International Journal of Computer Mathematics, 2014
Vol. 91, No. 8, 1730–1743, http://dx.doi.org/10.1080/00207160.2013.860449

Taylor & Francis
Taylor & Francis Group

A new linearization technique for minimax linear fractional programming

Hongwei Jiao and Sanyang Liu*

Department of Mathematics, Xidian University, Xi'an 710071, China

(*Received 1 June 2013; second revision received 29 July 2013; third revision received 24 October 2013; fourth revision received 21 October 2013; accepted 22 October 2013*)

This paper presents a deterministic global optimization algorithm for solving minimax linear fractional programming (MLFP). In this algorithm, a new linearization technique is proposed, which uses more information of the objective function of the (MLFP) than other techniques. By utilizing this new linearization technique, the initial nonconvex programming problem (MLFP) is reduced to a sequence of linear relaxation programming (LRP) problems, which can provide reliable lower bound of the optimal value. The proposed algorithm is convergent to the global minimum of the (MLFP) through the successive refinement of the feasible region and solutions of a series of the (LRP). Compared with the known algorithms, numerical results show that the proposed algorithm is robust and effective.

Keywords: global optimization; minimax linear fractional programming; linearization technique; branch-and-bound; linear relaxation programming

2010 AMS Subject Classifications: 90C30; 90C32; 65K05

1. Introduction

The minimax linear fractional programming (MLFP) problems can be formulated as the following nonlinear optimization problems:

$$(\text{MLFP}): \begin{cases} \min \quad \max\left\{\dfrac{n_1(x)}{d_1(x)}, \dfrac{n_2(x)}{d_2(x)}, \ldots, \dfrac{n_p(x)}{d_p(x)}\right\} \\ \text{s.t.} \quad x \in D = \{x \in R^n \mid Ax \leq b, x \geq 0\}, \end{cases}$$

where $A \in R^{m\times n}, b \in R^m$, $n_j(x)$ and $d_j(x), j = 1, \ldots, p$, are all linear affine functions, D is a nonempty compact set.

The problem (MLFP) is a special class of fractional programming problem, which has attracted the interest of practitioners and researchers for at least 30 years [1,26,30,33,36,44]. During the past 20 years, interest in these problems has been especially intense. In part, this is because since its initial development it has spawn a wide variety of applications, specially in measuring the efficiency or productivity of system [37], the design of electronic circuits [4], signal processing [11], neural networks [3], optimal design [2], iterative parameter estimation [13], financial planning

*Corresponding author. Email: liusanyang@126.com

[20], system identification [10,12,14,15,32,39] and so on. Another reason for the strong interest in this class of problems is that from a research point of view, these problems pose significant theoretical and computational challenges. This is mainly because these problems are global optimization problems, i.e. they are known to generally possess multiple local optima that are not globally optima. So it is necessary to put forward good solution method for the (MLFP).

During the past years, many algorithms have been proposed for globally solving MLFP. For instance, parametric programming method [9], partial linearization algorithm [6], interior-point algorithm [18], monotonic optimization approach [34], cutting plane algorithm [5], branch-and-bound algorithms [16,17], method of centres [35], dinkelbach-type algorithms [7,31], prox-regularization method [21]. In recent several years, some significant progress in this area of theoretical research has been obtained for solving MLFP or nonlinear fractional programming. For examples, when data both in the objective and constraints are uncertainty, Jeyakumar *et al.* [25] proposed a strong duality for robust minimax fractional programming problems; Gao and Rong [19] established the optimality conditions and duality for a class of nondifferentiable multiobjective generalized fractional programming problems; using a parametric approach, Lai *et al.* [29] presented the Kuhn–Tucker type necessary optimality conditions and proved the existence theorem of optimality for complex minimax fractional programming in the framework of generalized convexity; Zheng and Cheng [43] derived the Kuhn–Tucker type sufficient optimality conditions and established weak, strong and converse duality theorems for minimax fractional programming under nonsmooth generalized (F, ρ, θ)-d-univexity; Lai and Huang [27] established the sufficient optimality conditions for a minimax programming problem involving p fractional n-set functions under generalized invexity; Husain *et al.* [24] presented a second-order duality theorems for minimax fractional programming. In addition, in this area of algorithm research, when the denominator and numerator of each ratio are all continuous real-valued functions, Chen *et al.* [8] proposed a generic algorithm for generalized fractional programming; Strodiot *et al.* [38] proposed an inexact proximal point method for solving generalized fractional program with convex data; Wen and Wu [42] presented a parametric approach to solve the continuous-time linear fractional max–min problems. Recently, Gupta and Dangar [22] and Hu *et al.* [23] proposed two different second-order dualities for nondifferentiable minimax fractional programming problem, respectively; Lai and Huang [28] presented optimality conditions and parametric duality for nondifferentiable minimax fractional programming problem; Wen [40] derived some properties of the auxiliary parametric continuous-time generalized fractional programming problem, and concluded that solving the class of problem is equivalent to determine the root of the nonlinear equation. These properties make it possible to develop a numerical algorithm for solving the continuous-time generalized fractional programming problem; based on Ref. [40], Wen [41] presented an interval-type computational procedure for solving the continuous-time generalized fractional programming problem. But the most of the above literatures only presented optimality conditions or duality theory for MLFP problem, or can only find its local optimal solution, or can only solve the special form of MLFP problems. Therefore, in addition to the above-mentioned algorithms, up to now, although there has been significant progress in the development of theories for solving MLFP or nonlinear fractional programming, to our knowledge, less work has been still done for globally solving MLFP problem investigated in this paper.

