Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 3584494, 10 pages
https://doi.org/10.1155/2021/3584494

Research Article

An Efficient Branch-and-Bound Algorithm for Globally Solving Minimax Linear Fractional Programming Problem

Pujun Jia, Hongwei Jiao, Dongwei Shi, and Jingben Yin

School of Mathematical Sciences, Henan Institute of Science and Technology, Xinxiang 453003, China

Correspondence should be addressed to Hongwei Jiao; jiaohongwei@126.com

Received 4 August 2021; Revised 26 October 2021; Accepted 10 November 2021; Published 27 November 2021

Academic Editor: Konstantina Skouri

This paper presents an efficient outer space branch-and-bound algorithm for globally solving a minimax linear fractional programming problem (MLFP), which has a wide range of applications in data envelopment analysis, engineering optimization, management optimization, and so on. In this algorithm, by introducing auxiliary variables, we first equivalently transform the problem (MLFP) into the problem (EP). By using a new linear relaxation technique, the problem (EP) is reduced to a sequence of linear relaxation problems over the outer space rectangle, which provides the valid lower bound for the optimal value of the problem (EP). Based on the outer space branch-and-bound search and the linear relaxation problem, an outer space branch-and-bound algorithm is constructed for globally solving the problem (MLFP). In addition, the convergence and complexity of the presented algorithm are given. Finally, numerical experimental results demonstrate the feasibility and efficiency of the proposed algorithm.

1. Introduction

This paper considers the following minimax linear fractional programming problem:

$$
(\text{MLFP}): \begin{cases} \min \quad \max\left\{\dfrac{\varphi_1(x)}{\psi_1(x)}, \dfrac{\varphi_2(x)}{\psi_2(x)}, \ldots, \dfrac{\varphi_p(x)}{\psi_p(x)}\right\} \\ \\ \text{s.t.} \quad x \in D = \{x \in R^n | Ax \leq b\}, \end{cases} \tag{1}
$$

where $p \geq 2$, $A \in R^{m\times n}$, $b \in R^m$, D is a nonempty bounded polyhedral set, and $\varphi_i(x) = \sum_{j=1}^n c_{ij}x_j + d_i$ and $\psi_i(x) = \sum_{j=1}^n e_{ij}x_j + f_i$ are all bounded affine functions over D, and for $\forall x \in D$, we have that $\varphi_i(x) \geq 0$ and $\psi_i(x) > 0, i = 1, \ldots, p$.

The problem (MLFP) has aroused the interest of many practitioners and researchers. On the one hand, it has a wide range of applications, such as electronic circuit design [1], system identification [2–4], and finance and investment [5]. On the other hand, because it has multiple local optimal solutions that are not globally optimal, it is still challenging to solve the problem (MLFP). So, it is necessary to come up with an efficient algorithm to globally solve the problem (MLFP).

In the past few decades, many algorithms have been proposed for solving some special forms of the problem (MLFP). In general, these algorithms can be classified into the following categories: interior-point algorithm [6], parameter programming method [7], partial linearization algorithm [8], monotonic optimization method [9], cutting plane algorithm [10], and branch-and-bound algorithm [11–19]. In addition to the above algorithms, in recent years, Chen et al. [20] proposed a unified framework to study various versions of Dinkelbach-type algorithms for solving the generalized fractional programming problem; based on a proximal bundle method, Boualam and Roubi [21] designed a dual algorithm for solving the convex minimax fractional programming problem; by analyzing the dual method of centers, Boufi and Roubi [22] presented an algorithm for solving the generalized fractional programs; Lai and Huang [23] presented a duality programming problem for solving a complex nondifferentiable minimax fractional programming with complex variables. However, the above algorithms can only deal with some specific cases of the problem (MLFP). Therefore, it is necessary to propose an efficient algorithm for globally solving the general form of the problem (MLFP).

In this paper, we present an efficient outer space branch-and-bound algorithm for globally solving the minimax linear fractional programming problem (MLFP). In this algorithm, we first transform the problem (MLFP) into an outer space equivalent problem (EP) by introducing auxiliary variables. Then, in order to obtain the lower bound of the optimal value of the problem (EP), a new linear relaxation technique is proposed to construct the linear relaxation problem (LRP) of the problem (EP) over the outer space rectangle. And we can solve the problem (EP) by solving a sequence of linear relaxation problems over the outer space rectangle. Based on the outer space branch-and-bound search and the new linear relaxation problem over the outer space rectangle, we present an outer space branch-and-bound algorithm for globally solving the problem (MLFP). In addition, the global convergence and the computational complexity of the proposed algorithm are given. The numerical results demonstrate that the proposed algorithm can effectively find the global optimal solutions for all tested instances. The numerical comparisons among our algorithm and the algorithms of Feng et al. [8], Jiao and Liu [12], and Wang et al. [24] indicate that our algorithm outperforms the algorithms presented in Feng et al. [8], Jiao and Liu [12], and Wang et al. [24].

This paper is organized as follows. In Section 2, the equivalent problem (EP) of the problem (MFLP) is obtained, and by utilizing the new linear relaxation technique, the linear relaxation problem (LRP) of the problem (EP) over the outer space rectangle is constructed. In Section 3, an efficient outer space branch-and-bound algorithm is described, and the global convergence and the computational complexity of the proposed algorithm are given. Numerical results are reported to show the feasibility and efficiency of the proposed algorithm in Section 4. Finally, some conclusions are presented in Section 5.

