Applied Mathematics and Computation 204 (2008) 281–287

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

A global optimization algorithm for linear fractional programming ☆

Wang Chun-Feng *, Shen Pei-Ping

Department of Mathematics, Henan Normal University, Construction Road 134, Xinxiang 453007, PR China

ARTICLE INFO

Keywords:
Global optimization
Linear relaxation
Branch and bound
Fractional programming
Sum-of-ratios

ABSTRACT

In this paper, we present an efficient branch and bound method for general linear fractional problem (GFP). First, by using a transformation technique, an equivalent problem (EP) of GFP is derived, then by exploiting structure of EP, a linear relaxation programming (LRP) of EP is obtained. To implement the algorithm, the main computation involve solving a sequence of linear programming problem, which can be solved efficiently. The proposed algorithm is convergent to the global maximum through the successive refinement of the solutions of a series of linear programming problems. Numerical experiments are reported to show the feasibility of our algorithm.

1. Introduction

In this paper, we consider general linear fractional problem (GFP) as the following form:

$$\mathrm{GFP}:\begin{cases} v=\max \quad g(x)=\sum_{i=1}^{p}\frac{\sum_{j=1}^{n}c_{ij}x_j+d_i}{\sum_{j=1}^{n}e_{ij}x_j+f_i}, \\ \quad\quad \text{s.t.} \quad Ax\leqslant b,\ x\geqslant 0, \end{cases}$$

where $A\in R^{m\times n},\ b\in R^m, c_{ij},\ d_i, e_{ij},\ f_i$ are all arbitrary real number, $\Lambda \triangleq \{x\in R^n|Ax\leqslant b,\ x\geqslant 0\}$ is bounded with $int\Lambda\neq\emptyset$, and $\sum_{j=1}^{n}e_{ij}x_j+f_i\neq 0$, for $\forall x\in\Lambda$, $i=1,\ldots,p,\ j=1,\ldots,n.$

Linear sum of ratios problem is a special class optimization among fractional programming, which has attracted the interest of researchers and practitioners for a number of years. First reason is that it frequently appears in application and many other nonlinear problems can be transformed into this form. Another reason is, from a research point view, these problems poses significant theoretical and computational difficulties, i.e., it is known to generally possess multiple local optima that are not globally optima. So it is necessary to put forward good method.

During the past years, various algorithms have been proposed for solving special cases of problem GFP, which are indented only for the sum of linear ratios problem with the assumption that $\sum_{j=1}^{n}c_{ij}x_j+d_i\geqslant 0$, and $\sum_{j=1}^{n}e_{ij}x_j+f_i>0$ for any $x\in\Lambda$. For instance, when Λ is a polyhedral, and $m=2$, an algorithm has been developed which use the parametric simplex method [1]. When Λ is a polyhedral, but $m\geqslant 2$, some algorithms which search iteratively the non-convex outcome space until a global optimal solution is found have been proposed [2,3]. In addition, under the assumption that $\sum_{j=1}^{n}c_{ij}x_j+d_i\geqslant 0$, and $\sum_{j=1}^{n}e_{ij}x_j+f_i\neq 0$, a branch and bound algorithm has been proposed [4].

The purpose of this paper is to present a new global optimization method for a more general linear fractional programming by solving a sequence of linear programming problem over partitioned subsets. The main feature of this algorithm, (1) In GFP, we only request $\sum_{j=1}^{n}e_{ij}x_j+f_i\neq 0$, then the model of this paper is more general than other paper considered. (2) The

☆ This paper is supported by The National Natural Science Foundation of China (No. 10671057).
* Corresponding author.
E-mail address: zzyhwcf@126.com (C.-F. Wang).

0096-3003/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2008.06.045

algorithm economized the required computations by conducting the branch and bound search in R^p, rather in R^n. (3) The linear relaxation of EP is obtained which is easier in computation than the method in [5] and does not generate new variables. (4) The proposed branch and bound algorithm is convergent to the global maximum through the successive refinement of the linear relaxation of feasible region of the objection function and constraint functions and the solutions of a series of LRP. (5) Numerical experiments are given to show the feasibility of our algorithm.