In this paper, we present a branch-and-bound algorithm for globally solving MLFP problem. First, we transform the problem (MLFP) into an equivalent problem (EQ). Second, in order to obtain the lower bound of the problem (EQ), a new linearization technique is proposed for establishing linear relaxation programming (LRP) problem of the (EQ), which is incorporated into the branch-and-bound framework. Third, compared with the linear relaxation method in Ref. [17], the proposed new linearization technique uses more information of the objective function of the (MLFP). Compared with the similar previous linear relaxation techniques in Refs. [16,17], the new linearization technique for computing lower bounds can provide a tighter lower bound, which can suppress the rapid growth of branching tree in the branch-and-bound search procedure for solving the (MLFP) to improve the computational efficiency of the algorithm. Finally, the numerical experimental results show that the proposed algorithm has higher computational efficiency than Refs. [16,17], and can be used to globally solve the MLFP problems with large scale of fractional objective functions.

The remainder of this article is organized as follows. In Section 2, by using a transformation technique, the problem (EQ) is derived that is equivalent to the problem (MLFP). In Section 3, a new linearization technique is presented, then the LRP of the (EQ) is established. A rectangular branching rule and an algorithm for globally solving the (MLFP) are introduced in Section 4. In Section 5, the numerical results for some test examples in recent literatures with the proposed algorithm are reported. Finally, a few concluding remarks are given in Section 6.

2. Preliminaries

To globally solve the problem (MLFP), for each $i = 1, 2, \ldots, n$, we need to compute the initial lower bound $\underline{x}_i = \min_{x \in D} x_i$ and upper bound $\bar{x}_i = \max_{x \in D} x_i$ of each variable x_i by solving linear programming problems, then we can derive an initial rectangle

$$X^0 = \{x | \underline{x}_i \leq x_i \leq \bar{x}_i,\ i = 1, \ldots, n\}.$$

For each $j \in \{1, \ldots, p\}$, let $F_j(x) = n_j(x)/d_j(x)$, we can establish the equivalent problem (EQ) of the (MLFP) as follows.

$$\text{(EQ)} : \begin{cases} \min & t \\ \text{s.t.} & F_j(x) - t \leq 0, j = 1, \ldots, p, \\ & Ax \leq b,\ x \in X^0. \end{cases}$$

Lemma 1 *The problems (MLFP) and (EQ) have the same global optimal solutions and optimal value.*

Proof The proof of this lemma follows easily from the monotonic character of function, here it is omitted. ■

Lemma 2 *For each $j \in \{1, \ldots, p\}$, by the continuity of the function $n_j(x)/d_j(x)$, we have $d_j(x) \neq 0$.*

Proof The proof of this lemma can be easily followed from the continuity of the function $d_j(x)$, therefore, it is omitted. ■

By Lemma 2, we have $d_j(x) > 0$ or $d_j(x) < 0$. If $d_j(x) < 0$, through letting $n_j(x)/d_j(x) = -n_j(x)/-d_j(x)$, obviously, the denominator can be transformed into a positive value function. Hence, in the problem (EQ), we can always assume that $d_j(x) > 0$ holds. In addition, for each $j \in \{1, \ldots, p\}$, if $n_j(x)$ is an arbitrary function, there always exist a large enough positive number M_j such that $n_j(x) + M_j d_j(x) > 0$. since

$$\frac{n_j(x)}{d_j(x)} = \frac{n_j(x) + M_j d_j(x)}{d_j(x)} - M_j,$$

therefore, in the following, without loss of generality, we can always assume that $n_j(x) > 0$ and $d_j(x) > 0, j = 1, \ldots, p$.

In the following, we will only consider solving the problem (EQ), the principal construction in the development of a solution procedure for solving the problem (EQ) is construction of a LRP for obtaining the lower bounds of the optimal value for this problem. For the problem (EQ), we only need to construct a linear lower bounding function of each constraint function $F_j(x)$. The developed method uses a new linearization technique to derive the linear lower bounding function of every $F_j(x),\ j=1,\dots,p$.

3. New linearization technique

In this section, for each $F_j(x)$ in constraints, we will construct its linear lower bounding function. By the above assumption, we can let

$$F_j(x)=\exp[\ln(n_j(x))-\ln(d_j(x))].$$

First, for $\forall x\in X^k\subseteq X^0$, for any $j\in\{1,\dots,p\}$, some notations can be introduced as follows:

$$n_j^l=\min_{x\in X^k} n_j(x),\quad n_j^u=\max_{x\in X^k} n_j(x),$$
$$d_j^l=\min_{x\in X^k} d_j(x),\quad d_j^u=\max_{x\in X^k} d_j(x),$$
$$C_j=\frac{\ln(n_j^u)-\ln(n_j^l)}{n_j^u-n_j^l},\quad D_j=\frac{\ln(d_j^u)-\ln(d_j^l)}{d_j^u-d_j^l}.$$