2. Equivalent Problem and Its Linear Relaxation

For globally solving the problem (MLFP), we need to establish its equivalent problem (EP). For this purpose, we first compute the initial lower bound $\underline{\beta}_i^0 = \min_{x\in D}\psi_i(x)$ and the upper bound $\overline{\beta}_i^0 = \max_{x\in D}\psi_i(x)$ of the function $\psi_i(x)$ over D so that we can obtain the initial outer space rectangle:

$$\Omega^0 = \left\{\psi_i(x) \in R^p \,|\, \underline{\beta}_i^0 \le \psi_i(x) \le \overline{\beta}_i^0, i = 1, 2, \dots, p\right\}. \tag{2}$$

By introducing a new variable r, we can establish the outer space equivalent problem (EP) of the problem (MLFP) as follows:

$$(\text{EP}): \begin{cases} \min & r \\ & \dfrac{\varphi_i(x)}{\psi_i(x)} \le r, i = 1, 2, \dots, p, \\ \text{s.t.} & \\ & \underline{\beta}_i^0 \le \psi_i(x) \le \overline{\beta}_i^0,\ i = 1, 2, \dots, p, Ax \le b. \end{cases} \tag{3}$$

Theorem 1. *(x^*, r^*) is a global optimal solution of the problem (EP) if and only if x^* is a global optimal solution of the problem (MLFP).*

Proof. The conclusion of the theorem is obvious; therefore, the proof is omitted.

From Theorem 1, for globally solving the problem (MLFP), we can globally solve its equivalence problem (EP) instead; they have the same optimal solutions and optimal value. In order to globally solve the problem (EP), we need to construct its linear relaxation problem of the problem (EP) over the outer space rectangle, which can provide a reliable lower bound for the optimal value of the problem (EP). And the next main task is to construct the linear relaxation problem of the problem (EP).

First of all, let $\Omega = \left\{\psi_i(x) \in R^p | \underline{\beta}_i \le \psi_i(x) \le \overline{\beta}_i, i = 1, 2, \dots, p\right\}$ denote the initial rectangle Ω^0 or a subrectangle of Ω^0 that is generated by the branching operation. Then, for convenience in expression, some notations are introduced as follows:

$$\underline{\alpha}_i^0 = \min_{x\in D} \varphi_i(x), \overline{\alpha}_i^0 = \max_{x\in D} \varphi_i(x). \tag{4}$$

From the above formulas, it is easy to see that $0 \le \underline{\alpha}_i^0 \le \varphi_i(x) \le \overline{\alpha}_i^0, 0 < \underline{\beta}_i \le \psi_i(x) \le \overline{\beta}_i, i = 1, 2, \dots, p$. And we consider the term $(\varphi_i(x)/\psi_i(x)), i = 1, 2, \dots, p$, and it follows that $\overline{\alpha}_i^0\psi_i(x) - \underline{\beta}_i\varphi_i(x) \ge 0,\ \psi_i(x) - \underline{\beta}_i \ge 0$, so that we have

$$\left(\overline{\alpha}_i^0\psi_i(x) - \underline{\beta}_i\varphi_i(x)\right)\left(\psi_i(x) - \underline{\beta}_i\right) \ge 0. \tag{5}$$

Expanding the above formula, we have that

$$\overline{\alpha}_i^0\psi_i^2(x) - \overline{\alpha}_i^0\underline{\beta}_i\psi_i(x) - \underline{\beta}_i\varphi_i(x)\psi_i(x) + \underline{\beta}_i^2\varphi_i(x) \ge 0. \tag{6}$$

Since $\underline{\beta}_i^2\psi_i(x) > 0$, we divide both ends of the above inequality by $\underline{\beta}_i^2\psi_i(x)$ at the same time, and we can obtain that

$$\frac{\varphi_i(x)}{\psi_i(x)} \ge \frac{\varphi_i(x)}{\underline{\beta}_i} - \frac{\overline{\alpha}_i^0}{\underline{\beta}_i^2}\psi_i(x) + \frac{\overline{\alpha}_i^0}{\underline{\beta}_i}. \tag{7}$$

Let $\Psi_i(x, r) = (\varphi_i(x)/\psi_i(x)) - r$; we have the following relation:

$$\Psi_i(x, r) = \frac{\varphi_i(x)}{\psi_i(x)} - r \ge \frac{\varphi_i(x)}{\underline{\beta}_i} - \frac{\overline{\alpha}_i^0}{\underline{\beta}_i^2}\psi_i(x) + \frac{\overline{\alpha}_i^0}{\underline{\beta}_i} - r = \Psi_i^l(x, r). \tag{8}$$

Through the above discussions, the linear relaxation problem of the problem (EP) over the outer space rectangle Ω is constructed as follows:

$$(\text{LRP}): \begin{cases} \min & r \\ & \Psi_i^l(x, r) \le 0, i = 1, 2, \dots, p, \\ \text{s.t.} & \underline{\beta}_i \le \psi_i(x) \le \overline{\beta}_i, i = 1, 2, \dots, p, \\ & Ax \le b. \end{cases} \tag{9}$$

From the above results, the optimal value of the problem (LRP) is less than or equal to that of the problem (EP) over

Ω. Therefore, the problem (LRP) can provide a valid lower bound for the optimal value of the problem (EP) over Ω. Theorem 2 will show that the problem (LRP) will infinitely approximate the problem (EP) over Ω as $\|\overline{\beta}-\underline{\beta}\| \longrightarrow 0$. □

Theorem 2. *For any $\psi_i(x) \in \Omega = [\underline{\beta}, \overline{\beta}]$, we consider the functions $\Psi_i^l(x,r)$ and $\Psi_i(x,r)$. Then, we have that*

$$\lim_{\|\overline{\beta}-\underline{\beta}\| \longrightarrow 0} \left(\Psi_i(x,r) - \Psi_i^l(x,r)\right) \longrightarrow 0. \tag{10}$$

