB_k$ when it is terminated at the k_{th} iteration; therefore, x^k is a global optimal solution of the problem (MLFP). □

Theorem 3. *If the algorithm generates an infinite sequence $\{x^k\}$, then every accumulation point x^* of this sequence is a global optimal solution of the problem (MLFP).*

Proof. Let x^* be an accumulation point of the sequence $\{x^k\}$, and let $\{x^{k_q}\}$ be a subsequence $\{x^{k_q}\}$ of the sequence $\{x^k\}$ which is convergent to x^*. Obviously, in the proposed algorithm, the lower bound sequence $\{\mathrm{LB}_k\}$ is monotonic increasing and the upper bound sequence $\{UB_k\}$ is monotonic decreasing, so that $\{\mathrm{LB}_k\}$ and $\{UB_k\}$ are convergent and

$$\begin{aligned}\lim_{k\to\infty} UB_k &= \lim_{q\to\infty} UB_{k_q}\\ &= \lim_{q\to\infty} \max\left\{\Psi_1\left(x^{k_q}\right),\Psi_2\left(x^{k_q}\right),\dots,\Psi_p\left(x^{k_q}\right)\right\}\\ &= \max\left\{\Psi_1\left(x^*\right),\Psi_2\left(x^*\right),\dots,\Psi_p\left(x^*\right)\right\}.\end{aligned} \tag{48}$$

Without loss of generality, we assume that x^{k_q} is the solution of the problem (LRP) on X^{k_q} which satisfies $X^{k_{q+1}} \subseteq X^{k_q}$, $q = 1, 2, \dots$. Because the proposed rectangle partition is exhaustive; that is, $\lim_{q\to\infty} X^{k_q} = x^*$, and, from Theorem 1, we have

$$\begin{aligned}0 &\le UB_{k_q} - LB_{k_q}\\ &\le \max\left\{\left|\Psi_1\left(x^{k_q}\right) - G_1^l\left(x^{k_q}\right)\right|,\right.\\ &\qquad \left.\dots,\left|\Psi_p\left(x^{k_q}\right) - G_p^l\left(x^{k_q}\right)\right|\right\} \longrightarrow 0 \quad \text{as } q \longrightarrow \infty.\end{aligned} \tag{49}$$

Thus,

$$\lim_{q\to\infty}\left(UB_{k_q} - \mathrm{LB}_{k_q}\right) = 0. \tag{50}$$

Hence,

$$\begin{aligned}\lim_{k\to\infty}\mathrm{LB}_k &= \lim_{q\to\infty}\left(UB_{k_q} - \left(UB_{k_q} - \mathrm{LB}_{k_q}\right)\right) = \lim_{q\to\infty} UB_{k_q}\\ &= \max\left\{\Psi_1\left(x^*\right),\Psi_2\left(x^*\right),\dots,\Psi_p\left(x^*\right)\right\}.\end{aligned} \tag{51}$$

Therefore, x^* is a global optimal solution of the problem (MLFP). □

4. Numerical Experiments

To verify the performance of our algorithm, several test examples in recent literatures are implemented on on a Intel(R) Core(TM)2 Duo CPU (1.58 GHZ) microcompute; the algorithm program is coded in C++, and each linear relaxation programming problem is solved by using simplex method, and the convergence tolerance is set to $\epsilon = 5 \times 10^{-8}$ in our experiment. For the test problems, numerical results are illustrated in Table 1. For Examples 1–5, feasible errors ϵ_1 are set by 0.005, 0.005, 0, 0.005, and 0.001, respectively.

In Tables 1 and 2, the notations have been used for column headers: Iter: number of algorithm iteration; $L_{\max}$: the maximal number of algorithm active nodes necessary; time: execution time of algorithm in seconds.

In Table 1, optimal value is denoted by objective function value of the optimal solution in computational procedure of [19] and this paper, respectively.

Example 1 (see [12, 19]). Consider

$$\begin{aligned}\min\max\quad & \left\{\frac{3x_1 + x_2 - 2x_3 + 0.8}{2x_1 - x_2 + x_3}, \frac{4x_1 - 2x_2 + x_3}{7x_1 + 3x_2 - x_3}\right\}\\ \text{s.t.}\quad & x_1 + x_2 - x_3 \le 1,\\ & -x_1 + x_2 - x_3 \le -1,\\ & 12x_1 + 5x_2 + 12x_3 \le 34.8,\\ & 12x_1 + 12x_2 + 7x_3 \le 29.1,\\ & -6x_1 + x_2 + x_3 \le -4.1,\\ & 1.0 \le x_1 \le 1.1,\\ & 0.55 \le x_2 \le 0.65,\\ & 1.35 \le x_3 \le 1.45.\end{aligned} \tag{52}$$

Table 2: Numerical results for Example 5.