For the concave function $\ln(Z)$, we can construct its linear lower bounding function $L(\ln(Z))$ and linear upper bounding function $U(\ln(Z))$ over the interval $[Z^l,Z^u]$ as follows:

$$\begin{aligned}L(\ln(Z))&=C(Z-Z^l)+\ln Z^l,\\ U(\ln(Z))&=CZ-1-\ln C,\end{aligned}\tag{1}$$

such that

$$L(\ln(Z))\le\ln(Z)\le U(\ln(Z)),\tag{2}$$

where

$$C=\frac{\ln Z^u-\ln Z^l}{Z^u-Z^l},$$

Based on the above discussion, for any $j\in\{1,\dots,p\}$, substituting the above notations Z, Z^u, Z^l and C, in the forms (1) and (2), by the corresponding notations $n_j(x)$, n_j^u, n_j^l, C_j, $d_j(x)$, d_j^u, d_j^l and D_j, then we can obtain the following inequalities:

$$C_j[n_j(x)-n_j^l]+\ln n_j^l\le\ln(n_j(x))\le C_jn_j(x)-1-\ln C_j,\tag{3}$$

$$D_j[d_j(x)-d_j^l]+\ln d_j^l\le\ln(d_j(x))\le D_jd_j(x)-1-\ln D_j,\tag{4}$$

By the inequalities (3) and (4), then finally we can derive the lower bounding function $F_j^l(x)$ and the upper bounding function $F_j^u(x)$ of the function $F_j(x)$ for each $j\in\{1,\dots,p\}$, which underestimates and overestimates the value of the function $F_j(x)$ as follows:

$$F_j^l(x) = \exp\{C_j[n_j(x) - n_j^l] + \ln n_j^l - [D_j d_j(x) - 1 - \ln D_j]\},$$
$$F_j^u(x) = \exp\{C_j n_j(x) - 1 - \ln C_j - D_j[d_j(x) - d_j^l] - \ln d_j^l\},$$

such that

$$F_j^l(x) \le F_j(x) \le F_j^u(x), \text{ for } \forall\, x \in X^k \subseteq X^0.$$

Second, for $\forall x \in X^k$, we first let

$$Z_j = C_j[n_j(x) - n_j^l] + \ln n_j^l - [D_j d_j(x) - 1 - \ln D_j],$$
$$Z_j^u = \max_{x\in X^k}\{C_j[n_j(x) - n_j^l] + \ln n_j^l - [D_j d_j(x) - 1 - \ln D_j]\},$$
$$Z_j^l = \min_{x\in X^k}\{C_j[n_j(x) - n_j^l] + \ln n_j^l - [D_j d_j(x) - 1 - \ln D_j]\},$$
$$B_j = \frac{\exp(Z_j^u) - \exp(Z_j^l)}{Z_j^u - Z_j^l}.$$

For each convex function $\exp(Z_j)$, we can construct its linear lower bounding function $L(\exp(Z_j))$ over the interval $[Z_j^l, Z_j^u]$ as follows:

$$L_j(\exp(Z_j)) = B_j(1 + Z_j - \ln B_j), \tag{5}$$

such that

$$L_j(\exp(Z_j)) \le \exp(Z_j). \tag{6}$$

Based on the above discussion, for each $j \in \{1, \dots, p\}$, by Equations (5) and (6), then finally we derive the linear lower bounding function $L_j(x)$ of $F_j^l(x)$, which underestimates the value of the function $F_j^l(x)$ as follows:

$$L_j(x) = B_j\{1 + C_j[n_j(x) - n_j^l] + \ln n_j^l - [D_j d_j(x) - 1 - \ln D_j] - \ln B_j\},$$

such that

$$L_j(x) \le F_j^l(x), \quad \text{for } \forall\, x \in X^k \subseteq X^0.$$

According to the above linearization technique, for $\forall X^k \subseteq X^0$, we can construct the LRP problem of the (EQ) in X^k as follows:

$$(\text{LRP}): \begin{cases} \min & t \\ \text{s.t.} & L_j(x) - t \le 0,\ j = 1, \dots, p, \\ & Ax \le b, \\ & x \in X^k. \end{cases}$$

THEOREM 1 *For* $\forall x \in X^k = [\underline{x}, \bar{x}] \subseteq X^0$, *let* $u_j = n_j^u/n_j^l$, $v_j = d_j^u/d_j^l$, *then the error* $\Theta_j = F_j(x) - L_j(x) \to 0$ *as* $\|\bar{x} - \underline{x}\| \to 0$.

Proof Let

$$\Theta_j = [F_j(x) - F_j^l(x)] + [F_j^l(x) - L_j(x)] = \Theta_{j1} + \Theta_{j2},$$

then, first, we consider the difference Θ_{j1}. Let

$$G_j(x) = C_j n_j(x) - 1 - \ln C_j - D_j[d_j(x) - d_j^l] - \ln d_j^l,$$
$$H_j(x) = C_j[n_j(x) - n_j^l] + \ln n_j^l - [D_j d_j(x) - 1 - \ln D_j],$$

it follows

$$\begin{aligned}\Theta_{j1} &= F_j(x) - F_j^l(x)\\ &\le F_j^u(x) - F_j^l(x)\\ &= \exp\{C_j n_j(x) - 1 - \ln C_j - D_j[d_j(x) - d_j^l] - \ln d_j^l\}\\ &\quad - \exp\{C_j[n_j(x) - n_j^l] + \ln n_j^l - [D_j d_j(x) - 1 - \ln D_j]\}\\ &= \exp(G_j(x)) - \exp(H_j(x))\\ &\le \|G_j(x) - H_j(x)\| \sup_{\xi_j \in L(G_j(x), H_j(x))} \exp(\xi_j)\end{aligned}$$

where

$$L(G_j(x), H_j(x)) = \alpha G_j(x) + (1-\alpha)H_j(x), \quad \text{for } \forall\alpha \in [0,1].$$