Proof. From the definitions $\Psi_i(x,r)$ and $\Psi_i^l(x,r)$, we have

$$\begin{aligned} \left|\Psi_i(x,r) - \Psi_i^l(x,r)\right| &= \left|\frac{\varphi_i(x)}{\psi_i(x)} - \left(\frac{\varphi_i(x)}{\underline{\beta}_i} - \frac{\overline{\alpha}_i^0}{\underline{\beta}_i^2}\psi_i(x) + \frac{\overline{\alpha}_i^0}{\underline{\beta}_i}\right)\right| \\ &= \left|\frac{\varphi_i(x)}{\psi_i(x)} - \frac{\varphi_i(x)}{\underline{\beta}_i} + \frac{\overline{\alpha}_i^0}{\underline{\beta}_i^2}\psi_i(x) - \frac{\overline{\alpha}_i^0}{\underline{\beta}_i}\right| \le \left|\varphi_i(x)\left(\frac{1}{\psi_i(x)} - \frac{1}{\underline{\beta}_i}\right)\right| + \left|\frac{\overline{\alpha}_i^0}{\underline{\beta}_i}\left(\frac{\psi_i(x)}{\underline{\beta}_i} - 1\right)\right| \\ &= \left|\varphi_i(x)\frac{\underline{\beta}_i - \psi_i(x)}{\psi_i(x)\underline{\beta}_i}\right| + \left|\frac{\overline{\alpha}_i^0}{\underline{\beta}_i}\,\frac{\psi_i(x) - \underline{\beta}_i}{\underline{\beta}_i}\right| \le \overline{\alpha}_i^0\frac{\overline{\beta}_i - \underline{\beta}_i}{\underline{\beta}_i^2} + \frac{\overline{\alpha}_i^0}{\underline{\beta}_i}\,\frac{\overline{\beta}_i - \underline{\beta}_i}{\underline{\beta}_i} = \frac{2\left(\overline{\alpha}_i^0\right)^2}{\underline{\beta}_i^2}\left(\overline{\beta}_i - \underline{\beta}_i\right). \end{aligned} \tag{11}$$

Therefore, we have that

$$\lim_{\|\overline{\beta}-\underline{\beta}\| \longrightarrow 0} \left(\Psi_i(x,r) - \Psi_i^l(x,r)\right) \longrightarrow 0, \tag{12}$$

and the proof of the theorem is completed.

From Theorem 2, we can know that the linear relaxation function $\Psi_i^l(x,r)$ will infinitely approximate the function $\Psi_i(x,r)$ as $\|\overline{\beta}-\underline{\beta}\| \longrightarrow 0$, which guarantees the global convergence of the proposed algorithm. □

3. Algorithm, Global Convergence, and Complexity Analysis

In this section, we first describe the outer space branching rule. Then, based on the outer space partitioning search and the linear relaxation problem, an efficient outer space branch-and-bound algorithm is developed to globally solve the problem (MLFP). Finally, the global convergence and the computational complexity of the presented algorithm are derived.

3.1. Outer Space Partitioning Rule. The important factor to ensure that the proposed algorithm in this paper can converge to the global minimum of the problem (MLFP) is to choose a suitable outer space partitioning strategy. In this paper, we choose an outer space rectangle bisection method, which can ensure the global convergence of the proposed outer space branch-and-bound algorithm. The proposed outer space partitioning rule is given as follows.

Consider any node subproblem identified by the subrectangle $\Omega = \{\psi_i(x) \in R^p | \underline{\beta}_i \le \psi_i(x) \le \overline{\beta}_i, i = 1, 2, \ldots, p\} \subseteq \Omega^0$; the branching rule is described as follows:

(i) Let $\tau = \arg\max\{\overline{\beta}_i - \underline{\beta}_i | i = 1, 2, \ldots, p\}$.

(ii) Let

$$\begin{aligned} \Omega' &= \left\{\psi_i(x) \in R^p | \underline{\beta}_i \le \psi_i(x) \le \overline{\beta}_i, i = 1, 2, \ldots, p, i \ne \tau, \underline{\beta}_\tau \le \psi_\tau(x) \le \frac{\underline{\beta}_\tau + \overline{\beta}_\tau}{2}\right\}, \\ \Omega'' &= \left\{\psi_i(x) \in R^p | \underline{\beta}_i \le \psi_i(x) \le \overline{\beta}_i, i = 1, 2, \ldots, p, i \ne \tau, \frac{\underline{\beta}_\tau + \overline{\beta}_\tau}{2} \le \psi_\tau(x) \le \overline{\beta}_\tau\right\}. \end{aligned} \tag{13}$$

From the outer space partitioning rule, it can be seen that the outer space rectangle Ω is partitioned into two outer space subrectangles Ω' and Ω''.

3.2. Algorithm Statement. By the above discussions, the basic steps of the proposed algorithm are given as follows. Let $\mathrm{LB}(\Omega^k)$ and $(x(\Omega^k), r(\Omega^k))$ be the optimal objective functional value of the problem (LRP) and an element of the corresponding argmin over the subrectangle Ω^k, respectively.

Step 1: given the convergence tolerance $\varepsilon \ge 0$, the set of feasible points $F = \emptyset$ and the upper bound $\mathrm{UB}_0 = +\infty$.

Solve the problem LRP (Ω^0) for obtaining its optimal solution $(x(\Omega^0), r(\Omega^0))$ and optimal value $\mathrm{LB}(\Omega^0)$. Let $(x^0, r^0) = (x(\Omega^0), r(\Omega^0))$ and $\mathrm{LB}_0 = \mathrm{LB}(\Omega^0)$.

If (x^0, r^0) is feasible to the problem (EP), then we update F and UB_0.