Example 5		Our algorithm			Algorithm of [19]	
(p, M, N)	Iter	$L_{\max}$	Time (s)	Iter	$L_{\max}$	Time (s)
(3, 4, 5)	59	16	0.18905	90	88	0.308533
(5, 5, 5)	173	40	0.560235	440	438	2.070590
(5, 5, 6)	211	46	0.996647	328	314	1.954055
(6, 6, 5)	88	18	0.382545	348	348	2.012132
(7, 5, 6)	88	16	0.545829	174	159	1.315186
(30, 6, 6)	74	13	2.21828	151	148	6.439019
(50, 6, 6)	29	2	2.08805	95	81	9.848997
(10, 7, 6)	454	131	3.79913	685	659	7.590868
(7, 7, 7)	154	30	1.30391	242	235	2.713118
(7, 7, 8)	733	175	7.4577	8913	8901	157.815951
(20, 9, 8)	55	6	1.65562	125	123	4.672965
(9, 7, 10)	2243	570	42.8553	3581	3400	79.042373
(10, 10, 10)	258	20	5.69239	450	200	11.414042

Example 2 (see [12, 19]). Consider

$$\begin{aligned} \max \min \quad & \left\{\frac{37x_1 + 73x_2 + 13}{13x_1 + 13x_2 + 13}, \frac{63x_1 - 18x_2 + 39}{13x_1 + 26x_2 + 13}\right\} \\ \text{s.t.} \quad & 5x_1 - 3x_2 = 3, \\ & 0 \le x_1 \le 3. \end{aligned} \tag{53}$$

This example is originally presented in Phuong and Tuy [12], which has the same global optimal solution as the following problem 2″:

$$\begin{aligned} \min \max \quad & \left\{\frac{13x_1 + 13x_2 + 13}{37x_1 + 73x_2 + 13}, \frac{13x_1 + 26x_2 + 13}{63x_1 - 18x_2 + 39}\right\} \\ \text{s.t.} \quad & 5x_1 - 3x_2 = 3, \\ & 0 \le x_1 \le 3. \end{aligned} \tag{54}$$

Therefore, we can use the proposed algorithm to globally solve Example 2 by solving the problem 2″.

Example 3 (see [19]). Consider

$$\begin{aligned} \min \max \quad & \left\{\frac{2x_1 + 2x_2 - x_3 + 0.9}{x_1 - x_2 + x_3}, \frac{3x_1 - x_2 + x_3}{8x_1 + 4x_2 - x_3}\right\} \\ \text{s.t.} \quad & x_1 + x_2 - x_3 \le 1, \\ & -x_1 + x_2 - x_3 \le -1, \\ & 12x_1 + 5x_2 + 12x_3 \le 34.8, \\ & 12x_1 + 12x_2 + 7x_3 \le 29.1, \\ & -6x_1 + x_2 + x_3 \le -4.1, \\ & 1.0 \le x_1 \le 1.2, \\ & 0.55 \le x_2 \le 0.65, \\ & 1.35 \le x_3 \le 1.45. \end{aligned} \tag{55}$$

Example 4 (see [19]). Consider

$$\begin{aligned} \min \max \quad & \left\{\frac{3x_1 + x_2 - 2x_3 + 0.8}{2x_1 - x_2 + x_3}, \frac{4x_1 - 2x_2 + x_3}{7x_1 + 3x_2 - x_3}, \right. \\ & \left. \frac{3x_1 + 2x_2 - x_3 + 1.9}{x_1 - x_2 + x_3}, \frac{4x_1 - x_2 + x_3}{8x_1 + 4x_2 - x_3}\right\} \\ \text{s.t.} \quad & x_1 + x_2 - x_3 \le 1, \\ & -x_1 + x_2 - x_3 \le -1, \\ & 12x_1 + 5x_2 + 12x_3 \le 34.8, \\ & 12x_1 + 12x_2 + 7x_3 \le 29.1, \\ & -6x_1 + x_2 + x_3 \le -4.1, \\ & 1.0 \le x_1 \le 1.2, \\ & 0.55 \le x_2 \le 0.65, \\ & 1.35 \le x_3 \le 1.45. \end{aligned} \tag{56}$$

Example 5. Consider

$$\begin{aligned} \min \max \quad & \left\{\frac{\sum_{i=1}^{n} c_{1i}x_i + d_1}{\sum_{i=1}^{n} e_{1i}x_i + f_1}, \frac{\sum_{i=1}^{n} c_{2i}x_i + d_2}{\sum_{i=1}^{n} e_{2i}x_i + f_2}, \right. \\ & \left. \ldots, \frac{\sum_{i=1}^{n} c_{pi}x_i + d_p}{\sum_{i=1}^{n} e_{pi}x_i + f_p}\right\} \\ \text{s.t.} \quad & Ax \le b, \\ & 0 \le x_i \le 3, \quad i = 1, \ldots, n, \end{aligned} \tag{57}$$

where A is an $m \times n$ matrix, b is an m dimension vector, all elements of c_{ji}, e_{ji}, $j = 1, \ldots, p$, $i = 1, \ldots, n$, are randomly generated between 0 and 1; all elements of d_j, f_j, $j = 1, \ldots, p$ are randomly generated between 0 and p; all elements of A are randomly generated between 0 and 1; all elements of b are randomly generated between 0 and 16.