Let

$$\begin{aligned}G_j(x) - H_j(x) &= \{C_j n_j(x) - 1 - \ln C_j - D_j[d_j(x) - d_j^l] - \ln d_j^l\}\\ &\quad - \{C_j[n_j(x) - n_j^l] + \ln n_j^l - [D_j d_j(x) - 1 - \ln D_j]\}\\ &= \{[C_j n_j(x) - 1 - \ln C_j] - [C_j[n_j(x) - n_j^l] + \ln n_j^l]\}\\ &\quad + \{[D_j d_j(x) - 1 - \ln D_j] - [D_j[d_j(x) - d_j^l] + \ln d_j^l]\}\\ &= \{[C_j n_j(x) - 1 - \ln C_j] - \ln(n_j(x))\}\\ &\quad + \{\ln(n_j(x)) - [C_j[n_j(x) - n_j^l] + \ln n_j^l]\}\\ &\quad + \{[D_j d_j(x) - 1 - \ln D_j] - \ln(d_j(x))\}\\ &\quad + \{\ln(d_j(x)) - [D_j[d_j(x) - d_j^l] + \ln d_j^l]\}\\ &= \Theta_{j1.1} + \Theta_{j1.2} + \Theta_{j1.3} + \Theta_{j1.4}.\end{aligned}$$

Since $\Theta_{j1.1}$ is a convex function about $n_j(x)$, it follows that it can attain the maximum $\Theta_{j1.1}^{\max}$ at the point n_j^u or n_j^l. Then through computing, we derive

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

Since $\Theta_{j1.2}$ is a concave function about $n_j(x)$, we can know $\Theta_{j1.2}$ can attain the maximum $\Theta_{j1.2}^{\max}$ at the point $n_j(x) = 1/C_j$. Then through computing, we derive

$$\Theta_{j1.2}^{\max} = \frac{\ln u_j}{u_j - 1} - 1 - \ln\frac{\ln u_j}{u_j - 1}.$$

Since $u_j \to 1$ as $\|\bar{x} - \underline{x}\| \to 0$, then we have $\Theta_{j1.1}^{\max} \to 0$ and $\Theta_{j1.2}^{\max} \to 0$ as $\|\bar{x} - \underline{x}\| \to 0$.

Similarly, we can prove that, since $v_j \to 1$ as $\|\bar{x} - \underline{x}\| \to 0$, then we have

$$\Theta_{j1.3}^{\max} = \Theta_{j1.4}^{\max} = \frac{\ln v_j}{v_j - 1} - 1 - \ln \frac{\ln v_j}{v_j - 1} \to 0 \quad \text{as } \|\bar{x} - \underline{x}\| \to 0.$$

Therefore, we have

$$\|G_j(x) - H_j(x)\| = \|\Theta_{j1.1} + \Theta_{j1.2} + \Theta_{j1.3} + \Theta_{j1.4}\| \le \|\Theta_{j1.1}\| + \|\Theta_{j1.2}\| + \|\Theta_{j1.3}\| + \|\Theta_{j1.4}\|.$$

By the above proof, we have

$$\|G_j(x) - H_j(x)\| \to 0, \quad \text{as } \|\bar{x} - \underline{x}\| \to 0.$$

Since $\exp(\xi_j)$ is a continuous and bounded function about variable x, we have

$$\Theta_{j1} \to 0, \quad \text{as } \|\bar{x} - \underline{x}\| \to 0.$$

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

$$\begin{aligned}\Theta_{j2} &= F_j^l(x) - L_j(x)\\ &= \exp\{C_j[n_j(x) - n_j^l] + \ln(n_j^l) - [D_j d_j(x) - 1 - \ln D_j]\} - B_j\{1 + C_j[n_j(x) - n_j^l]\\ &\quad + \ln(n_j^l) - [D_j(d_j(x)) - 1 - \ln D_j] - \ln B_j\}\\ &= \exp(Z_j) - B_j\{1 + Z_j - \ln B_j\}\end{aligned}$$

Since Θ_{j2} is a convex function about Z_j, for any $Z_j \in [Z_j^l, Z_j^u]$ defined in the former. Then, it follows that Θ_{j2} can obtain the maximum $\Theta_{j2}^{\max}$ at the points Z_j^l or Z_j^u. Let

$$T_j = \frac{\exp(Z_j^u - Z_j^l) - 1}{Z_j^u - Z_j^l},$$

then through computing, we can derive the following form:

$$\Theta_{j2}^{\max} = \Theta_{j2}(Z_j^l) = \Theta_{j2}(Z_j^u) = \exp(Z_j^l)(1 - T_j + T_j \ln T_j).$$

Since $T_j \to 1$ as $|Z_j^u - Z_j^l| \to 0$, and $|Z_j^u - Z_j^l| \to 0$ as $\|\bar{x} - \underline{x}\| \to 0$. So it is obvious that $\Theta_{j2}^{\max} \to 0$ as $\|\bar{x} - \underline{x}\| \to 0$. Therefore, we have $\Theta_{j2} \to 0$ as $\|\bar{x} - \underline{x}\| \to 0$.

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

The above theorem ensures each $L_j(x)$ will approximate the corresponding function $F_j(x)$ as $\|\bar{x} - \underline{x}\| \to 0$.

