Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 18, No. 1, pp. 195–212, 2022

IMAGE SPACE BRANCH-AND-BOUND ALGORITHM FOR GLOBALLY SOLVING MINIMAX LINEAR FRACTIONAL PROGRAMMING PROBLEM*

Hongwei Jiao†, Junqiao Ma and Youlin Shang

Abstract: This paper presents an image space branch-and-bound algorithm for a minimax linear fractional programming problem (MLFP), which is widely used in data envelopment analysis, system identification and so on. In this algorithm, based on equivalent transformation and new linearizing technique, we convert the original problem into a linear relaxation programming problem, which can be used to calculate the lower bound of the optimal value of the original problem. By subsequently refining the initial image space region and successively solving a series of linear relaxation problems, the proposed algorithm is globally convergent to the optimal solution of the problem (MLFP). By analyzing the computational complexity of the algorithm, we give a maximum evaluation of number of iterations of the algorithm for the first time. Finally, numerical experimental results demonstrate the feasibility and effectiveness of the proposed algorithm.

Key words: *minimax linear fractional programming, global optimization, linearizing technique, image space branch-and-bound algorithm, computational complexity*

Mathematics Subject Classification: *90C26, 90C32*

---

1 Introduction

In this paper, we consider the following minimax linear fractional programming problem

$$\text{(MLFP)}: \begin{cases} \min \max & \left\{ \dfrac{\sum_{j=1}^{n} d_{1j}x_j + g_1}{\sum_{j=1}^{n} e_{1j}x_j + h_1}, \dfrac{\sum_{j=1}^{n} d_{2j}x_j + g_2}{\sum_{j=1}^{n} e_{2j}x_j + h_2}, \ldots, \dfrac{\sum_{j=1}^{n} d_{pj}x_j + g_p}{\sum_{j=1}^{n} e_{pj}x_j + h_p} \right\} \\ \text{s.t.} & x \in D = \{x \in R^n | Ax \leq b,\ x \geq 0\}, \end{cases}$$

where $p \geq 2$, $A \in R^{m\times n}$, $b \in R^m$, $D \neq \emptyset$, $\sum_{j=1}^{n} d_{ij}x_j + g_i$ and $\sum_{j=1}^{n} e_{ij}x_j + h_i$ are all bounded linear functions defined on D, and for any $x \in D$, the denominator $\sum_{j=1}^{n} e_{ij}x_j + h_i \neq 0, i = 1, 2, \ldots, p$.

*This work was supported by the National Natural Science Foundation of China (11871196; 12071133; 12071112), the China Postdoctoral Science Foundation (2017M622340), the Key Scientific and Technological Research Projects in Henan Province (202102210147; 192102210114).

†Corresponding author.

The problem (MLFP) has aroused interest of practitioners and researchers for many years. On the one hand, the problem (MLFP) has a wide range of applications in many fields, such as data envelopment analysis, electronic circuit design [2], system identification [11, 12, 34], optimal design [3], signal processing [13], iterative parameters estimation [14]. On the other hand, the problem (MLFP) is a nonconvex global optimization problem, which may have multiple local optimal solutions that are not global optimal solutions, it is still challenging to solve this problem. Therefore, it is necessary to design an effective algorithm to globally solve the problem (MLFP).

In the past few decades, some algorithms have been proposed for solving the problem (MLFP) and its special form. According to the characteristic structures of these algorithms, which can be classified into the following categories: interior-point algorithm [15], parameter programming method [9], partial linearization algorithm [16], monotonic optimization method [27], cutting plane algorithm [4], branch-and-bound algorithm [17, 21]. Recently, Ghazi and Roubi [21] proposed a DC method for solving the minimax fractional programming problem with ratios of convex functions; Addoune et al. [1] presented a proximal point algorithm to solve the generalized fractional programming problem; Based on the proximal bundle method, Boualam and Roubi [5] designed a dual algorithm for convex minimax fractional programming problem; Boufi and Roubi [6] gave a dual method of centers for the generalized fractional programs; Boufi and Haffari [7] presented some optimization conditions and a method of centers for the minimax fractional programs with difference of convex functions; Smail et al. [29] presented a proximal bundle algorithm for solving the nonlinear constrained convex minimax fractional programs; Roubi and Haffari [28] proposed a prox-dual regularization algorithm for solving the generalized fractional programs; Boufi and Roubi [8] gave some duality results and a dual bundle method for solving the minimax fractional programming problem; Chen et al. [10] proposed a generic algorithm for generalized fractional programming problem. However, the above reviewed methods can only deal with particular forms of the problem (MLFP), or they are difficult to solve large-scale practical problems. Therefore, it is still necessary to propose a practical efficient algorithm for solving the general form of the problem (MLFP).

In addition to the algorithms reviewed above, some optimality conditions and duality theorems for the minimax fractional programming problem have also been obtained. For example, Li et al. [23] presented the optimality condition and duality for the minimax fractional programming problems with data uncertainty; Lai and Huang [24] gave the dual theorem for the nondifferentiable minimax fractional programming problem under the generalized convex condition; Lai and Liu [25] gave the Kuhn-Tucker type sufficient optimal conditions for complex minimax fractional programming problem under generalized convex functions; Gao and Rong [19] gave the optimality conditions and duality for a class of nondifferentiable multi-objective generalized fractional programming problems; Lai et al. [26] proposed some necessary and sufficient optimal conditions for the minimax fractional programs. For an excellent review of algorithms and theories for the minimax fractional programming problem, the reader can be referred to Schaible and Shi [30] and Stancu-Minasian [31, 32].

In this paper, we present an image space branch-and-bound algorithm for globally solving the problem (MLFP). In the algorithm, first of all, by introducing new variables, an equivalent problem (EP1) of the problem (MLFP) is constructed. Next, we present a new linearizing technique for constructing the linear relaxation programming problem of the problem (EP1). By subsequently refining the initial image space region, and by solving a series of linear relaxation programming problems, the proposed algorithm is globally convergent to the optimal solution of the problem (MLFP). Finally, the numerical results indicate that the proposed algorithm can effectively find the globally optimal solutions of all test examples with any given tolerance ϵ.

Compared with the existing branch-and-bound algorithms [16, 21, 35], the novelty and main contributions of this paper are given as follows. (1) The branching search of the proposed algorithm takes place in the image space R^p of the linear fractional functions $\frac{\sum_{j=1}^{n} d_{ij}x_j+g_i}{\sum_{j=1}^{n} e_{ij}x_j+h_i}, i=1,\ldots,p$, rather than in the variable dimension space R^n, where n usually far exceeds p, which may mitigate the required computational efforts of the proposed algorithm. (2) By analysing the computational complexity of the proposed algorithm, we give a maximum estimation of number of iterations of the algorithm for the first time, which is not available in other literatures [16, 21, 35]. (3) Numerical results show that the proposed algorithm has higher computational efficiency than some existing algorithms [16, 21, 35].

The outline of this article is as follows. In section 2, we transform the original problem into an equivalent problem, and a linearizing technique is proposed for constructing the linear relaxation programming problem (LRP) of the problem (MLFP). In section 3, we present an image space branch-and-bound algorithm, and give the global convergence of the algorithm. In section 4, we give a maximum estimation of number of iterations of the algorithm by analysing the computational complexity of the algorithm. In Section 5, the numerical experiments are reported. Finally, some conclusions are given in Section 6.