This paper is organized as follows. In Section 2, by using a transformation technique, problem EP is derived that is equivalent to problem GFP. The rectangular branching process, the upper and lower bounding process used in this approach are defined and studied in Section 3. The algorithm is introduced in Section 4, and its convergence is shown. Section 5 report some numerical results obtained by solving some examples. Finally, the summary of this paper is given.

2. Preliminaries

In this section, we first give an important theorem, which is the foundation of the global optimization algorithm.

Theorem 1. *Assume* $\sum_{j=1}^{n} e_{ij}x_j + f_i \neq 0$ *for* $\forall x \in \Lambda$, *then* $\sum_{j=1}^{n} e_{ij}x_j + f_i > 0$ *or* $\sum_{j=1}^{n} e_{ij}x_j + f_i < 0$.

Proof. By the intermediate value theorem, the conclusion is obvious.

For $\forall x \in \Lambda$, let $I_+ = \left\{i | \sum_{j=1}^{n} e_{ij}x_j + f_i > 0, i = 1, \ldots, p\right\}, I_- = \left\{i | \sum_{j=1}^{n} e_{ij}x_j + f_i < 0, i = 1, \ldots, p\right\}$. Then, we have

$$\sum_{i=1}^{p} \frac{\sum_{j=1}^{n} c_{ij}x_j + d_i}{\sum_{j=1}^{n} e_{ij}x_j + f_i} = \sum_{i \in I_+} \frac{\sum_{j=1}^{n} c_{ij}x_j + d_i}{\sum_{j=1}^{n} e_{ij}x_j + f_i} + \sum_{i \in I_-} \frac{-(\sum_{j=1}^{n} c_{ij}x_j + d_i)}{-(\sum_{j=1}^{n} e_{ij}x_j + f_i)}. \tag{1}$$

Obviously, in (1), denominators are all positive. Hence, in problem GFP, we can assume that $\sum_{j=1}^{n} e_{ij}x_j + f_i > 0$ is always holds. In addition, sice

$$\min \sum_{i=1}^{p} \frac{\sum_{j=1}^{n} c_{ij}x_j + d_i}{\sum_{j=1}^{n} e_{ij}x_j + f_i} = \min \sum_{i=1}^{p} \left(\frac{\sum_{j=1}^{n} c_{ij}x_j + d_i}{\sum_{j=1}^{n} e_{ij}x_j + f_i} + M_i \right) = \min \sum_{i=1}^{p} \frac{\sum_{j=1}^{n} c_{ij}x_j + d_i + M_i \left(\sum_{j=1}^{n} e_{ij}x_j + f_i \right)}{\sum_{j=1}^{n} e_{ij}x_j + f_i}, \tag{2}$$

where $M_i (i = 1, \ldots, p)$ is a positive number, if M_i large enough, $\sum_{j=1}^{n} c_{ij}x_j + d_i + M_i(\sum_{j=1}^{n} e_{ij}x_j + f_i) > 0$ can be satisfied. Therefore, in the following, without loss of generality, we can assume that $\sum_{j=1}^{n} c_{ij}x_j + d_i \geqslant 0$ and $\sum_{j=1}^{n} e_{ij}x_j + f_i > 0$ in GFP.

Next, we show how to convert problem GFP into an equivalent problem EP.