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

4. Algorithm and its convergence

In this section, we present a branch-and-bound algorithm for globally solving the (EQ). The critical element in guaranteeing convergence to a global minimum is the choice of a suitable partitioning strategy. In this paper, we choose a standard bisection rule. This branching rule is given as follows. Consider any node sub-problem identified by the hyper-rectangle $X = [\underline{x}, \bar{x}] \subseteq X^0$, let $q \in \arg\max\{\bar{x}_i - \underline{x}_i : i = 1, \ldots, n\}$ and partition X by subdividing the interval $[\underline{x}_q, \bar{x}_q]$ into the subintervals $[\underline{x}_q, (\underline{x}_q + \bar{x}_q)/2]$ and $[(\underline{x}_q + \bar{x}_q)/2, \bar{x}_q]$.

Let $LB(X^k)$ be the optimal value of the (LRP) on the sub-hyper-rectangles X^k and $x^k = x(X^k)$ be an element of corresponding argmin. In order to validate the robustness and efficiency of our method, here we use the same algorithm step as Refs. [16,17] with new LRP problem, the basic steps of the algorithm are summarized as follows.

Step 1. (*Initialization*)

Initialize the convergence tolerance ε; the feasible error ϵ_1; the iteration counter $k := 0$; the set of active node $\Omega_0 = X^0$; the upper bound $\text{UB} = +\infty$; the set of feasible points $F := \emptyset$.

Solve the $\text{LRP}(X^0)$, obtain $\text{LB}_0 := \text{LB}(X^0)$ and (x^0, t^0). With feasible error ϵ_1, if (x^0, t^0) is feasible to the (EQ), update F and UB, if necessary. If $\text{UB} - \text{LB}_0 \leq \varepsilon$, then stop with x^0 as the prescribed solution to the (EQ); otherwise, proceed to Step 2.

Step 2. (*Bounding*)

Let

$$\text{UB} = \min_{(x,t)\in F} t.$$

If $F \neq \emptyset$, then the known best feasible solution is

$$\tilde{x} = \arg\min_{(x,t)\in F} \max\{F_1(x), F_2(x), \ldots, F_p(x)\}.$$

Step 3. (*Branching*)

Partition X^k into two new sub-hyper-rectangles according to the above branching rule. Call the set of new partitioned rectangles as $\bar{X}^k$.

Step 4. (*Bounding*)

For each $X \in \bar{X}^k$, compute the lower bound $\text{LB}(X)$ and $(x(X), t(X))$ by solving the $\text{LRP}(X)$. If $\text{LB}(X) > UB$, then let $\bar{X}^k := \bar{X}^k \backslash X$, else if $(x(X), t(X))$ is feasible to the (EQ) with feasible error ϵ_1, then update UB, F and $\tilde{x}$, if necessary, and let

$$\Omega_k = (\Omega_k \backslash X) \cup \bar{X}^k,$$

update lower bound

$$\text{LB}_k = \inf_{X\in\Omega_k} \text{LB}(X).$$

Step 5. (*Termination*)

Let

$$\Omega_{k+1} = \Omega_k \backslash \{X : \text{UB} - \text{LB}(X) \leq \varepsilon, X \in \Omega_k\}.$$

If $\Omega_{k+1} = \emptyset$, then algorithm stops, UB is the global optimal value for the (EQ), and $\tilde{x}$ is a global optimization solution for the (EQ). Otherwise, let $k := k + 1$, select X^k such that $X^k = \arg\min_{X\in\Omega_k} \text{LB}(X)$, return to Step 3.

THEOREM 2 *The above algorithm either terminates finitely with the solution being optimal to the* (MLFP)*, or generates an infinite sequence of iterations such that along any infinite branch of the branch-and-bound tree, and accumulation point of the sequence* $\{x^k\}$ *will be the global optimal solution of the* (MLFP)*.*

Proof The proof of the theorem can be similarly given by the Theorem 3 in Ref. [16]. ■

5. Computational results

In order to compare our algorithm (using new LRP problem) with known algorithms (recent literatures Refs. [16,17,34]) with respect to robustness (finding the optimum), and efficiencies (number of function evaluations), some numerical examples appeared in recent literatures (Refs. [16,34]) and an example randomly generated are implemented on a Intel(R) Core(TM)2 Duo CPU (1.58G HZ) microcomputer. The algorithm is coded in C++ program and each linear programming is solved by using the simplex method, and the convergence tolerance is set to $\epsilon = 5 \times 10^{-8}$ in our experiment. For the test problems, the results obtained using the above algorithm are illustrated in Tables 1–4. For Examples 1–9, feasible error ϵ_1 are set by $0.005, 0.001, 0.001, 0.005, 0.001, 0.001, 0.001, 0.001, 0.001$, respectively.

In Tables 1–4, the notations have been used for column headers: Iter, number of algorithm iterations; $L_{\max}$, the maximal number of algorithm active nodes necessary; Time, execution time of algorithm in seconds.

Example 1 (Refs. [16,34])

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

Example 2 (Refs. [16,34])

$$\begin{aligned}
\max \quad & \min \left\{ \frac{37x_1 + 73x_2 + 13}{13x_1 + 13x_2 + 13}, \frac{63x_1 - 18x_2 + 39}{13x_1 + 26x_2 + 13} \right\} \\
\text{s.t.} \quad & 5x_1 - 3x_2 = 3, \quad 1.5 \le x_1 \le 3.
\end{aligned}$$

Table 1. Numerical comparison with Refs. [16,34] for Examples 1–4.