In Table 2, the notations have been also used for column headers: p: the number of linear fractional functions in the objective function; m: represents the number of rows for A; n: stands for the dimension of considered problem. From the experimental results in Table 2, it is seen that our algorithm can effectively solve the minimax linear fractional programming problem (MLFP) with large scale number of fractional functions.

5. Concluding Remarks

In this paper, an effective branch and bound algorithm is proposed for globally solving the minimax linear fractional programming problem (MLFP). In this algorithm, the lower bound is computed during the branch and bound search by solving linear relaxation programming (LRP) of the problem (MLFP), which can be derived by using the new linear relaxation bounding technique and can be effectively solved by using the simplex method or interior point algorithm. The proposed algorithm is convergent to the global minimum of the problem (MLFP) through the successive refinement of the feasible region and solutions of a series of the LRP. Numerical results for several test problems have been reported to show the feasibility and effectiveness of the proposed algorithm. It is hoped that the ideas and methods used to construct the algorithm will offer useful tools for solving the minimax linear fractional programming problem.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China under Grant (11171094) and the Science and Technology Key Project of Education Department of Henan Province (14A110024).

References

[1] S. Schaible and J. Shi, "Recent developments in fractional programming: single ratio and maxmin cale," in *Proceedings of the 3rd International Conference on Nonlinear Analysis and Convex Analysis*, pp. 493–506, 2003.

[2] S. Schaible, "Fractional programming," in *Handbook of Global Optimization*, R. Horst and P. M. Pardalos, Eds., vol. 168, pp. 495–608, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1995.

[3] I. M. Stancu-Minasian, *Fractional Programming: Theory, Methods and Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[4] I. Barrodale, "Best rational approximation and strict quasi-convexity," *SIAM Journal on Numerical Analysis*, vol. 10, no. 1, pp. 8–12, 1973.

[5] M. H. Goedhart and J. Spronk, "Financial planning with fractional goals," *European Journal of Operational Research*, vol. 82, no. 1, pp. 111–124, 1995.

[6] R. Jagannathan and S. Schaible, "Duality in generalized fractional programming via Farkas' lemma," *Journal of Optimization Theory and Applications*, vol. 41, no. 3, pp. 417–424, 1983.

[7] A. I. Barros, J. B. G. Frenk, S. Schaible, and S. Zhang, "Using duality to solve generalized fractional programming problems," *Journal of Global Optimization*, vol. 8, no. 2, pp. 139–170, 1996.

[8] J. P. Crouzeix, J. A. Ferland, and S. Schaible, "An algorithm for generalized fractional programs," *Journal of Optimization Theory and Applications*, vol. 47, no. 1, pp. 35–49, 1985.

[9] M. Gugat, "A fast algorithm for a class of generalized fractional programs," *Management Science*, vol. 42, no. 10, pp. 1493–1499, 1996.

[10] R. W. Freund and F. Jarre, "An interior-point method for fractional programs with convex constraints," *Mathematical Programming*, vol. 67, pp. 407–440, 1994.

[11] Y. E. Nesterov and A. S. Nemirovskii, "An interior-point method for generalized linear-fractional programming," *Mathematical Programming*, vol. 69, pp. 177–204, 1995.

[12] N. T. H. Phuong and H. Tuy, "A unified monotonic approach to generalized linear fractional programming," *Journal of Global Optimization*, vol. 26, no. 3, pp. 229–259, 2003.

[13] C. Audet, P. Hansen, and E. Carrizosa, "An exact method for fractional goal programming," *Journal of Global Optimization*, vol. 29, no. 1, pp. 113–120, 2004.

[14] M. Locatelli, "Approximation algorithm for a class of global optimization problems," *Journal of Global Optimization*, vol. 55, no. 1, pp. 13–25, 2013.

[15] A. I. Barros and J. B. G. Frenk, "Generalized fractional programming and cutting plane algorithms," *Journal of Optimization Theory and Applications*, vol. 87, no. 1, pp. 103–120, 1995.

[16] A. Roubi, "Method of centers for generalized fractional programming," *Journal of Optimization Theory and Applications*, vol. 107, no. 1, pp. 123–143, 2000.

[17] J.-J. Strodiot, J.-P. Crouzeix, J. A. Ferland, and V. H. Nguyen, "An inexact proximal point method for solving generalized fractional programs," *Journal of Global Optimization*, vol. 42, no. 1, pp. 121–138, 2008.

[18] J. C. Bernard and J. A. Ferland, "Convergence of interval-type algorithms for generalized fractional programming," *Mathematical Programming A*, vol. 43, no. 1–3, pp. 349–363, 1989.

[19] Q. Feng, H. Jiao, H. Mao, and Y. Chen, "A deterministic algorithm for min-max and max-min linear fractional programming problems," *International Journal of Computational Intelligence Systems*, vol. 4, no. 2, pp. 134–141, 2011.