Let $\bar{l}_i = \min_{x \in \Lambda} \sum_{j=1}^{n} e_{ij}x_j + f_i, \bar{u}_i = \max_{x \in \Lambda} \sum_{j=1}^{n} e_{ij}x_j + f_i, i = 1, \ldots, p$. Define $H^0 = \{y \in R^p | l_i^0 \leqslant y_i \leqslant u_i^0, i = 1, \ldots, p\}$ with $l_i^0 = \frac{1}{\bar{u}_i}, u_i^0 = \frac{1}{\bar{l}_i}$, then problem GFP can be converted into an equivalent non-convex programming problem as follows:

$$\mathrm{EP}(H^0): \begin{cases} v(H^0) = \max \quad \varphi_0(x, y) = \sum_{i=1}^{p} y_i \left(\sum_{j=1}^{n} c_{ij}x_j + d_i \right), \\ \qquad \text{s.t.} \quad \varphi_i(x, y) = y_i \left(\sum_{j=1}^{n} e_{ij}x_j + f_i \right) \leqslant 1, \quad i = 1, \ldots, p, \\ \qquad\qquad x \in \Lambda, \ y \in H^0. \end{cases}$$

The key equivalence result for problem GFP and $\mathrm{EP}(H^0)$ is given by the following theorem.

Theorem 2. *If* $(x^*, y_1^*, \ldots, y_p^*)$ *is a global optimal solution for problem* $\mathrm{EP}(H^0)$, *then* x^* *is a global optimal solution for problem GFP. Converse, if* x^* *is a global optimal solution for problem GFP, then* $(x^*, y_1^*, \ldots, y_p^*)$ *is a global optimal solution for problem* $\mathrm{EP}(H^0)$, *where* $y_i^* = \sum_{j=1}^{n} e_{ij}x_j^* + f_i, \ i = 1, \ldots, p$.

Proof. The proof of this theorem follows easily from the definitions of problems GFP and $\mathrm{EP}(H^0)$, therefore, it is omitted. □

From Theorem 2, in order to globally solve problem GFP, we may globally solving problem $\mathrm{EP}(H^0)$ instead.

3. Basic operations

In this section, based on the above equivalent problem, a branch and bound algorithm is proposed for solving the global optimal solution of GFP. The main idea of this algorithm consists of three basic operations: successively refined partitioning of the feasible set, estimation of upper and lower bounds for the optimal value of the objective function. Next, we begin the establishment of algorithm with the basic operations needed in a branch and bound scheme.

3.1. Branching process

In this algorithm, the branching process is performed in R^p, rather in R^n, that iteratively subdivides the p-dimensional rectangle H^0 of problem $\mathrm{EP}(H^0)$ into smaller subrectangles that are also of dimension p. Let $H = \{y \in R^p | l_i \leqslant y_i \leqslant u_i, i = 1, \ldots, p\}$ denote the initial rectangle H^0 or subrectangle of it, the branching rule as follows:

(i) Let $\tau_i = \frac{1}{2}(l_i + u_i),\ i = 1, \ldots, p$,
(ii) Let

$$H^1 = \{y \in R^p | l_i \leqslant y_i \leqslant \tau_i, i = 1, \ldots, p\},$$
$$H^2 = \{y \in R^p | \tau_i \leqslant y_i \leqslant u_i, i = 1, \ldots, p\}.$$

It follows easily that this branching process is exhaustive, i.e. if $\{H^k\}$ denotes a nested subsequence of rectangles (i.e. $H^{k+1} \subseteq H^k$, for all k) formed by branching process, then there exists a unique point $y \in R^p$ such that $\bigcap_k H^k = \{y\}$.

3.2. Upper bound and lower bound

For each rectangle $H = \{y \in R^p | l_i \leqslant y_i \leqslant u_i,\ i = 1, \ldots, p\} (H \subseteq H^0)$ formed by the branching process, the upper bound process is used to compute an upper bound $UB(H)$ for the optimal value $v(H)$ of problem EP(H)

$$\text{EP}(H): \begin{cases} v(H) = \max \quad \varphi_0(x, y) = \sum_{i=1}^{p} y_i \left(\sum_{j=1}^{n} c_{ij} x_j + d_i \right), \\ \qquad\qquad \text{s.t.} \quad \varphi_i(x, y) = y_i \left(\sum_{j=1}^{n} e_{ij} x_j + f_i \right) \leqslant 1, \quad i = 1, \ldots, p, \\ \qquad\qquad\qquad x \in \Lambda,\ y \in H. \end{cases}$$

It will be seen from below, the upper bound $UB(H)$ can be found by solving an ordinary linear program.