Example	Ref.	Optimal solution	Optimal value	Iter	$L_{\max}$	Time (s)
1	[34]	(1.015695, 0.590494, 1.403675)	0.573102	1	–	0.06
	[16]	(1.015678086, 0.590676676, 1.403391837)	0.572810738	6	5	0.017700826
	This paper	(1.015569677, 0.591847484, 1.401570580)	0.571392040	1	2	0.00692539
2	[34]	(1.5, 1.5)	1.489510	1	–	0.00
	[16]	(1.5, 1.5)	1.49072061	6	7	0.00680581
	This paper	(1.5, 1.5)	1.49661806	3	4	0.00487959
3	[16]	(1.016666667, 0.55, 1.45)	1.346854863	8	8	0.0228725
	This paper	(1.016666667, 0.55, 1.45)	1.344502171	4	4	0.0144057
4	[16]	(1.008333333, 0.5, 1.45)	2.284427051	7	8	0.0288736
	This paper	(1.008333333, 0.5, 1.45)	2.280126353	3	4	0.0197122

Table 2. Numerical comparison with Ref. [16] for Example 5.

Example 5	Algorithm of [16]			This paper		
(p, M, N)	Iter	$L_{\max}$	Time (s)	Iter	$L_{\max}$	Time (s)
(5, 4, 3)	81	57	0.235426	71	50	0.275371234
(6, 5, 5)	176	162	0.981466	70	59	0.693835
(7, 5, 6)	174	159	1.313451	103	94	1.615823
(7, 5, 7)	61	50	0.596088	44	27	1.078511772
(9, 6, 7)	822	782	10.026984	584	571	16.093457809
(9, 7, 10)	3581	3400	79.042373	2329	2172	49.680818639
(20, 7, 10)	141	74	5.909754	11	12	1.847142886
(45, 7, 10)	146	26	17.423466	18	12	6.381171066
(50, 7, 10)	45	46	7.727302	32	33	12.693666502

Table 3. Numerical comparison with Ref. [17] for Example 5.

Example 5	Algorithm of [17]			This paper		
(p, M, N)	Iter	$L_{\max}$	Time (s)	Iter	$L_{\max}$	Time (s)
(2, 1, 5)	1809	1693	2.83591	46	43	0.232999
(2, 3, 5)	1811	1695	3.63135	42	39	0.230242
(3, 3, 5)	15,978	14,521	64.1806	62	61	0.357087
(4, 3, 3)	330	310	0.53731	17	18	0.037901
(10, 2, 3)	112	107	0.379368	26	27	0.144406
(11, 3, 3)	9213	8505	42.2537	17	18	0.154271
(12, 3, 5)	12,804	12,096	143.786	39	38	0.722472
(18, 3, 5)	1655	1502	17.5244	107	108	2.798912
(25, 10, 4)	677	654	17.5517	19	20	0.612503

Table 4. Numerical comparison with Ref. [17] for Examples 6–9.

Example	Ref.	Optimal solution	Optimal value	Iter	$L_{\max}$	Time (s)
6	[17]	(1.0, 0.55, 1.45)	1.160759760	113	106	0.235226
	This paper	(1.0, 0.55, 1.45)	1.160998779	6	7	0.0104895
7	[17]	(1.339843750, 0.50, 1.943285553)	0.985599329	580	420	1.19123
	This paper	(1.345382850, 0.50, 1.946283817)	0.989117392	21	20	0.0557114
8	[17]	(1.504882813, 0.350, 1.550)	1.117065399	747	638	1.68613
	This paper	(1.504885652, 0.350, 1.550)	1.117070767	20	20	0.0819911
9	[17]	(1.753906250, 0.350, 1.550)	1.117416325	2901	2534	6.52166
	This paper	(1.752859889, 0.350, 1.550)	1.117793086	26	22	0.148766

Example 3 (Ref. [16])

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

$$1.0 \leq x_1 \leq 1.2,$$
$$0.55 \leq x_2 \leq 0.65,$$
$$1.35 \leq x_3 \leq 1.45.$$

Example 4 (Ref. [16])

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

Example 5

$$\begin{aligned} \min \quad & \max \left\{ \frac{\sum_{i=1}^N n_{1i}x_i + \bar{n}_1}{\sum_{i=1}^N d_{1i}x_i + \bar{d}_1}, \frac{\sum_{i=1}^N n_{2i}x_i + \bar{n}_2}{\sum_{i=1}^N d_{2i}x_i + \bar{d}_2}, \ldots, \frac{\sum_{i=1}^N n_{pi}x_i + \bar{n}_p}{\sum_{i=1}^N d_{pi}x_i + \bar{d}_p} \right\} \\ \text{s.t.} \quad & Ax \leq b, \\ & 0 \leq x_i \leq 3, \ i = 1, \ldots, N, \end{aligned}$$

where A is an $M \times N$ matrix, b is an M dimension vector, all elements of $n_{ji}, d_{ji}, j = 1, \ldots, p, i = 1, \ldots, N$, are randomly generated between 0 and 1; all elements of $\bar{n}_j, \bar{d}_j, j = 1, \ldots, p$, 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 Tables 2–3, the notations have been also used for column headers: p is the number of linear fractional function in the objective function; M represents the number of rows for A; N stands for the dimension of considered problem.