If $\mathrm{UB}_0 - \mathrm{LB}_0 \le \varepsilon$, then the algorithm stops with that (x^0, r^0) is an ε-global optimal solution for the problem (EP); otherwise, set $\Theta_0 = \{\Omega^0\}$ be the set of all active notes, $k = 1$, and go to step 2.

Step 2: let $\mathrm{UB}_k = \mathrm{UB}_{k-1}$, select and subdivide Ω^{k-1} into two subrectangles $\Omega^{k,1}$ and $\Omega^{k,2}$ based on the branching rule. Set $\overline{\Omega} = \{\Omega^{k,1}, \Omega^{k,2}\}$.

Step 3: for each $\Omega^{k,s} \in \overline{\Omega}\,(s = 1, 2)$, solve the problem LRP $(\Omega^{k,s})$ to get the lower bound $\mathrm{LB}(\Omega^{k,s})$ and $(x(\Omega^{k,s}), r(\Omega^{k,s}))$.

If $\mathrm{LB}(\Omega^{k,s}) > \mathrm{UB}_k$, let $\overline{\Omega} = \overline{\Omega} \backslash \Omega^{k,s}$; otherwise, set

$$F = F \cup \{(x(\Omega), r(\Omega))\}. \tag{14}$$

Update the upper bound:

$$\mathrm{UB}_k = \min\left\{\mathrm{UB}_k, r\left(\Omega^{k,s}\right)\right\}. \tag{15}$$

If $\mathrm{UB}_k = r(\Omega^{k,s})$, let $(x^k, r^k) = (x(\Omega^{k,s}), r(\Omega^{k,s}))$, and it is obvious that (x^k, r^k) is the best feasible solution for the problem (EP). Let $\Theta_k = (\Theta_{k-1} \backslash \Omega^{k-1}) \cup \overline{\Omega}$ and $\mathrm{LB}_k = \min\{\mathrm{LB}(\Omega) | \Omega \in \Theta_k\}$.

Step 4: set $\Theta_{k+1} = \Theta_k \backslash \{\Omega\colon \mathrm{UB}_k - \mathrm{LB}(\Omega) > \varepsilon, \Omega \in \Theta_k\}$.

If $\Theta_{k+1} = \varnothing$, then the algorithm stops, and (x^k, r^k) is an ϵ-global optimal solution of the problem (EP).

Otherwise, select the new subrectangle Ω^k such that $\Omega^{k+1} = \operatorname{argmin}_{\Omega \in \Theta_k} \mathrm{LB}(\Omega)$; let $k = k + 1$, and return to step 2.

3.3. Convergence Analysis. The following theorem gives the proof of the convergence of the above algorithm.

Theorem 3. *The proposed algorithm either terminates finitely with the solution x^k which is a global ϵ-optimal solution for the problem (MLFP) or generates an infinite sequence $\{x^k\}$ of iterations such that any infinite branch of the branch-and-bound tree and any accumulation point will be a global optimal solution of the problem (MLFP).*

Proof. When the algorithm is finitely terminated, the conclusion is obvious. When the algorithm is not finitely terminated, Horst and Tuy [25] point out that a sufficient condition for the algorithm to be convergent to the global minimum is that the bounding operation must be consistent and the selection operation is bound improving.

Let LB_k be a lower bound computed in stage k, and let UB_k be the best upper bound at iteration k not necessarily occurring inside the same subrectangle with LB_k. A bounding operation is called consistent if, at every step, any unfathomed partition can be further refined, and since our subdivision process is the bisection, the branching process is exhaustive. Therefore, from Theorem 2, we have that

$$\lim_{k \longrightarrow \infty} \left(\mathrm{UB}_k - \mathrm{LB}_k\right) = 0. \tag{16}$$

Then, this implies that the employed bounding operation is consistent.

A selection operation is called bound improving if at least one partition element where the actual upper bound is attained is selected for further partition after a finite number of refinements. Obviously, since the partition element where the actual upper bound is attained is selected for further partition in the immediately following iteration, the employed selection operation is bound improving.

In general, it can show that the bounding operation is consistent and that selection operation is bound improving. Therefore, the proposed algorithm is convergent to the global minimum of the problem (MFLP), and the proof of the theorem is completed. □

3.4. Computational Complexity Analysis. In this section, we will analyze the complexity of the algorithm. To this end, we define the size $\Delta(\Omega)$ of a rectangle $\Omega = \{\psi_i(x) \in R^p | \underline{\beta}_i \le \psi_i(x) \le \overline{\beta}_i, i = 1, 2, \ldots, p\} \subseteq \Omega^0$, given by

$$\Delta(\Omega) \coloneqq \max\left\{\overline{\beta}_i - \underline{\beta}_i | i = 1, 2, \ldots, p\right\}. \tag{17}$$

Additionally, for convenience, we denote by

$$\mu = \max\left\{\frac{2\left(\overline{\alpha}_i^0\right)^2}{\left(\underline{\beta}_i^0\right)^2}, i = 1, 2, \ldots, p\right\}. \tag{18}$$

Theorem 4. *Given the convergence tolerance $\varepsilon > 0$, if there exists a rectangle Ω^k generated by the algorithm at k_{th} iteration, such that $\Delta(\Omega^k) \le (\varepsilon/\mu)$, then we have*

$$\mathrm{UB} - \mathrm{LB}\left(\Omega^k\right) \le \varepsilon, \tag{19}$$

where $LB(\Omega^k)$ is the optimal value to the problem (LRP (Ω^k)) and UB is the best current known upper bound of the optimum value to the problem (EP).