2 Equivalent Problem and Its Linear Relaxation

In the following, we firstly convert the problem (MLFP) into an equivalent problem. For this purpose, for each $i=1,\ldots,p$, we need to calculate the minimum value $L_i^0=\min_{x\in D}\frac{\sum_{j=1}^{n} d_{ij}x_j+g_i}{\sum_{j=1}^{n} e_{ij}x_j+h_i}$ and the maximum value $U_i^0=\max_{x\in D}\frac{\sum_{j=1}^{n} d_{ij}x_j+g_i}{\sum_{j=1}^{n} e_{ij}x_j+h_i}$ of the linear fractional function $\frac{\sum_{j=1}^{n} d_{ij}x_j+g_i}{\sum_{j=1}^{n} e_{ij}x_j+h_i}$ over D. To obtain the values of L_i^0 and $U_i^0, i=1,\ldots,p$, here, we firstly consider the following linear fractional programming problem:

$$L_i^0=\min_{x\in D}\frac{\sum_{j=1}^{n} d_{ij}x_j+g_i}{\sum_{j=1}^{n} e_{ij}x_j+h_i}, i=1,2,\ldots,p. \tag{2.1}$$

As everyone knows, the linear fractional function $\frac{\sum_{j=1}^{n} d_{ij}x_j+g_i}{\sum_{j=1}^{n} e_{ij}x_j+h_i}$ is a quasi-convex function, so that it can attain the minimum value at some vertex of D. Since the denominator $\sum_{j=1}^{n} e_{ij}x_j+h_i\neq 0$, we have $\sum_{j=1}^{n} e_{ij}x_j+h_i<0$ or $\sum_{j=1}^{n} e_{ij}x_j+h_i>0$, when $\sum_{j=1}^{n} e_{ij}x_j+h_i<0$, since $\frac{\sum_{j=1}^{n} d_{ij}x_j+g_i}{\sum_{j=1}^{n} e_{ij}x_j+h_i}=\frac{-(\sum_{j=1}^{n} d_{ij}x_j+g_i)}{-(\sum_{j=1}^{n} e_{ij}x_j+h_i)}$, then without loss of generality, we can always suppose that $\sum_{j=1}^{n} e_{ij}x_j+h_i>0$. To solve the problem (2.1), for any $i\in\{1,2,\ldots,p\}$, by introducing new variables $t_i = \frac{1}{\sum\limits_{j=1}^{n} e_{ij}x_j + h_i}$, and let $z_j = t_i x_j$, then we can convert the problem (2.1) into the following equivalent linear programming problem:

$$\begin{cases} \min & \sum\limits_{j=1}^{n} d_{ij}z_j + g_i t_i \\ \text{s.t.} & \sum\limits_{j=1}^{n} e_{ij}z_j + h_i t_i = 1 \\ & Az \leq bt_i. \end{cases} \tag{2.2}$$

Remark 2.1. $x^* \in R^n$ is the global optimal solution of the problem (2.1) if and only if $(z^*, t_i^*) \in R^{n+1}$ is the global optimal solution of the problem (2.2), the problems (2.1) and (2.2) have the equal global optimal values. Therefore, for each $i = 1, 2, \ldots, p$, L_i^0 can be obtained by solving the linear programming problem (2.2).

Similarly, we can get the maximum value $U_i^0 = \max\limits_{x \in D} \frac{\sum\limits_{j=1}^{n} d_{ij}x_j + g_i}{\sum\limits_{j=1}^{n} e_{ij}x_j + h_i}$ of the linear fractional function $\frac{\sum\limits_{j=1}^{n} d_{ij}x_j + g_i}{\sum\limits_{j=1}^{n} e_{ij}x_j + h_i}, i = 1, \ldots, p$. So we can obtain the initial image space rectangle $\Omega^0 = \{y \in R^p \mid L_i^0 \leq y_i \leq U_i^0,\ i = 1, \ldots, p\}$.

Next, by introducing new variables r and $y_i = \frac{\sum\limits_{j=1}^{n} d_{ij}x_j + g_i}{\sum\limits_{j=1}^{n} e_{ij}x_j + h_i}, i = 1, 2, \ldots, p$, we can get the equivalent problem (EP) of the problem (MLFP) as follows.

$$(\text{EP}): \begin{cases} \min & r \\ \text{s.t.} & y_i - r \leq 0, i = 1, 2, \ldots, p \\ & y_i = \frac{\sum\limits_{j=1}^{n} d_{ij}x_j + g_i}{\sum\limits_{j=1}^{n} e_{ij}x_j + h_i}, i = 1, 2, \ldots, p \\ & x \in D, y \in \Omega^0. \end{cases}$$

It is easy to know that the feasible region

$$K = \left\{ (x, y, r) \in R^{n+p+1} \mid \ y_i - r \leq 0,\ \frac{\sum\limits_{j=1}^{n} d_{ij}x_j + g_i}{\sum\limits_{j=1}^{n} e_{ij}x_j + h_i} - y_i = 0, i = 1, \ldots, p, x \in D, y \in \Omega^0 \right\}$$

of the problem (EP) is a nonempty bounded compact set, and the feasible region $K \neq \emptyset$ if and only if $D \neq \emptyset$.

In the following, for globally solving the problem (MLFP), we can solve its equivalent problem (EP) instead, and the problems (MLFP) and (EP) have the same global minimum value.

Since the denominator $\sum\limits_{j=1}^{n} e_{ij}x_j + h_i \neq 0$, the problem (EP) can be rewritten into the following equivalent problem (EP1), which has the same global optimal solution and optimal value as the problem (EP).

$$(\mathrm{EP1}):\begin{cases}\min & r\\ \text{s.t.} & y_i - r \le 0, i = 1,2,\dots,p\\ & y_i\left(\sum\limits_{j=1}^{n} e_{ij}x_j + h_i\right) = \sum\limits_{j=1}^{n} d_{ij}x_j + g_i, i = 1,2,\dots,p\\ & x \in D, \quad y \in \Omega^0.\end{cases}$$

To globally solve the problem (EP1), we need to construct its linear relaxation programming problem, which can provide a reliable lower bound of the global minimum value of the problem (EP1) in the branch-and-bound search. The detailed derivation process of the linear relaxation programming problem is given below.

For any $y \in \Omega = \{y \in R^p \mid L_i \le y_i \le U_i,\ i = 1,\dots,p\} \subseteq \Omega^0$, we have

$$y_i\left(\sum_{j=1}^{n} e_{ij}x_j + h_i\right) \ge \sum_{j=1,e_{ij}>0}^{n} e_{ij}L_i x_j + \sum_{j=1,e_{ij}<0}^{n} e_{ij}U_i x_j + h_i y_i$$

and

$$y_i\left(\sum_{j=1}^{n} e_{ij}x_j + h_i\right) \le \sum_{j=1,e_{ij}>0}^{n} e_{ij}U_i x_j + \sum_{j=1,e_{ij}<0}^{n} e_{ij}L_i x_j + h_i y_i.$$

Based on the above conclusions, for any $\Omega \subseteq \Omega^0$, we can construct the linear relaxation programming problem (LRP) of the problem (EP1) as follows.

$$(\mathrm{LRP}):\begin{cases}\min & r\\ \text{s.t.} & y_i - r \le 0,\ i = 1,2,\dots,p\\ & \sum\limits_{j=1,e_{ij}>0}^{n} e_{ij}L_i x_j + \sum\limits_{j=1,e_{ij}<0}^{n} e_{ij}U_i x_j + h_i y_i \le \sum\limits_{j=1}^{n} d_{ij}x_j + g_i,\ i = 1,\dots,p,\\ & \sum\limits_{j=1,e_{ij}>0}^{n} e_{ij}U_i x_j + \sum\limits_{j=1,e_{ij}<0}^{n} e_{ij}L_i x_j + h_i y_i \ge \sum\limits_{j=1}^{n} d_{ij}x_j + g_i,\ i = 1,\dots,p,\\ & x \in D,\ y \in \Omega.\end{cases}$$

Through the above derivation process, we know that, for any $\Omega \subseteq \Omega^0$, all feasible points of the problem (EP1) over the Ω are feasible to the problem (LRP) over the Ω, and the optimal value of the problem (LRP) over the Ω is less than or equal to that of the problem (EP1) over the Ω. Therefore, the optimal value of the problem (LRP) over the Ω can provide a valid lower bound for the minimum value of the problem (EP1) over the Ω during the branch-and-bound search.