Example 6

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

Example 7

$$\begin{aligned}\min\quad \max\Bigg\{&\frac{3x_1+4x_2-x_3+0.5}{2x_1-x_2+x_3+0.5},\frac{3x_1-x_2+3x_3+0.5}{9x_1+5x_2-x_3+0.5},\\&\frac{4x_1-x_2+5x_3+0.5}{11x_1+6x_2-x_3},\frac{5x_1-x_2+6x_3+0.5}{12x_1+7x_2-x_3+0.9}\Bigg\}\\ \text{s.t.}\quad & x_1+x_2-x_3\le 1,\\ & -x_1+x_2-x_3\le -1,\\ & 12x_1+5x_2+12x_3\le 42,\\ & 12x_1+12x_2+7x_3\le 55,\\ & -6x_1+x_2+x_3\le -3,\\ & 1.0\le x_1\le 2.0,\\ & 0.50\le x_2\le 2.0,\\ & 0.50\le x_3\le 2.0.\end{aligned}$$

Example 8

$$\begin{aligned}\min\quad \max\Bigg\{&\frac{3x_1+4x_2-x_3+0.9}{2x_1-x_2+x_3+0.5},\frac{3x_1-x_2+3x_3+0.5}{9x_1+5x_2-x_3+0.5},\frac{4x_1-x_2+5x_3+0.5}{11x_1+6x_2-x_3+0.9},\\&\frac{5x_1-x_2+6x_3+0.5}{12x_1+7x_2-x_3+0.9},\frac{6x_1-x_2+7x_3+0.6}{11x_1+6x_2-x_3+0.9}\Bigg\}\\ \text{s.t.}\quad & 2x_1+x_2-x_3\le 2,\\ & -2x_1+x_2-2x_3\le -1,\\ & 11x_1+6x_2+12x_3\le 45,\\ & 11x_1+13x_2+6x_3\le 52,\\ & -7x_1+x_2+x_3\le -2,\\ & 1.0\le x_1\le 2.0,\\ & 0.35\le x_2\le 0.9,\\ & 1.0\le x_3\le 1.55.\end{aligned}$$

Example 9

$$\begin{aligned}\min\quad \max\Bigg\{&\frac{5x_1+4x_2-x_3+0.9}{3x_1-x_2+2x_3+0.5},\frac{3x_1-x_2+4x_3+0.5}{9x_1+3x_2-x_3+0.5},\frac{4x_1-x_2+6x_3+0.5}{12x_1+7x_2-x_3+0.9},\\&\frac{7x_1-x_2+7x_3+0.5}{11x_1+9x_2-x_3+0.9},\frac{7x_1-x_2+7x_3+0.7}{11x_1+7x_2-x_3+0.8}\Bigg\}\\ \text{s.t.}\quad & 2x_1+2x_2-x_3\le 3,\\ & -2x_1+x_2-3x_3\le -1,\\ & 11x_1+7x_2+12x_3\le 47,\end{aligned}$$

$$13x_1 + 13x_2 + 6x_3 \leq 56,$$
$$-6x_1 + 2x_2 + 3x_3 \leq -1,$$
$$1.0 \leq x_1 \leq 2.0,$$
$$0.35 \leq x_2 \leq 0.9,$$
$$1.0 \leq x_3 \leq 1.55.$$

From the experimental results in Tables 1–4, it is seen that the proposed algorithm has higher computational efficiency than Refs. [16,17], and can be used to globally solve the MLFP problems with large scale of fractional objective functions.

6. Concluding remarks

To solve the problem (MLFP), a global optimization algorithm is presented, which combines branch-and-bound scheme with the new linearization technique. The proposed algorithm is convergent to the global minimum through the successive refinement of linear relaxation of the feasible region and the subsequent solutions of a series of LRP problems. The main work of the algorithm involves solving ordinary linear programming problems that do not grow in size from iteration to iteration, and these problems can be efficiently solved by using the simplex method. Numerical results for several examples are given to illustrate the feasibility and effectiveness of the presented algorithm. It is hoped that the ideas and methods used to create the algorithm will offer useful tools for solving MLFP problem.

Acknowledgements

This paper is supported by the National Natural Science Foundation of China under Grant (61373174) and the Key Technology Projects of Henan Province (122102110038).

References

[1] I. Ahmad and Z. Husain, *Duality in nondifferentiable minimax fractional programming with generalized convexity*, Appl. Math. Comput. 176 (2006), pp. 545–551.
[2] C. Bajona-Xandri and J.E. Martinez-Legaz, *Lower subdifferentiability in minimax fractional programming*, Optimization 45 (1999), pp. 1–12.
[3] P. Balasubramaniam and S. Lakshmanan, *Delay-interval-dependent robust-stability criteria for neutral stochastic neural networks with polytopic and linear fractional uncertainties*, Int. J. Comput. Math. 88(10) (2011), pp. 2001–2015.
[4] I. Barrodale, *Best rational approximation and strict quasiconvexity*, SIAM J. Numer. Anal. 10 (1973), pp. 8–12.
[5] A.I. Barros and J.B.G. Frenk, *Generalized fractional programming and cutting plane algorithms*, J. Optim. Theory Appl. 87 (1995), pp. 103–120.
[6] Y. Benadada and J.A. Fedand, *Partial linearization for generalized fractional programming*, Z. Oper. Res. 32 (1988), pp. 101–106.
[7] J. Borde and J.P. Crouzeix, *Convergence of a dinkelbach-type algorithm in generalized fractional programming*, Z. Oper. Res. 31(1) (1987), pp. A31–A54.
[8] H.J. Chen, S. Schaible, and R.L. Sheu, *Generic algorithm for generalized fractional programming*, J. Optim. Theory Appl. 141 (2009), pp. 93–105.
[9] J.P. Crouzeix, J.A. Ferland, and S. Schaible, *An algorithm for generalized fractional programs*, J. Optim. Theory Appl. 47 (1985), pp. 135–149.
[10] F. Ding, *Coupled-least-squares identification for multivariable systems*, IET Control Theory Appl. 7(1) (2013), pp. 68–79.
[11] F. Ding, *Decomposition based fast least squares algorithm for output error systems*, Signal Process. 93 (2013), pp. 1235–1242.
[12] F. Ding, *Hierarchical multi-innovation stochastic gradient algorithm for hammerstein nonlinear system modeling*, Appl. Math. Model. 37 (2013), pp. 1694–1704.