Proof. From Theorem 2, $\Delta(\Omega^k) \le (\varepsilon/\mu)$, and the definition of $\overline{\alpha}_i^0$, $\underline{\beta}_i$, and $\overline{\beta}_i$, we have

$$\begin{aligned}
\left|\mathrm{UB} - \mathrm{LB}\left(\Omega^k\right)\right| &\le \max\left\{\frac{2\left(\overline{\alpha}_i^0\right)^2}{\underline{\beta}_i^2}\left(\overline{\beta}_i - \underline{\beta}_i\right), i = 1, 2, \ldots, p\right\} \\
&\le \max \le \varepsilon, \left\{\frac{2\left(\overline{\alpha}_i^0\right)^2}{\left(\underline{\beta}_i^0\right)^2}\left(\overline{\beta}_i - \underline{\beta}_i\right), i = 1, 2, \ldots, p\right\} \\
&\le \mu\Delta(\Omega),
\end{aligned} \tag{20}$$

and the proof of the theorem is completed. □

Remark 1. Given the convergence tolerance $\varepsilon > 0$, the proposed algorithm finds a global ε-optimal solution to the problem (MFLP) in at most iterations:

$$N = 2^{\sum_{i=1}^{p} \lceil \log_2 \left(\mu\left(\overline{\beta}_i^0 - \underline{\beta}_i^0\right)/\varepsilon\right) \rceil} - 1. \tag{21}$$

Based on the above discussions, the maximum iterations of the algorithm can be obtained by analyzing the computational complexity of the proposed algorithm.

4. Numerical Experiments

In this section, we numerically compare our algorithm with the existing branch-and-bound-algorithms presented in the works of Feng et al. [8], Jiao and Liu [12], and Wang et al. [24]. All algorithms are coded in MATLAB R2014a; our algorithm is executed on the microcomputer with Intel(R) Core(TM) i7-10700K CPU @3.80 GHz processor and 32 GB RAM. All test examples and their numerical results are listed as follows.

First of all, for Examples 1–8, when n is smaller, with the given approximation error $\varepsilon = 10^{-6}$, numerical comparisons among our algorithm and the algorithms presented in works of Feng et al. [8], Jiao and Liu [12], and Wang et al. [24] are given in Table 1. Next, for randomly generated large-scale Example 9, when $n \geq 1000$, with the given approximation error $\varepsilon = 10^{-4}$, numerical computational results of our algorithm are given in Table 2. When $n \geq 1000$, since the algorithms presented in works of Feng et al. [8], Jiao and Liu [12], and Wang et al. [24] all failed to solve the randomly generated large-scale Example 9 in 3600 s, we only present the numerical computational results of our algorithm in Table 2. For all numerical tests of randomly generated Example 9, we solved arbitrary ten independently generated test problems and recorded their average numerical results among these ten test problems.

From the numerical results for Examples 1–8 in Table 1, first of all, we can observe that our algorithm can obtain the almost same optimal solution and optimal value as the algorithms presented in the works of Feng et al. [8], Jiao and Liu [12], and Wang et al. [24]. Secondly, in terms of computational efficiency, in all cases, our algorithm outperforms the algorithms presented in the works of Feng et al. [8], Jiao and Liu [12], and Wang et al. [24].

From the numerical results for the randomly generated large-scale Example 9 in Table 2, it is obvious that the proposed algorithm can solve randomly generated large-scale Example 9 with the large-size variables ($n \geq 1000$). However, the algorithms of Feng et al. [8], Jiao and Liu [12], and Wang et al. [24] all failed to solve the randomly generated large-scale Example 9. Therefore, this demonstrates the robustness and stability of our algorithm.

Example 1 (see [12, 14]).

$$\begin{cases} \min \quad \max\left\{\dfrac{3x_1 + x_2 - 2x_3 + 0.8}{2x_1 - x_2 + x_3}, \dfrac{4x_1 - 2x_2 + x_3}{7x_1 + 3x_2 - x_3}\right\} \\ \text{s.t.} \quad x_1 + x_2 - x_3 \leq 1, \\ \quad -x_1 + x_2 - x_3 \leq -1, \\ \quad 12x_1 + 5x_2 + 12x_3 \leq 34.8, \\ \quad 12x_1 + 12x_2 + 7x_3 \leq 29.1, \\ \quad -6x_1 + x_2 + x_3 \leq -4.1, \\ \quad 1.0 \leq x_1 \leq 1.1, 0.55 \leq x_2 \leq 0.65, 1.35 \leq x_3 \leq 1.45. \end{cases} \tag{22}$$

Example 2 (see [12, 14]).

$$\begin{cases} \max \quad \min\left\{\dfrac{37x_1 + 73x_2 + 13}{13x_1 + 13x_2 + 13}, \dfrac{63x_1 - 18x_2 + 39}{13x_1 + 26x_2 + 13}\right\} \\ \text{s.t.} \quad 5x_1 - 3x_2 = 3, \\ \quad 1.5 \leq x_1 \leq 3. \end{cases} \tag{23}$$

Example 3 (see [12, 14]).

$$\begin{cases} \min \quad \max\left\{\dfrac{2x_1 + 2x_2 - x_3 + 0.9}{x_1 - x_2 + x_3}, \dfrac{3x_1 - x_2 + x_3}{8x_1 + 4x_2 - x_3}\right\} \\ \text{s.t.} \quad x_1 + x_2 - x_3 \leq 1, \\ \quad -x_1 + x_2 - x_3 \leq -1, \\ \quad 12x_1 + 5x_2 + 12x_3 \leq 34.8, \\ \quad 12x_1 + 12x_2 + 7x_3 \leq 29.1, \\ \quad -6x_1 + x_2 + x_3 \leq -4.1, \\ \quad 1.0 \leq x_1 \leq 1.2, 0.55 \leq x_2 \leq 0.65, 1.35 \leq x_3 \leq 1.45. \end{cases} \tag{24}$$

Table 1: Numerical comparisons among the algorithm presented in the works of Feng et al. [8], Jiao et al. [12], Wang et al. [24], and our algorithm on Examples 1–8.