Next, we will prove that the problem (LRP) will infinitely approximate the problem (EP1) over Ω as $\|U - L\| \to 0$.

Consequently, without loss of generality, for any $y \in \Omega \subseteq \Omega^0$, define the following functions for convenience in expression, let

$$\varphi_i(x, y_i) = y_i \left(\sum_{j=1}^{n} e_{ij} x_j + h_i \right) = \sum_{j=1}^{n} e_{ij} y_i x_j + h_i y_i,$$

$$\underline{\varphi}_i(x, y_i) = \sum_{j=1, e_{ij}>0}^{n} e_{ij} L_i x_j + \sum_{j=1, e_{ij}<0}^{n} e_{ij} U_i x_j + h_i y_i,$$

$$\overline{\varphi}_i(x, y_i) = \sum_{j=1, e_{ij}>0}^{n} e_{ij} U_i x_j + \sum_{j=1, e_{ij}<0}^{n} e_{ij} L_i x_j + h_i y_i.$$

Obviously, for any $i \in \{1, 2, \ldots, p\}$, we have $\underline{\varphi}_i(x, y_i) \leq \varphi_i(x, y_i) \leq \overline{\varphi}_i(x, y_i)$.

Theorem 2.2. *For each $i \in \{1, 2, \ldots, p\}$, we have*

$$|\varphi_i(x, y_i) - \underline{\varphi}_i(x, y_i)| \to 0 \quad \textit{as} \quad \|U - L\| \to 0$$

and

$$|\overline{\varphi}_i(x, y_i) - \varphi_i(x, y_i)| \to 0 \quad \textit{as} \quad \|U - L\| \to 0.$$

Proof. From the definitions of the functions $\underline{\varphi}_i(x, y_i)$, $\varphi_i(x, y_i)$, and $\overline{\varphi}_i(x, y_i)$, for any $x \in D, y_i \in [L_i, U_i]$, we have that

$$\begin{aligned}
\left|\varphi_i(x, y_i) - \underline{\varphi}_i(x, y_i)\right| &= \left| y_i \left(\sum_{j=1}^{n} e_{ij} x_j + g_i \right) - [\sum_{j=1, e_{ij}>0}^{n} e_{ij} L_i x_j + \sum_{j=1, e_{ij}<0}^{n} e_{ij} U_i x_j + g_i y_i] \right| \\
&= \left| \sum_{j=1, e_{ij}>0}^{n} (y_i - L_i) e_{ij} x_j + \sum_{j=1, e_{ij}<0}^{n} (U_i - y_i) e_{ij} x_j \right| \\
&\leq (U_i - L_i) \times \left| \sum_{j=1, e_{ij}>0}^{n} e_{ij} x_j + \sum_{j=1, e_{ij}<0}^{n} e_{ij} x_j \right| \\
&= (U_i - L_i) \left(\sum_{j=1}^{n} e_{ij} x_j \right).
\end{aligned}$$

Since $\sum_{j=1}^{n} e_{ij} x_j$ is a bounded function, we have that

$$|\varphi_i(x, y_i) - \underline{\varphi}_i(x, y_i)| \to 0 \text{ as } \|U - L\| \to 0.$$

Similarly, we can prove that

$$|\overline{\varphi}_i(x, y_i) - \varphi_i(x, y_i)| \to 0 \ \text{ as } \|U - L\| \to 0,$$

and the proof of the theorem is finished. □

From Theorem 2.2, it follows that the functions $\underline{\varphi}_i(x, y_i)$ and $\overline{\varphi}_i(x, y_i)$ can infinitely approximate the function $\varphi_i(x, y_i)$ as $\|U - L\| \to 0$, which guarantees the global convergence of the branch-and-bound algorithm.

3 Algorithm and Its Convergence Analysis

In this section, we firstly introduce a rectangle branching rule. Next, based the former linear relaxation programming problem and the branch-and-bound framework, we give an image space branch-and-bound algorithm for globally solving the problem (MLFP). Meanwhile, we prove the global convergence of the algorithm and analyse the computational complexity of the algorithm.

$\boxed{3.1}$ Branching rule

For each iteration of the algorithm, the branching process will generate more refined partitions to find the optimal solution of the problem (EP1). Here, we choose the simplest standard bisection rule to subdivide the image space R^p of the linear fractional functions $\frac{\sum_{j=1}^{n} d_{ij}x_j+g_i}{\sum_{j=1}^{n} e_{ij}x_j+h_i}, i=1,\dots,p$, over D. This guarantees the global convergence of the proposed algorithm. For any given image space rectangle

$$\Omega=\{y\in R^p \mid L_i\le y_i\le U_i, i=1,\dots,p\}\subseteq\Omega^0$$

The selected branching rule is given as follows. Let

$$\lambda=\arg\max\{U_i-L_i, i=1,2,\dots,p\},$$

subdivide Ω into the following two sub-rectangles:

$$\hat{\Omega}^1=\{y\in R^p|L_i\le y_i\le\frac{L_i+U_i}{2}, i=\lambda;\ L_i\le y_i\le U_i, i=1,2,\dots,p, i\ne\lambda\}$$

and

$$\hat{\Omega}^2=\{y\in R^p|\frac{L_i+U_i}{2}\le y_i\le U_i, i=\lambda;\ L_i\le y_i\le U_i, i=1,2,\dots,p, i\ne\lambda\}.$$

As stated in [20], it is clear that, the proposed branching rule is exhaustive, and there will generate a nested rectangular subsequence $\{\Omega^k\}$.

$\boxed{3.2}$ Image space branch-and-bound algorithm

In this subsection, we will present an image space branch-and-bound algorithm for globally solving the problem (MLFP). The basic steps of the proposed image space branch-and-bound algorithm are given as follows.

Step 1. Given the approximation error $\epsilon>0$ and the initial rectangle Ω^0. By solving the problem (LRP) over Ω^0, we can obtain the optimal solution $(x^0,\hat{y}^0)$ and optimal value $r(x^0,\hat{y}^0)$ of the problem (LRP) over Ω^0.

Let $LB_0=r\left(x^0,\hat{y}^0\right)$ and $y_i^0=\frac{\sum_{j=1}^{n} d_{ij}x_j^0+g_i}{\sum_{j=1}^{n} e_{ij}x_j^0+h_i}, i=1,2,\dots,p.$

Let

$$UB_0=\max\left\{\frac{\sum_{j=1}^{n} d_{1j}x_j^0+g_1}{\sum_{j=1}^{n} e_{1j}x_j^0+h_1},\frac{\sum_{j=1}^{n} d_{2j}x_j^0+g_2}{\sum_{j=1}^{n} e_{2j}x_j^0+h_2},\cdots,\frac{\sum_{j=1}^{n} d_{pj}x_j^0+g_p}{\sum_{j=1}^{n} e_{pj}x_j^0+h_p}\right\}.$$

If $UB_0-LB_0\le\epsilon$, then the algorithm stops, and x^0 is a global ϵ-optimal solution for the problem (MLFP).