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

[14] F. Ding and Y. Gu, *Performance analysis of the auxiliary model-based least-squares identification algorithm for one-step state-delay systems*, Int. J. Comput. Math. 89(15) (2012), pp. 2019–2028.

[15] 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(2) (2013), pp. 176–184.

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

[17] 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), pp. 806–809.

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

[19] Y. Gao and W.-D. Rong, *Optimality conditions and duality for a class of nondifferentiable multiobjective generalized fractional programming problems*, Appl. Math. Ser. B 23(3) (2008), pp. 331–344.

[20] M.H. Goedhart and J. Spronk, *Financial planning with fractional goals*, Eur. J. Oper. Res. 82 (1995), pp. 111–124.

[21] M. Gugat, *Prox-regularization methods for generalized fractional programming*, J. Optim. Theory Appl. 99(3) (1998), pp. 691–722.

[22] S.K. Gupta and D. Dangar, *On second-order duality for nondifferentiable minimax fractional programming*, J. Comput. Appl. Math. 255(1) (2014), pp. 878–886.

[23] Q. Hu, Y. Chen, and J. Jian, *Second-order duality for non-differentiable minimax fractional programming*, Int. J. Comput. Math. 89(1) (2012), pp. 11–16.

[24] Z. Husain, I. Ahmad, and Sarita Sharma, *Second order duality for minmax fractional programming*, Optim. Lett. 3 (2009), pp. 277–286.

[25] V. Jeyakumar, G.Y. Li, and S. Srisatkunarajah, *Strong duality for robust minimax fractional programming problems*, Eur. J. Oper. Res. 228 (2013), pp. 331–336.

[26] H. Jiao, *A branch and bound algorithm for globally solving a class of nonconvex programming problems*, Nonlinear Anal. Theory 70 (2009), pp. 1113–1123.

[27] H.-C. Lai and T.-Y. Huang, *Optimality conditions for nondifferentiable minimax fractional programming with complex variables*, J. Math. Anal. Appl. 359 (2009), pp. 229–239.

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

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

[30] X. Li, R. Ding, and L. Zhou, *Least-squares-based iterative identification algorithm for Hammerstein nonlinear systems with non-uniform sampling*, Int. J. Comput. Math. 90(7) (2013), pp. 1524–1534.

[31] J.Y. Lin and R.L. Sheu, *Modified dinkelbach-type algorithm for generalized fractional programs with infinitely many ratios*, J. Optim. Theory Appl. 126(2) (2005), pp. 323–343.

[32] Y.J. Liu and R. Ding, *Consistency of the extended gradient identification algorithm for multi-input multi-output systems with moving average noises*, Int. J. Comput. Math. 90(9) (2013), pp. 1840–1852.

[33] H. Liu, X. Li, and Y. Huang, *Trust-region method for box-constrained semismooth equations and its applications to complementary problems*, Int. J. Comput. Math. 89(17) (2012), pp. 2281–2306.

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

[35] A. Roubi, *Method of centers for generalized fractional programming*, J. Optim. Theory Appl. 107(1) (2000), pp. 123–143.

[36] M. Soleimani-Damaneh, *On fractional programming problems with absolute-value functions*, Int. J. Comput. Math. 88(4) (2011), pp. 661–664.

[37] I.M. Stancu-Minasian, *Fractional Programming: Theory, Methods and Applications*, Kluwer, Dordrecht, 1997.

[38] J.-J. Strodiot, J.-P. Crouzeix, J.A. Ferland, and V.H. Nguyen, *An inexact proximal point method for solving generalized fractional programs*, J. Global Optim. 42 (2008), pp. 121–138.

[39] W. Wang, J.H. Li, and R.F. Ding, *Maximum likelihood parameter estimation algorithm for controlled autoregressive models*, Int. J. Comput. Math. 88(16) (2011), pp. 3458–3467.

[40] C.-F. Wen, *Continuous-time generalized fractional programming problems. Part I: Basic theory*, J. Optim. Theory Appl. 157 (2013), pp. 365–399.

[41] C.-F. Wen, *Continuous-time generalized fractional programming problems, Part II: An interval-type computational procedure*, J. Optim. Theory Appl. 156 (2013), pp. 819–843.

[42] C.-F. Wen and H.-C. Wu, *Using the parametric approach to solve the continuous-time linear fractional max-min problems*, J. Global Optim. 54 (2012), pp. 129–153.

[43] X.J. Zheng and L. Cheng, *Minimax fractional programming under nonsmooth generalized (F,ρ,θ)-d-univexity*, J. Math. Anal. Appl. 328 (2007), pp. 676–689.

[44] J. Zhu, H. Liu, and B. Hao, *A new semismooth newton method for NCPs based on the penalized KK function*, Int. J. Comput. Math. 89(4) (2012), pp. 543–560.