Example	Refs.	Optimal value	Optimal solution	Iter	CPU time
1	Feng et al. [8]	0.57310	(1.0157, 0.5905, 1.4037)	1897	88.40531
	Jiao and Liu [12]	0.57310	(1.0157, 0.5905, 1.4037)	22	1.05172
	Wang et al. [24]	0.57310	(1.0157, 0.5905, 1.4037)	22	1.38462
	Ours	0.57335	(1.0157, 0.5902, 1.4041)	1	0.20335
2	Feng et al. [8]	0.67136	(1.5000, 1.5000)	123	3.69388
	Jiao and Liu [12]	0.67136	(1.5000, 1.5000)	16	0.52438
	Wang et al. [24]	0.67136	(1.5000, 1.5000)	20	1.11778
	Ours	0.67136	(1.5000, 1.5000)	1	0.05756
3	Feng et al. [8]	1.34783	(1.0167, 0.5500, 1.4500)	428	14.23580
	Jiao and Liu [12]	1.34783	(1.0167, 0.5500, 1.4500)	22	1.01556
	Wang et al. [24]	1.34783	(1.0167, 0.5500, 1.4500)	19	1.26376
	Ours	1.34783	(1.0167, 0.5500, 1.4500)	17	0.89739
4	Feng et al. [8]	2.40000	(1.0167, 0.5500, 1.4500)	494	17.10538
	Jiao and Liu [12]	2.40000	(1.0167, 0.5500, 1.4500)	23	0.93779
	Wang et al. [24]	2.40000	(1.0167, 0.5500, 1.4500)	19	1.33390
	Ours	2.40000	(1.0167, 0.5500, 1.4500)	43	2.13518
5	Feng et al. [8]	1.16157	(1.0000, 0.5500, 1.4500)	313	10.15298
	Jiao and Liu [12]	1.16157	(1.0000, 0.5500, 1.4500)	21	0.85291
	Wang et al. [24]	1.16157	(1.0000, 0.5500, 1.4500)	17	1.18794
	Ours	1.16157	(1.0000, 0.5500, 1.4500)	19	0.88201
6	Feng et al. [8]	0.98971	(1.3452, 0.5000, 1.9464)	2272	113.77662
	Jiao and Liu [12]	0.98971	(1.3452, 0.5000, 1.9465)	40	1.84189
	Wang et al. [24]	0.98971	(1.3452, 0.5000, 1.9465)	39	2.04739
	Ours	0.99279	(1.3500, 0.5000, 1.9417)	91	4.06822
7	Feng et al. [8]	1.11789	(1.5054, 0.3500, 1.5500)	2076	96.72385
	Jiao and Liu [12]	1.11789	(1.5054, 0.3500, 1.5500)	39	1.58835
	Wang et al. [24]	1.11789	(1.5054, 0.3500, 1.5500)	41	2.08152
	Ours	1.12533	(1.4844, 0.3500, 1.5500)	98	4.67947
8	Feng et al. [8]	1.11838	(1.7538, 0.3500, 1.5500)	10326	477.60639
	Jiao and Liu [12]	1.11838	(1.7538, 0.3500, 1.5500)	46	1.77758
	Wang et al. [24]	1.11838	(1.7538, 0.3500, 1.5500)	60	2.60233
	Ours	1.13750	(1.4546, 0.3500, 1.3967)	149	6.75431

Table 2: Numerical results of our algorithm on Example 9 with the large-size variables.

(p, m, n)	Avg.N	Avg.T
(2, 100, 1000)	84.4	25.780946
(2, 100, 2000)	150.6	197.582295
(2, 100, 3000)	136.6	385.307614
(2, 100, 4000)	171.4	837.706223
(2, 100, 5000)	77.8	476.919714
(2, 100, 6000)	137.1	1418.259703
(2, 100, 7000)	140.7	2077.776716
(2, 100, 8000)	193.6	4235.722436
(2, 100, 10000)	80.2	2002.843398
(3, 100, 1000)	536.8	164.539888
(3, 100, 2000)	598.1	687.34879
(3, 100, 3000)	535.3	1249.842719
(3, 100, 4000)	760.7	3928.820802
(3, 100, 5000)	486.4	2941.007123
(3, 100, 6000)	558.1	5126.950975

Example 4 (see [12, 14]).

$$\begin{cases} \min \quad \max\left\{\dfrac{3x_1 + x_2 - 2x_3 + 0.8}{2x_1 - x_2 + x_3}, \dfrac{4x_1 - 2x_2 + x_3}{7x_1 + 3x_2 - x_3}, \dfrac{3x_1 + 2x_2 - x_3 + 1.9}{x_1 - x_2 + x_3}, \dfrac{4x_1 - x_2 + x_3}{8x_1 + 4x_2 - x_3}\right\} \\ \quad\quad\quad x_1 + x_2 - x_3 \leq 1, \\ \quad\quad\quad -x_1 + x_2 - x_3 \leq -1, \\ \quad\quad\quad 12x_1 + 5x_2 + 12x_3 \leq 34.8, \\ \text{s.t.} \quad 12x_1 + 12x_2 + 7x_3 \leq 29.1, \\ \quad\quad\quad -6x_1 + x_2 + x_3 \leq -4.1, \\ \quad\quad\quad 1.0 \leq x_1 \leq 1.2, 0.55 \leq x_2 \leq 0.65, 1.35 \leq x_3 \leq 1.45. \end{cases} \tag{25}$$