In the following, for convenience of expression, let

$$\begin{aligned} T_i^+ &= \{j | c_{ij} > 0, \quad j = 1, \ldots, n\},\ i = 1, \ldots, p, \\ T_i^- &= \{j | c_{ij} < 0, \quad j = 1, \ldots, n\},\ i = 1, \ldots, p, \\ D^+ &= \{i | d_i > 0, \quad i = 1, \ldots, p\}, \\ D^- &= \{i | d_i < 0, \quad i = 1, \ldots, p\}, \\ E_i^+ &= \{j | e_{ij} > 0, \quad j = 1, \ldots, n\},\ i = 1, \ldots, p, \\ E_i^- &= \{j | e_{ij} < 0, \quad j = 1, \ldots, n\},\ i = 1, \ldots, p. \end{aligned}$$

First, consider objective function $\varphi_0(x, y)$, we have

$$\varphi_0(x, y) = \sum_{i=1}^{p} y_i \left(\sum_{j=1}^{n} c_{ij} x_j + d_i \right) \leqslant \sum_{i=1}^{p} \left(\sum_{j \in T_i^+} c_{ij} x_j u_i + \sum_{j \in T_i^-} c_{ij} x_j l_i \right) + \sum_{i \in D^+} d_i u_i + \sum_{i \in D^-} d_i l_i \triangleq \varphi_0^u(x).$$

Then, consider constraint function $\varphi_i(x, y),\ i = 1, \ldots, p$,

$$\varphi_i(x, y) = y_i \left(\sum_{j=1}^{n} e_{ij} x_j + f_i \right) \geqslant \sum_{j \in E_i^+} e_{ij} x_j l_i + \sum_{j \in E_i^-} e_{ij} x_j u_i + \beta_i \triangleq \varphi_i^l(x),$$

where

$$\beta_i = \begin{cases} f_i l_i, & \text{if } f_i \geqslant 0, \\ f_i u_i & \text{if } f_i < 0. \end{cases}$$

Based on the above discussion, we can construct a linear relaxation programming (LRP) as follows, which provides an upper bound for the optimal value $v(H)$ of problem EP(H).

$$\text{LRP}(H): \begin{cases} UB(H) = \max \quad \varphi_0^u(x), \\ \qquad\qquad\quad \text{s.t.} \quad \varphi_i^l(x) \leqslant 1, \quad i = 1, \ldots, p, \\ \qquad\qquad\qquad\quad x \in \Lambda. \end{cases}$$

Remark 1. Let $v[P]$ denotes the optimal value of the problem P, then, from the above discussion, the optimal values of LRP(H) and EP(H) satisfy $v[\text{LRP}(H)] \geqslant v[\text{EP}(H)]$ for $\forall H \subseteq H^0$.

Remark 2. Obviously, if $\overline{H} \subseteq H \subseteq H^0$, then $UB(\overline{H}) \leqslant UB(H)$.

Another basic operation is to determinate a lower bound for the optimal value $v(H^0)$ of problem EP(H^0). By the upper bound process, through solving LRP(H), we will have a optimal solution $\tilde{x}^*$. Let $\tilde{y}_i^* = \frac{1}{\sum_{j=1}^{n} e_{ij} \tilde{x}_j^* + f_i}$, obviously, $(\tilde{x}^*, \tilde{y}^*)$ is a feasible solution of problem EP(H^0), hence, $\varphi_0(\tilde{x}^*, \tilde{y}^*)$ provides a lower bound for the optimal value $v(H^0)$ of problem EP(H^0).