Otherwise, denote $T^0=\left\{\left(x^0,y^0\right)\right\}$ as the set of feasible points, let $k=0$, let $C_0=\left\{\Omega^0\right\}$ be the set of all active nodes, and let $F=\{x^0\}$.

Step 2. Use the proposed branching rule to subdivide Ω^k into two sub-rectangles $\Omega^{k,1}$ and $\Omega^{k,2}$, and let $H=\left\{\Omega^{k,1},\Omega^{k,2}\right\}$.

Step 3. For each rectangle $\Omega^{k,\alpha}, \alpha = 1, 2$, solve the problem (LRP) over $\Omega^{k,\alpha}$ to obtain its optimal solution $(x(\Omega^{k,\alpha})), \hat{y}(\Omega^{k,\alpha}))$ and optimal value $r(x(\Omega^{k,\alpha})), \hat{y}(\Omega^{k,\alpha}))$, and let $LB(\Omega^{k,\alpha}) = r(x(\Omega^{k,\alpha})), \hat{y}(\Omega^{k,\alpha}))$.

If $LB(\Omega^{k,\alpha}) > UB_k$, then let $H = H \backslash \Omega^{k,\alpha}$.

Otherwise, let

$$UB(\Omega^{k,\alpha}) = \max\left\{ \frac{\sum_{j=1}^{n} d_{1j} x(\Omega^{k,\alpha}))_j + g_1}{\sum_{j=1}^{n} e_{1j} x(\Omega^{k,\alpha}))_j + h_1}, \frac{\sum_{j=1}^{n} d_{2j} x(\Omega^{k,\alpha}))_j + g_2}{\sum_{j=1}^{n} e_{2j} x(\Omega^{k,\alpha}))_j + h_2}, \cdots, \frac{\sum_{j=1}^{n} d_{pj} x(\Omega^{k,\alpha}))_j + g_p}{\sum_{j=1}^{n} e_{pj} x(\Omega^{k,\alpha}))_j + h_p} \right\},$$

update the upper bound by letting

$$UB_k = \min\left\{ UB_{k-1}, UB(\Omega^{k,\alpha}) \right\},$$

denote x^k as the best feasible solution corresponding to the minimum upper bound, and let $y_i^k = \frac{\sum_{j=1}^{n} d_{ij} x_j^k + g_i}{\sum_{j=1}^{n} e_{ij} x_j^k + h_i}$, $i = 1, 2, \ldots, p$. Obviously, (x^k, y^k) is the best feasible solution for the problem (EP1).

Let $C_k = \left(C_k \backslash \Omega^k\right) \cup H$, $LB_k = \min\{LB(\Omega) \mid \Omega \in C_k\}$, $T^k = T^{k-1} \cup \left\{\left(x^k, y^k\right)\right\}$, and $F = F \cup \{x^k\}$.

Step 4. Let $C_{k+1} = \{\Omega \mid UB(\Omega) - LB_k > \epsilon, \Omega \in C_k\}$.

If $C_{k+1} \doteq \emptyset$, then the algorithm terminates with that x^k is a global ϵ-optimal solution for the problem (MLFP). Otherwise, select the rectangle Ω^{k+1} such that $\Omega^{k+1} = \arg\min_{\Omega \in C_{k+1}} LB(\Omega)$, which will be subdivided in next iteration, let $k = k + 1$, and return to Step 2.

$\boxed{3.3}$ Convergence analysis

In the subsection, by using the infinite partition of the image space rectangle of the linear fractional functions $\frac{\sum_{j=1}^{n} d_{ij} x_j + g_i}{\sum_{j=1}^{n} e_{ij} x_j + h_i}, i = 1, \ldots, p$, based on the structural characteristics of the branch-and-bound algorithm framework, combined with the updating method of the upper and lower bounds and the continuity of objective function, we prove the global convergence of the image space branch-and-bound algorithm for the first time.

Without losing of generality, we denote r^* as the global optimization value of the problem (EP1), denote r^k as the objective function value of the problem (EP1) corresponding to (x^k, y^k), and define

$$\Psi(x) = \max\left\{ \frac{\sum_{j=1}^{n} d_{1j} x_j + g_1}{\sum_{j=1}^{n} e_{1j} x_j + h_1}, \frac{\sum_{j=1}^{n} d_{2j} x_j + g_2}{\sum_{j=1}^{n} e_{2j} x_j + h_2}, \cdots, \frac{\sum_{j=1}^{n} d_{pj} x_j + g_p}{\sum_{j=1}^{n} e_{pj} x_j + h_p} \right\},$$

the global convergence of the presented image space branch-and-bound algorithm can be given by the following theorem.

Theorem 3.1. *For any given approximation error ϵ, the presented algorithm either finitely terminates at a global ϵ-optimal solution for the problem (MLFP), or generates an infinite* *solution sequence $\{x^k\}$, and any of its accumulation point is the global optimal solution for the problem (MLFP).*

Proof. Suppose that the presented algorithm finitely terminates after k iterations, x^k and (x^k, y^k) are better feasible solutions of the problem (MLFP) and the problem (EP1), respectively. From the termination conditions of the algorithm, the updating methods of the lower bound and upper bound, and the steps of the branch-and-bound algorithm, we can obtain that

$$LB_k \leq r^*,\ r^* \leq r(x^k, y^k),\ \Psi(x^k) = r(x^k, y^k) = r^k,\ r^k - \epsilon \leq LB_k.$$

Thus, we have

$$\Psi(x^k) - \epsilon = r(x^k, y^k) - \epsilon \leq LB_k \leq r^* \leq r(x^k, y^k) = \Psi(x^k).$$

Therefore, we get that x^k is a global ϵ-optimum solution for the problem (MLFP).

Assume that the sequences $\{x^k\}$ and $\{(x^k, y^k)\}$ are generated by the algorithm, and which are the infinite solution sequences of the problem (MLFP) and the problem (EP1), where

$$y_i^k = \frac{\sum\limits_{j=1}^{n} d_{ij}x_j^k + g_i}{\sum\limits_{j=1}^{n} e_{ij}x_j^k + h_i}, i = 1, \ldots, p.$$

Let x^* be an accumulation point of the sequence $\{x^k\}$, without losing generality, suppose that

$$\lim_{k\to\infty} x^k = x^*.$$

Since the continuity of the linear fractional function $\frac{\sum_{j=1}^{n} d_{ij}x_j+g_i}{\sum_{j=1}^{n} e_{ij}x_j+h_i}$, $\frac{\sum_{j=1}^{n} d_{ij}x_j^k+g_i}{\sum_{j=1}^{n} e_{ij}x_j^k+h_i} = y_i^k \in [L_i^k, U_i^k], i = 1, 2, \ldots, p$, and the exhaustiveness of the rectangle branching method, we have the following conclusions:

$$\frac{\sum\limits_{j=1}^{n} d_{ij}x_j^* + g_i}{\sum\limits_{j=1}^{n} e_{ij}x_j^* + h_i} = \lim_{k\to\infty} \frac{\sum\limits_{j=1}^{n} d_{ij}x_j^k + g_i}{\sum\limits_{j=1}^{n} e_{ij}x_j^k + h_i} = \lim_{k\to\infty} y_i^k = \lim_{k\to\infty} [L_i^k, U_i^k] = \lim_{k\to\infty} \bigcap_k [L_i^k, U_i^k] = y_i^*.$$