Example 5 (see [12, 14]).

$$\begin{cases} \min \quad \max\left\{\dfrac{2.1x_1 + 2.2x_2 - x_3 + 0.8}{1.1x_1 - x_2 + 1.2x_3}, \dfrac{3.1x_1 - x_2 + 1.3x_3}{8.2x_1 + 4.1x_2 - x_3}\right\} \\ \quad\quad\quad x_1 + x_2 - x_3 \leq 1, \\ \quad\quad\quad -x_1 + x_2 - x_3 \leq -1, \\ \quad\quad\quad 12x_1 + 5x_2 + 12x_3 \leq 40, \\ \text{s.t.} \quad 12x_1 + 12x_2 + 7x_3 \leq 50, \\ \quad\quad\quad -6x_1 + x_2 + x_3 \leq -2, \\ \quad\quad\quad 1.0 \leq x_1 \leq 1.2, 0.55 \leq x_2 \leq 0.65, 1.35 \leq x_3 \leq 1.45. \end{cases} \tag{26}$$

Example 6 (see [12, 14]).

$$\begin{cases} \min \quad \max\left\{\dfrac{3x_1+4x_2-x_3+0.5}{2x_1-x_2+x_3+0.5}, \dfrac{3x_1-x_2+3x_3+0.5}{9x_1+5x_2-x_3+0.5}, \dfrac{4x_1-x_2+5x_3+0.5}{11x_1+6x_2-x_3}, \dfrac{5x_1-x_2+6x_3+0.5}{12x_1+7x_2-x_3+0.9}\right\} \\ \text{s.t.} \quad \begin{array}{l} x_1+x_2-x_3\le 1, \\ -x_1+x_2-x_3\le -1, \\ 12x_1+5x_2+12x_3\le 42, \\ 12x_1+12x_2+7x_3\le 55, \\ -6x_1+x_2+x_3\le -3, \\ 1.0\le x_1\le 2.0, 0.50\le x_2\le 2.0, 0.50\le x_3\le 2.0. \end{array} \end{cases} \tag{27}$$

Example 7 (see [12, 14]).

$$\begin{cases} \min \quad \max\left\{\dfrac{3x_1+4x_2-x_3+0.9}{2x_1-x_2+x_3+0.5}, \dfrac{3x_1-x_2+3x_3+0.5}{9x_1+5x_2-x_3+0.5}, \dfrac{4x_1-x_2+5x_3+0.5}{11x_1+6x_2-x_3+0.9}, \dfrac{5x_1-x_2+6x_3+0.5}{12x_1+7x_2-x_3+0.9}, \dfrac{6x_1-x_2+7x_3+0.6}{11x_1+6x_2-x_3+0.9}\right\} \\ \text{s.t.} \quad \begin{array}{l} 2x_1+x_2-x_3\le 2, \\ -2x_1+x_2-2x_3\le -1, \\ 11x_1+6x_2+12x_3\le 45, \\ 11x_1+13x_2+6x_3\le 52, \\ -7x_1+x_2+x_3\le -2, \\ 1.0\le x_1\le 2.0, 0.35\le x_2\le 0.9, 1.0\le x_3\le 1.55. \end{array} \end{cases} \tag{28}$$

Example 8 (see [12, 14]).

$$\begin{cases} \min \quad \max\left\{\dfrac{5x_1+4x_2-x_3+0.9}{3x_1-x_2+2x_3+0.5}, \dfrac{3x_1-x_2+4x_3+0.5}{9x_1+3x_2-x_3+0.5}, \dfrac{4x_1-x_2+6x_3+0.5}{12x_1+7x_2-x_3+0.9}, \dfrac{7x_1-x_2+7x_3+0.5}{11x_1+9x_2-x_3+0.9}, \dfrac{7x_1-x_2+7x_3+0.7}{11x_1+7x_2-x_3+0.8}\right\} \\ \text{s.t.} \quad \begin{array}{l} 2x_1+2x_2-x_3 \le 3, \\ -2x_1+x_2-3x_3 \le -1, \\ 11x_1+7x_2+12x_3 \le 47, \\ 13x_1+13x_2+6x_3 \le 56, \\ -6x_1+2x_2+3x_3 \le -1, \\ 1.0 \le x_1 \le 2.0, 0.35 \le x_2 \le 0.9, 1.0 \le x_3 \le 1.55. \end{array} \end{cases} \tag{29}$$

Example 9

$$\begin{cases} \min \quad \max\left\{\dfrac{\sum_{j=1}^n d_{1j}x_j+g_1}{\sum_{j=1}^n e_{1j}x_j+h_1}, \dfrac{\sum_{j=1}^n d_{2j}x_j+g_2}{\sum_{j=1}^n e_{2j}x_j+h_2}, \dots, \dfrac{\sum_{j=1}^n d_{pj}x_j+g_p}{\sum_{j=1}^n e_{pj}x_j+h_p}\right\} \\ \text{s.t.} \quad \begin{array}{l} \displaystyle\sum_{j=1}^n a_{kj}x_j \le b_k, \quad k=1,2,\dots,m, \\ x_j \ge 0, \quad j=1,2,\dots,n, \end{array} \end{cases} \tag{30}$$

where all d_{ij} and e_{ij}, $i=1,2,\dots,p$ and $j=1,2,\dots,n$, and b_k and a_{kj}, $k=1,2,\dots,m$ and $j=1,2,\dots,n$, are randomly generated in $[0,10]$; all g_i and h_i, $i=1,2,\dots,p$, are randomly generated in $[0,1]$.

From the numerical results in Tables 1 and 2, it is seen that the proposed algorithm has higher computational efficiency than the algorithms presented in the works of Feng et al. [8], Jiao and Liu [12], and Wang et al. [24], which can be used to globally solve the problem (MLFP) with the large-size variables.