4. Algorithm and its convergence

Based upon the results and operations given in Section 3, the branch and bound algorithm for problem GFP may be stated as follows:

Branch and bound algorithm

Step 0 Choose $\epsilon \geqslant 0$. Let H^0 be denoted by

$$H^0 = \{y \in R^p | l_i^0 \leqslant y_i \leqslant u_i^0, \quad i = 1, \dots, p\},$$

Find an optimal solution x^0 and the optimal value $UB(H^0)$ for problem LRP(H^0). Set $UB_0 = UB(H^0),\ x^c = x^0$. Set

$$y_i^c = \frac{1}{\sum_{j=1}^n e_{ij} x_j^c + f_i}, \quad i \in \{1, 2, \dots, p\},\ LB_0 = \varphi_0(x^c, y^c).$$

If $UB_0 - LB_0 \leqslant \epsilon$, stop. (x^c, y^c) and x^c are global ϵ-optimal solutions for problems EP(H^0) and GFP, respectively. Otherwise, set $P_0 = \{H^0\},\ F = \emptyset,\ k = 1$, and go to Step 1.

Step 1 Set $LB_k = LB_{k-1}$. Subdivide H^{k-1} into two p-dimensional rectangles $H^{k,1},\ H^{k,2} \subseteq R^p$ via the branching rule. Set $F = F \bigcup \{H^{k-1}\}$.

Step 2 For $j = 1, 2$, compute $UB(H^{k,j})$ and, if $UB(H^{k,j}) \neq -\infty$, find an optimal solution $x^{k,j}$ for problem LRP($\widehat{H}$) with $\widehat{H} = H^{k,j}$. Set t = 0.

Step 3 Set $t = t + 1$. If $t > 2$, go to Step 5. Otherwise, continue.

Step 4 If $UB(H^{k,t}) \leqslant LB_k$, set $F = F \bigcup \{H^{k,t}\}$, and go to step 3. Otherwise, Set

$$y_i^{k,t} = \frac{1}{\sum_{j=1}^n e_{ij} x_j^{k,t} + f_i}, \quad i \in \{1, \dots, p\}.$$

Let

$$LB_k = \max\{LB_k, \varphi_0(x^{k,t}, y^{k,t})\}.$$

If

$$LB_k > \varphi_0(x^{k,t}, y^{k,t}),$$

go to Step 3. If

$$LB_k = \varphi_0(x^{k,t}, y^{k,t}),$$

set

$$x^c = x^{k,t}, \quad (x^c, y^c) = (x^{k,t}, y^{k,t}),$$

and set

$$F = F \bigcup \{H \in P_{k-1} | UB(H) \leqslant LB_k\},$$

and continue.

Step 5 Set $P_k = \{H | H \in (P_{k-1} \bigcup \{H^{k,1}, H^{k,2}\}),\ H \notin F\}$.

Step 6 Set $UB_k = \max\{UB(H) | H \in P_k\}$, and let $H^k \in P_k$ satisfy $UB_k = UB(H^k)$. If $UB_k - LB_k \leqslant \epsilon$, stop. (x^c, y^c) and x^c are global ϵ-optimal solutions for problems $EP(H^0)$ and GFP, respectively. Otherwise, set $k = k + 1$ and go to Step 1.

The convergence properties of the algorithm are given in the following theorem.

Theorem 3

(a) *If the algorithm is finite, then upon termination,* (x^c, y^c) *and* x^c *are global* ϵ*-optimal solutions for problems* EP(H^0) *and GFP, respectively.*

(b) *For each* $k \geqslant 0$, *let* x^k *denote the incumbent solution* x^c *for problem GFP at the end of Step k. If the algorithm is infinite, every accumulation point of which is a global optimal solution for problem GFP, and*

$$\lim_{k \to \infty} UB_k = \lim_{k \to \infty} LB_k = v.$$

Proof

(a) If the algorithm is finite, then it terminates in Step $k \geqslant 0$. Upon termination, since (x^c, y^c) is found by solving problem EP(H), for some $H \subseteq H^0$, for an optimal solution x^c and setting

$$y_i^c = \frac{1}{\sum_{j=1}^n e_{ij} x_j^c + f_i}, \quad i \in \{1, \ldots, p\},$$

x^c is a feasible solution for problem GFP, and (x^c, y^c) is a feasible solution for problem EP(H^0). Upon termination of the algorithm, $UB_k - LB_k \leqslant \epsilon$ is satisfied. From Step 0 and Step 1 and Step 4, this implies that $UB_k - \varphi_0(x^c, y^c) \leqslant \epsilon$. By the algorithm, it shows that

$$UB_k \geqslant v.$$

Since (x^c, y^c) is a feasible solution for problem $EP(H^0)$,

$$\varphi_0(x^c, y^c) \leqslant v.$$

Taken together, this implies that

$$v \leqslant UB_k \leqslant \varphi_0(x^c, y^c) + \epsilon \leqslant v + \epsilon.$$