Above that, since (x^*, y^*) is a feasible solution to the problem (EP1) and $\{LB_k\}$ is an increasing lower bound sequence such that $LB_k \leq r^*$, we have that

$$r(x^*, y^*) \geq r^* \geq \lim_{k\to\infty} LB_k = \lim_{k\to\infty} r(x^k, y^k) = r(x^*, y^*). \tag{3.1}$$

Hence, from the renewing method of the upper bound and the continuity of the function $\Psi(x)$, we can get the following conclusions:

$$\lim_{k\to\infty} r^k = \lim_{k\to\infty} r(x^k, y^k) = r(x^*, y^*) = \Psi(x^*) = \lim_{k\to\infty} \Psi(x^k). \tag{3.2}$$

By combing the above inequalities (3.1) and (3.2), we have that

$$\lim_{k\to\infty} r^k = r^* = \Psi(x^*) = \lim_{k\to\infty} \Psi(x^k) = r(x^*, y^*) = \lim_{k\to\infty} LB_k.$$

Therefore, it is known that any of accumulation point x^* of the infinite solution sequence $\{x^k\}$ is a global optimum solution of the problem (MLFP), and the proof is completed. $\square$

4 Computational Complexity Analysis

In this section, by using the maximum edge infinite subdivision of the image space rectangle of the linear fractional functions $\frac{\sum_{j=1}^{n} d_{ij}x_j+g_i}{\sum_{j=1}^{n} e_{ij}x_j+h_i}, i=1,\ldots,p$, and combined with the region deleting criterion in the steps of the branch-and-bound algorithm, the computational complexity of the image space branch-and-bound algorithm proposed in this paper is analyzed for the first time, and the maximum number of iterations of the algorithm is deduced in detail.

First of all, define the size $\zeta(\Omega)$ of the rectangle $\Omega=\{y\in R^p \mid L_i\le y_i\le U_i,\ i=1,\ldots,p\}$ as

$$\zeta(\Omega) := \max\{U_i-L_i, i=1,2,\ldots,p\}.$$

Lemma 4.1. *For arbitrarily given convergence tolerance $\epsilon>0$, if a rectangle Ω^k is generated after k iterations, which satisfies that $\zeta(\Omega^k)\le\epsilon$, then we get that*

$$UB-LB(\Omega^k)\le\epsilon,$$

where $LB(\Omega^k)$ is the optimal value of the problem (LRP) over the rectangle Ω^k, and UB is the currently best upper bound of the optimal value for the problem (EP1).

Proof. Without loss of generality, suppose that $(x^k,\hat{y}^k,\hat{r}^k)$ is the optimal solution of the problem (LRP) over the rectangle Ω^k, let $y_i^k=\frac{\sum_{j=1}^{n} d_{ij}x_j^k+g_i}{\sum_{j=1}^{n} e_{ij}x_j^k+h_i}, i=1,2,\ldots,p$, and let $r^k=\max\{y_1^k,y_2^k,\ldots,y_p^k\}$, then (x^k,y^k,r^k) is a feasible solution to the problem (EP1) over the rectangle Ω^k.

From the definitions of UB_k and $LB(\Omega^k)$, we have

$$r(x^k,y^k)\ge UB_k\ge LB(\Omega^k)=r(x^k,\hat{y}^k).$$

Thus, by steps of the proposed image space branch-and-bound algorithm, we can follow that

$$UB_k-LB(\Omega^k)\le r(x^k,y^k)-r(x^k,\hat{y}^k)=r^k-\hat{r}^k\le\max\{U_i^k-L_i^k, i=1,2,\ldots,p\}=\zeta(\Omega^k).$$

Furthermore, from the above inequalities and $\zeta(\Omega^k)\le\epsilon$, we can get that

$$UB_k-LB(\Omega^k)\le\zeta(\Omega^k)\le\varepsilon,$$

and the proof of the theorem is completed. □

From step 4 of the proposed algorithm and Lemma 4.1, it is easy to know that Ω^k will be deleted if $\zeta(\Omega^k)\le\epsilon$. Therefore, the algorithm stops if the sizes of all rectangles Ω^k generated by the algorithm satisfy $\zeta(\Omega^k)\le\epsilon$. We can use Lemma 4.1 to estimate the maximum number of iterations.

Theorem 4.2. *Given any convergence tolerance $\epsilon>0$, the image space branch-and-bound algorithm finds an ϵ-global optimal solution of the problem (MLFP) in at most*

$$S=2^{\sum_{i=1}^{p}\lceil\log_2\frac{(U_i^0-L_i^0)}{\epsilon}\rceil}-1$$

iterations, where $\Omega^0=\{y\in R^p|L_i^0\le y_i\le U_i^0, i=1,\ldots,p\}$.

Proof. Assume that a rectangle $\Omega = \{y \in R^p | L_i \leq y_i \leq U_i, i = 1, \ldots, p\} \subseteq \Omega^0$ is selected from the set of the partitioned sub-rectangles of Ω^0 for every iteration. Without losing generality, suppose that, after s_i iterations, there exists a subinterval $\Omega_i^{s_i} = [L_i^{s_i}, U_i^{s_i}]$ of the interval $\Omega_i^0 = [L_i^0, U_i^0]$, which satisfies that, for each $i = 1, 2, \ldots, p$,

$$U_i^{s_i} - L_i^{s_i} \leq \epsilon. \tag{4.1}$$

By the bisection method of rectangle, for each $i = 1, 2, \ldots, p$, we have

$$U_i^{s_i} - L_i^{s_i} = \frac{1}{2^{s_i}}(U_i^0 - L_i^0). \tag{4.2}$$

Consequently, by (4.1) and (4.2), for each $i = 1, 2, \ldots, p$, we can get that

$$\frac{1}{2^{s_i}}(U_i^0 - L_i^0) \leq \epsilon,$$

i.e., for each $i = 1, 2, \ldots, p$, we have that

$$s_i \geq \log_2 \frac{(U_i^0 - L_i^0)}{\epsilon}.$$

Next, let

$$\bar{s}_i = \lceil \log_2 \frac{(U_i^0 - L_i^0)}{\epsilon} \rceil, \quad i = 1, 2, \ldots, p.$$