5. Concluding Remarks

In this paper, based on the outer space partitioning search and the new linear relaxation problem, we present an outer space branch-and-bound algorithm for globally solving the problem (MLFP). The proposed algorithm is convergent to the global optimal value by the successive refinement of the outer space region and the subsequent solutions of a series of linear relaxation programming problems over the outer space region. The main work of the proposed algorithm involves solving a series of linear relaxation programming problems over the outer space region which does not grow in size for each iteration and which can be efficiently solved by the simplex method. Compared with the algorithms presented in the works of Feng et al. [8], Jiao and Liu [12], and Wang et al. [24], numerical results for some test examples are given to illustrate the feasibility and effectiveness of our new algorithm. It is also hoped that the ideas and methods used to create the algorithm will offer a useful tool for globally solving the problem (MLFP).

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (11871196 and 12071133), China Postdoctoral Science Foundation (2017M622340), Key Scientific and Technological Research Projects in Henan Province (202102210147 and 192102210114), Science and Technology Climbing Program of Henan Institute of Science and Technology (2018JY01), and Henan Institute of Science and Technology Postdoctoral Science Foundation.

References

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

[2] F. Ding, "Coupled-least-squares identification for multivariable systems," *IET Control Theory & Applications*, vol. 7, no. 1, pp. 68–79, 2013.

[3] 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 & Applications*, vol. 7, no. 2, pp. 176–184, 2013.

[4] W. Wang, J. Li, and R. Ding, "Maximum likelihood parameter estimation algorithm for controlled autoregressive autoregressive models," *International Journal of Computer Mathematics*, vol. 88, no. 16, pp. 3458–3467, 2011.

[5] C. D. Maranas, I. P. Androulakis, C. A. Floudas, A. J. Berger, and J. M. Mulvey, "Solving long-term financial planning problems via global optimization," *Journal of Economic Dynamics and Control*, vol. 21, no. 8-9, pp. 1405–1425, 1997.

[6] R. W. Freund and F. Jarre, "An interior-point method for fractional programs with convex constraints," *Mathematical Programming*, vol. 67, no. 1-3, pp. 407–440, 1994.

[7] J. P. Crouzeix, J. A. Ferland, and S. Schaible, "An algorithm for generalized fractional programs," *Journal of Optimization Theory and Applications*, vol. 47, pp. 135–149, 1985.

[8] Q. G. Feng, H. P. Mao, and H. W. Jiao, "A feasible method for a class of mathematical problems in manufacturing system," *Key Engineering Materials*, vol. 460-461, pp. 806–809, 2011.

[9] 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.

[10] 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.

[11] 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.

[12] H. Jiao and S. Liu, "A new linearization technique for minimax linear fractional programming," *International Journal of Computer Mathematics*, vol. 91, no. 8, pp. 1730–1743, 2014.

[13] I. Ahmad and Z. Husain, "Duality in nondifferentiable minimax fractional programming with generalized convexity," *Applied Mathematics and Computation*, vol. 176, no. 2, pp. 545–551, 2006.

[14] C. Bajona-Xandri and J. E. MartÍnez-Legaz, "Lower subdifferentiability in minimax fractional programming," *Optimization*, vol. 45, no. 1-4, pp. 1–12, 1999.

[15] P. Balasubramaniam and S. Lakshmanan, "Delay-interval-dependent robust-stability criteria for neutral stochastic neural networks with polytopic and linear fractional uncertainties," *International Journal of Computer Mathematics*, vol. 88, no. 10, pp. 2001–2015, 2011.

[16] F. Ding, "Decomposition based fast least squares algorithm for output error systems," *Signal Processing*, vol. 93, no. 5, pp. 1235–1242, 2013.

[17] F. Ding, "Two-stage least squares based iterative estimation algorithm for CARARMA system modeling," *Applied Mathematical Modelling*, vol. 37, no. 7, pp. 4798–4808, 2013.

[18] J. Zhu, H. Liu, and B. Hao, "A new semismooth Newton method for NCPs based on the penalized KK function," *International Journal of Computer Mathematics*, vol. 89, no. 4, pp. 543–560, 2012.

[19] Y. Liu and R. Ding, "Consistency of the extended gradient identification algorithm for multi-input multi-output systems with moving average noises," *International Journal of Computer Mathematics*, vol. 90, no. 9, pp. 1840–1852, 2013.

[20] H. J. Chen, S. Schaible, and R. L. Sheu, "Generic algorithm for generalized fractional programming," *Journal of Optimization Theory and Applications*, vol. 141, no. 1, pp. 93–105, 2009.

[21] H. Boualam, A. Roubi, and A. Roubi, "Dual algorithms based on the proximal bundle method for solving convex minimax fractional programs," *Journal of Industrial and Management Optimization*, vol. 15, no. 4, pp. 1897–1920, 2019.

[22] K. Boufi and A. Roubi, "Dual method of centers for solving generalized fractional programs," *Journal of Global Optimization*, vol. 69, no. 7, pp. 387–426, 2017.

[23] H.-C. Lai and T.-Y. Huang, "Nondifferentiable minimax fractional programming in complex spaces with parametric duality," *Journal of Global Optimization*, vol. 53, no. 2, pp. 243–254, 2012.

[24] C.-F. Wang, Y. Jiang, and P.-P. Shen, "A new branch-and-bound algorithm for solving minimax linear fractional programming," *Jurnal Matematika*, vol. 38, no. 1, pp. 113–123, 2018.

[25] R. Horst and H. Tuy, *Global Optimization, Deterministic Approaches*, Springer-Verlag, Berlin, Germany, 1990.