Then, after $S_1 = \sum_{i=1}^{p} \bar{s}_i$ iterations, the proposed image space branch-and-bound algorithm will generate at most $S_1 + 1$ sub-rectangles, which can be denoted as $\Omega^1, \Omega^2, \ldots, \Omega^{S_1+1}$, and which must satisfy

$$\zeta(\Omega^t) = 2^{S_1 - t}\zeta(\Omega^{S_1}) = 2^{S_1 - t}\zeta(\Omega^{S_1+1}),\ t = S_1, S_1 - 1, \ldots, 2, 1,$$

where $\zeta(\Omega^{S_1}) = \zeta(\Omega^{S_1+1}) = \max\{U_i^{\bar{s}_i} - L_i^{\bar{s}_i}, i = 1, 2, \ldots, p\}$ and

$$\Omega^0 = \bigcup_{t=1}^{S_1+1} \Omega^t.$$

Meanwhile, put these $S_1 + 1$ rectangles into the set C^{S_1+1}, i.e.,

$$C^{S_1+1} = \{\Omega^t |\ t = 1, 2, \ldots, S_1 + 1\}.$$

By (4.1), we have

$$\zeta(\Omega^{S_1}) = \zeta(\Omega^{S_1+1}) \leq \epsilon. \tag{4.3}$$

Without losing generality, we denote by $\bar{\zeta} = \zeta(\Omega^{S_1}) = \zeta(\Omega^{S_1+1})$. Hence, from (4.3), we have

$$\bar{\zeta} \leq \epsilon.$$

Thus, by (4.3), Lemma 4.1 and Step 4 of the algorithm, the sub-rectangles Ω^{S_1} and Ω^{S_1+1} have been examined clearly and should be discarded from the partitioning set C^{S_1+1}. Now, the remaining sub-rectangles are put into the set C^{S_1}, where

$$C^{S_1} = C^{S_1+1} \setminus \{\Omega^{S_1}, \Omega^{S_1+1}\} = \{\Omega^t | t = 1, 2, \ldots, S_1 - 1\},$$

and the remaining sub-rectangles Ω^t $(t = 1, 2, \ldots, S_1 - 1)$ will be examined further.

Next, considering the sub-rectangle Ω^{S_1-1}, by the rectangular bisection method, we can subdivide the sub-rectangle Ω^{S_1-1} into two sub-rectangles $\Omega^{S_1-1,1}$ and $\Omega^{S_1-1,2}$, which satisfy that

$$\Omega^{S_1-1} = \Omega^{S_1-1,1} \bigcup \Omega^{S_1-1,2}$$

and

$$\zeta(\Omega^{S_1-1}) = 2\zeta(\Omega^{S_1-1,1}) = 2\zeta(\Omega^{S_1-1,2}) = 2\bar{\zeta}.$$

Thus, after the presented algorithm performed $S_1 + (2^1 - 1)$ iterations, Ω^{S_1-1} has been examined clearly and should be discarded from the partitioning set C^{S_1} by using (4.3), Lemma 4.1 and Step 4 of the algorithm. At the same time, the remaining sub-rectangles will be moved into the set C^{S_1-1}, where

$$C^{S_1-1} = C^{S_1+1} \setminus \{\Omega^{S_1-1}, \Omega^{S_1}, \Omega^{S_1+1}\} = \{\Omega^t |\ t = 1, 2, \ldots, S_1 - 2\}.$$

Similarly, after the proposed algorithm executed $S_1+(2^1-1)+(2^2-1)$ iterations, the sub-rectangle Ω^{S_1-2} has been examined clearly and should be discarded from the partitioning set C^{S_1-1}. At the same time, all remaining sub-rectangles will be put into the set C^{S_1-2}, where

$$C^{S_1-2} = C^{S_1+1} \setminus \{\Omega^{S_1-2}, \Omega^{S_1-1}, \Omega^{S_1}, \Omega^{S_1+1}\} = \{\Omega^t |\ t = 1, 2, \ldots, S_1 - 3\}.$$

Repeat the above iteration process, until all sub-rectangles $\Omega^t(t = 1, 2, \ldots, S_1 + 1)$ are completely removed from Ω^0. Thus, after at most

$$S = S_1 + (2^1 - 1) + (2^2 - 1) + (2^3 - 1) + \cdots + (2^{S_1-1} - 1) = 2^{S_1} - 1 = 2^{\sum\limits_{i=1}^{p} \lceil \log_2 \frac{(U_i^0 - L_i^0)}{\epsilon} \rceil} - 1$$

iterations, the presented algorithm will stop, and the proof is completed. □

Remark 4.3. From Theorem 4.2, we can know that, the running time of the image space branch-and-bound algorithm is bounded by

$$(2S + 1)T(m + 2p, n + p)$$

for finding an ϵ-global optimal solution of the problem (MLFP), where $T(m + 2p, n + p)$ denotes by the time taken to solve a linear programming problem with $n + p$ variables and $m + 2p$ linear constraints.

5 Numerical Experiment

In this section, we numerically compare our algorithm with the software BARON [22] and the existing branch-and-bound-algorithms presented in Feng et al. [16], Jiao & Liu [21], and Wang et al. [35]. All used algorithms are coded in MATLAB R2014a, all test problems are solved on the same microcomputer with Intel(R) Core(TM) i5-7200U CPU @2.50GHz processor and 16 GB RAM. We set the maximum time limit for all algorithms to 3600 seconds. All test problems and their numerical results are listed as follows.

Problem 1.

$$\begin{cases} \min \ \max \left\{ \dfrac{\sum\limits_{j=1}^{n} d_{1j}x_j + g_1}{\sum\limits_{j=1}^{n} e_{1j}x_j + h_1}, \dfrac{\sum\limits_{j=1}^{n} d_{2j}x_j + g_2}{\sum\limits_{j=1}^{n} e_{2j}x_j + h_2}, \ldots, \dfrac{\sum\limits_{j=1}^{n} d_{pj}x_j + g_p}{\sum\limits_{j=1}^{n} e_{pj}x_j + h_p} \right\} \\ \text{s. t.} \ \ \sum\limits_{j=1}^{n} a_{kj}x_j \le b_k, \ k = 1, 2, \ldots, m, \\ \qquad x_j \ge 0, \ j = 1, 2, \ldots, n, \end{cases}$$

where $d_{ij}, e_{ij}, b_k, a_{kj}, i = 1, 2, \ldots, p, k = 1, 2, \ldots, m, j = 1, 2, \ldots, n$, are all randomly generated in the interval $[0, 10]$; g_i and $h_i, i = 1, 2, \ldots, p$, are all randomly generated in the unit interval $[0, 1]$. What needs to be clearly pointed out is that, test Problem 1 has the little constant numbers g_i and h_i at the numerators and denominators of linear fractional functions.

First of all, for test Problem 1 with small-size variables, with the given approximation error $\epsilon = 10^{-4}$, numerical comparisons among the algorithms in Feng et al. [16], Jiao & Liu [21], Wang et al. [35], and our algorithm are listed in Table 1, respectively. Next, for test Problem 1 with large-size variables, with the given approximation error $\epsilon = 10^{-4}$, numerical comparisons between our algorithm and BARON are listed in Table 2, respectively. For all numerical tests, we solved arbitrary ten independently generated test examples and recorded their best, worst, and average results among these ten test examples, and we highlighted in bold the winners of average results in their numerical comparisons. What needs to be pointed out here is that "$-$" represents that the selected algorithm failed to terminate in $3600s$.

From the numerical results for test Problem 1 with small-size variables in Table 1, first of all, we can observe that, when $p \ge 2, m \ge 10$, and $n \ge 4$, the algorithm of Feng et al. [16] failed to terminate in $3600s$ for some of arbitrary ten independently generated test examples; when $p \ge 2, m \ge 10$, and $n \ge 10$, the algorithm of Wang et al. [35] failed to terminate in $3600s$ for some of arbitrary ten independently generated test examples; when $p \ge 4, m \ge 10$, and $n \ge 10$, the algorithm of Jiao & Liu [21] failed to terminate in $3600s$ for some of arbitrary ten independently generated test examples; but in all cases, our algorithm can globally solve all arbitrary ten independently generated test examples. Secondly, in terms of computational performance, when $p \ge 2, m \ge 10$, and $n \ge 4$, in all cases, the algorithms presented in Feng et al. [16], Jiao & Liu [21] and Wang et al. [35] are more time-consuming than our algorithm, so that our algorithm outperforms the algorithms presented in Feng et al. [16], Jiao & Liu [21], and Wang et al. [35].

From the numerical results for test Problem 1 with large-size variables in Table 2, first of all, we can observe that the software BARON is more time-consuming than our algorithm, though the number of iterations for the software BARON is smaller than our algorithm. Secondly, in terms of computational performance of the algorithm, our algorithm outperforms the software BARON. Especially, when we fixed $m = 100$, let $p = 2$ and $n \ge 8000$, or let $p = 3$ and $n \ge 7000$, the software BARON failed to terminate in $3600s$ for some of arbitrary ten independently generated test examples, but in all cases, our algorithm can globally solve all arbitrary ten independently generated test examples.

In general, from the numerical results in Tables 1 and 2, we can see that, when the scale of the test Problem 1 is larger, the computational performance of our algorithm is obviously better than the software BARON and the algorithms presented in Feng et al. [16], Jiao & Liu [21], and Wang et al. [35]. From the numerical results, we can conclude that our algorithm can globally solve the large-size test Problem 1 in a reasonable time.

Table 1: Numerical comparisons between some algorithms and our algorithm on Problem 1.

(p,m,n)	algorithms	#iter			time(s)		
		min.	ave.	max.	min.	ave.	max.
(2,10,2)	Feng et al. [16]	392	3496.6	17914	11.42	130.35	723.47
	Wang et al. [35]	29	58.5	109	0.78	1.78	3.22
	Jiao & Liu [21]	**27**	**47.4**	**75**	**0.81**	**1.45**	**2.25**
	Our algorithm	11	150.6	659	0.18	3.88	18.62
(2,10,4)	Feng et al. [16]	–	–	–	–	–	–
	Wang et al. [35]	51	745.3	2492	2.24	23.00	74.35
	Jiao & Liu [21]	60	300.3	835	2.10	9.06	24.50
	Our algorithm	**15**	**101**	**294**	**0.36**	**2.95**	**8.89**
(2,10,6)	Feng et al. [16]	–	–	–	–	–	–
	Wang et al. [35]	142	4677.8	20401	5.09	165.89	754.32
	Jiao & Liu [21]	167	1507.6	8543	5.29	51.06	296.86
	Our algorithm	**40**	**80.7**	**161**	**1.07**	**2.54**	**5.34**
(2,10,8)	Feng et al. [16]	–	–	–	–	–	–
	Wang et al. [35]	1489	15624.2	57619	399.81	532.63	**2042.29**
	Jiao & Liu [21]	420	1842.7	6978	13.68	62.72	242.83
	Our algorithm	**56**	**221**	**936**	**2.10**	**6.50**	**25.61**
(2,10,10)	Feng et al. [16]	–	–	–	–	–	–
	Wang et al. [35]	–	–	–	–	–	–
	Jiao & Liu [21]	0	3952.1	18813	0	128.66	620.33
	Our algorithm	**49**	**91.9**	**164**	**1.78**	**2.85**	**4.10**
(3,10,10)	Feng et al. [16]	–	–	–	–	–	–
	Wang et al. [35]	–	–	–	–	–	–
	Jiao & Liu [21]	2534	14271.3	44772	84.24	508.25	1712.22
	Our algorithm	**183**	**555**	**2129**	**6.80**	**20.27**	**73.72**
(4,10,10)	Feng et al. [16]	–	–	–	–	–	–
	Wang et al. [35]	–	–	–	–	–	–
	Jiao & Liu [21]	–	–	–	–	–	–
	Our algorithm	**671**	**7489.9**	**26210**	**25.81**	**315.32**	**1183.59**
(5,10,10)	Feng et al. [16]	–	–	–	–	–	–
	Wang et al. [35]	–	–	–	–	–	–
	Jiao & Liu [21]	–	–	–	–	–	–
	Our algorithm	**2122**	**9692.4**	**33056**	**86.57**	**409.15**	**1288.55**

6 Conclusion

This paper studies the minimax linear fractional programming problem and presents an image space branch-and-bound algorithm. In this algorithm, we proposed a novel linearizing technique for constructing the linear relaxation programming problem of the equivalent problem. In contrast to the existing branch-and-bound algorithms, the branching search takes place in the image space rectangle of the linear fractional functions $\frac{\sum_{j=1}^{n} d_{ij}x_j+g_i}{\sum_{j=1}^{n} e_{ij}x_j+h_i}, i=1,\ldots,p$, which mitigates the required computational efforts and the computational complexity of the algorithm. Our algorithm can find a global ε-approximate optimal solution in at most $S=2^{\sum_{i=1}^{p}\lceil\log_2\frac{(U_i^0-L_i^0)}{\epsilon}\rceil}-1$ iterations. Numerical results demonstrate the superiority and effi-

Table 2: Numerical comparisons between BARON and our algorithm on Problem 1.

(p, m, n)	algorithms	#iter			time(s)		
		min.	ave.	max.	min.	ave.	max.
(2,100,1000)	BARON	1	1	1	114.19	161.83	198.98
	Our algorithm	**41**	**56.2**	**79**	**44.65**	**73.62**	**107.21**
(2,100,2000)	BARON	1	1	1	71.19	867.20	2026.48
	Our algorithm	**41**	**69.7**	**182**	**88.42**	**182.54**	**525.55**
(2,100,3000)	BARON	1	1	1	214.97	2279.31	3162.7
	Our algorithm	**42**	**61.6**	**97**	**123.10**	**249.89**	**343.63**
(2,100,4000)	BARON	1	1	1	423.69	521.29	704.05
	Our algorithm	**39**	**45**	**112**	**235.22**	**201.06**	**614.96**
(2,100,5000)	BARON	1	1	1	676.52	1400.99	5305.7
	Our algorithm	**38**	**57.2**	**73**	**307.30**	**409.34**	**599.31**
(2,100,6000)	BARON	1	1	1	1310.03	1614.77	2130.42
	Our algorithm	**41**	**54.3**	**71**	**322.08**	**461.39**	**635.40**
(2,100,7000)	BARON	1	1	1	1714.22	2464.83	3218.08
	Our algorithm	**47**	**65.1**	**82**	**550.85**	**709.27**	**981.18**
(2,100,8000)	BARON	1	1	1	–	–	–
	Our algorithm	**46**	**75.6**	**204**	**586.17**	**1036.01**	**3171.08**
(3,100,1000)	BARON	1	1	1	17.86	207.69	310.17
	Our algorithm	**103**	**203.9**	**493**	**146.04**	**312.89**	**744.01**
(3,100,2000)	BARON	1	1	1	79.53	844.95	2121.8
	Our algorithm	**98**	**204.8**	**426**	**299.41**	**718.02**	**1676.95**
(3,100,3000)	BARON	1	1	1	312.98	1245.00	3203.69
	Our algorithm	**118**	**219.9**	**398**	**603.52**	**1136.81**	**2220.43**
(3,100,4000)	BARON	1	1	1	549.14	656.52	796.19
	Our algorithm	**106**	**199.4**	**447**	**695.33**	**1402.37**	**3255.30**
(3,100,5000)	BARON	1	1	1	1022.33	1233.59	1402.52
	Our algorithm	**83**	**169.5**	**268**	**619.62**	**1414.56**	**2253.12**
(3,100,6000)	BARON	1	1	1	1667.3	2160.854	2716.88
	Our algorithm	**103**	**193.9**	**318**	**984.5**	**1894.83**	**3148.57**
(3,100,7000)	BARON	–	–	–	–	–	–
	Our algorithm	**121**	**153**	**200**	**1381.38**	**1789.36**	**2343.83**
(3,100,8000)	BARON	–	–	–	–	–	–
	Our algorithm	**115**	**170.1**	**249**	**1740.10**	**2330.88**	**3045.42**

ciency of our algorithm. In future work, by using robust dual approach in [33], we will extend and apply the proposed algorithm to solve the uncertain minimax fractional programming introduced in [23] .

Acknowledgements

All authors are very grateful to the responsible editors and the anonymous reviewers for their valuable comments and suggestions, which have greatly improved the earlier version of our paper.

References

[1] S. Addoune, M. E. Haffari and A. Roubi, A proximal point algorithm for generalized fractional programs, *Optimization* 66 (2017) 1495–1517.

[2] I. Barrodale, Best rational approximation and strict quasiconvexity, *SIAM J. Numer. Anal.* 10 (1973) 8–12.

[3] C. Bajona-Xandri and J.E. Martinez-Legaz, Lower subdifferentiability in minimax fractional programming, *Optimization* 45 (1999) 1–12.

[4] A.I. Barros and J.B.G. Frenk, Generalized fractional programming and cutting plane algorithms, *J. Optim. Theory Appl.* 87 (1995) 103–120.

[5] H. Boualam and A. Roubi, Dual algorithms based on the proximal bundle method for solving convex minimax fractional programs, *J. Ind. Manag. Optim.* 15(4) (2019) 1897–1920.

[6] K. Boufi and A. Roubi, Dual method of centers for solving generalized fractional programs, *J. Glob. Optim.* 69 (2017) 387–426.

[7] K. Boufi, M. E. Haffari and A. Roubi, Optimality conditions and a method of centers for minimax fractional programs with difference of convex functions, *J. Optim. Theory Appl.* 187 (2020) 105–132.

[8] K. Boufi and A. Roubi, Duality results and dual bundle methods based on the dual method of centers for minimax fractional programs, *SIAM J. Optimiz.* 29 (2019) 1578–1602.

[9] J.P. Crouzeix, J.A. Ferland and S. Schaible, An algorithm for generalized fractional programs, *J. Optim. Theory Appl.* 47 (1985) 135–149.

[10] H.J. Chen, S. Schaible and R. L. Sheu, Generic algorithm for generalized fractional programming, *J. Optim. Theory Appl.* 141 (2009) 93–105.

[11] F. Ding, Coupled-least-squares identification for multivariable systems, *IET Control Theory Appl.* 7 (2013) 68–79.

[12] F. Ding, X. Liu and J. Chu, Gradient-based and least-squares-based iterative algorithms for Hammerstein systems using the hierarchical identification principle, *IET Control Theory Appl.* 7 (2013) 176–184.

[13] F. Ding, Decomposition based fast least squares algorithm for output error systems, *Signal Process.* 93 (2013) 1235–1242.

[14] F. Ding, Two-stage least squares based iterative estimation algorithm for CARARMA system modeling, *Appl. Math. Model.* 37 (2013) 4798–4808.

[15] R.W. Freund and F. Jarre, An interior-point method for fractional programs with convex constraints, *Math. Program.* 67 (1994) 407–440.

[16] Q. Feng, H. Mao and H. Jiao, A feasible method for a class of mathematical problems in manufacturing system, *Key Eng. Mater.* 460-461 (2011) 806–809.

[17] Q. Feng, H. Jiao and H. Mao, A deterministic algorithm for min-max and max-min linear fractional programming problems, *Int. J. Comput. Int. Sys.* 4 (2011) 134–141.

[18] A. Ghazi and A. Roubi, A DC approach for minimax fractional optimization programs with ratios of convex functions, *Optim. Methods Softw.* (2020) DOI:10.1080/10556788.2020.1818234

[19] Y. Gao and W.D. Rong, Optimality conditions and duality for a class of nondifferentiable multi-objective generalized fractional programming problems, *Appl. Math. Ser. B* 23 (2008) 331–344.

[20] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches.* 2nd Edition, Spring Verlag, Berlin, 1993.

[21] H.W. Jiao and S.Y. Liu, A new linearization technique for minimax linear fractional programming, *Int. J. Comput. Math.* 91 (2014) 1730–1743.

[22] A. Khajavirad and N.V. Sahinidis, A hybrid LP/NLP paradigm for global optimization relaxations, *Math. Program. Comput.* 10 (2018) 383–421.

[23] X.B. Li, Q.L. Wang and Z. Lin, Optimality conditions and duality for minimax fractional programming problems with data uncertainty, *J. Ind. Manag. Optim.* 15 (2019) 1133–1151.

[24] H.C. Lai and T.Y. Huang, Nondifferentiable minimax fractional programming in complex spaces with parametric duality, *J. Global Optim.* 53 (2012) 243–254.

[25] H.C. Lai, J.C. Liu and S. Schaible, Complex minimax fractional programming of analytic functions, *J. Optim. Theory Appl.* 137 (2008) 171–184.

[26] H.C. Lai, J.C. Liu and K. Tanaka, Necessary and sufficient conditions for minimax fractional programming, *J. Math. Anal. Appl.* 230 (1999) 311–328.

[27] N.T.H. Phuong and H. Tuy, A unified monotonic approach to generalized linear fractional programming, *J. Global Optim.* 26 (2003) 229–259.

[28] A. Roubi and M. E. Haffari, Prox-dual regularization algorithm for generalized fractional programs, *J. Ind. Manag. Optim.* 13 (2017) 1991–2013.

[29] A. Smail, B. Karima and R. Ahmed, Proximal Bundle Algorithms for nonlinearly constrained convex minimax fractional programs, *J. Optim. Theory Appl.* 179 (2018) 212–239.

[30] S. Schaible and J. Shi, Fractional programming: The sum-of-ratios case, *Optim. Method Softw.* 18 (2003) 219–229.

[31] I.M. Stancu-Minasian, A eighth bibliography of fractional programming, *Optimization* 66 (2017) 439–470.

[32] I.M. Stancu-Minasian, A ninth bibliography of fractional programming, *Optimization* 68 (2019) 2123–2167.

[33] X. Sun, K.L. Teo and L. Tang, Dual approaches to characterize robust optimal solution sets for a class of uncertain optimization problems, *J. Optim. Theory Appl.* 182(3) (2019) 984–1000.

[34] W. Wang, J.H. Li and R.F. Ding, Maximum likelihood parameter estimation algorithm for controlled autoregressive models, *Int. J. Comput. Math.* 88 (2011) 3458–3467.

[35] C.-F. Wang, Y. Jiang and P.-P. Shen, A new branch-and-bound algorithm for solving minimax linear fractional programming, *J. of Math. (PRC)* 38 (2018) 113–123.

Manuscript received 28 June 2021
revised 6 September 2021
accepted for publication 20 October 2021

Hongwei Jiao
School of Mathematical Sciences
Henan Institute of Science and Technology
Xinxiang 453003, China
E-mail address: jiaohongwei@126.com

Junqiao Ma
School of Mathematical Sciences
Henan Institute of Science and Technology
Xinxiang 453003, China
E-mail address: majmur@163.com

Youlin Shang
School of Mathematics and Statistics
Henan University of Science and Technology
Luoyang 471023, China
E-mail address: mathshang@sina.com