Therefore,

$$v - \epsilon \leqslant \varphi_0(x^c, y^c) \leqslant v. \tag{3}$$

Since $y_i^c = \frac{1}{\sum_{j=1}^n e_{ij} x_j^c + f_i}, \quad i = 1, \ldots, p$, we have

$$g(x^c) = \varphi_0(x^c, y^c).$$

From (3), this implies that

$$v - \epsilon \leqslant g(x^c) \leqslant v,$$

and the proof of part (a) is complete.

(b) Suppose that the algorithm is infinite. Then it generates a sequence of incumbent solutions for problem EP(H^0), which we may denote by $\{(x^k, y^k)\}$. For each $k \geqslant 1$, (x^k, y^k) is found by solving problem EP(H^k), for some rectangle $H^k \subseteq H^0$, for an optimal solution $x^k \in \Lambda$, and setting $y_i^k = \frac{1}{\sum_{j=1}^n e_{ij} x_j^k + f_i}, \quad i \in \{1, \ldots, p\}$. Therefore, the sequence $\{x^k\}$ consists of feasible solutions for problem GFP. Let $\bar{x}$ be an accumulation point of $\{x^k\}$, and assume without loss of generality that

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

Then, since Λ is a compact set, $\bar{x} \in \Lambda$. Furthermore, since $\{x^k\}$ is infinite, we may assume that without loss of generality that, for each k, $H^{k+1} \subseteq H^k$. From Horst and Tuy [6], since the rectangles H^k, $k \geqslant 1$, are formed by rectangular bisection, this implies that, for some point $\bar{y} \in R^p$

$$\lim_{k \to \infty} H^k = \bigcap_k H^k = \{\bar{y}\}. \tag{4}$$

Let $\overline{H} = \{\bar{y}\}$ and, for each k, let H^k be given by

$$H^k = \{y \in R^p | l_i^k \leqslant y_i \leqslant u_i^k, \quad i = 1, \ldots, p\}.$$

Since $H^{k+1} \subset H^k \subset H^0$, for each k, by Remark 2 and Step 4, this implies that $\{UB(H^k)\}$ is a nonincreasing sequence of real numbers bounded below by v. Therefore, $\lim_{k \to \infty} UB(H^k)$ is a finite number and satisfies

$$\lim_{k \to \infty} UB(H^k) \geqslant v. \tag{5}$$

For each k, from Step 2, $UB(H^k)$ equal to the optimal value of the problem LRP(H^k) and x^k is an optimal solution for this problem. From (4), we have

$$\lim_{k \to \infty} l^k = \lim_{k \to \infty} u^k = \{\bar{y}\} = \overline{H}.$$

Since $\lim_{k \to \infty} x^k = \bar{x},\ l_i^k \leqslant \frac{1}{\sum_{j=1}^n e_{ij} x_j^k + f_i} \leqslant u_i^k$, and the continuity of $\sum_{j=1}^n e_{ij} x_j + f_i$,

$$\frac{1}{\sum_{j=1}^n e_{ij} \bar{x}_j + f_i} = \bar{y}_i, \quad i = 1, \ldots, p.$$

This implies that $(\bar{x}, \bar{y})$ is a feasible solution for problem EP(H^0). Therefore,

$$\varphi_0(\bar{x}, \bar{y}) \leqslant v.$$

Combing (5), we obtain that

$$\varphi_0(\bar{x}, \bar{y}) \leqslant v \leqslant \lim_{k \to \infty} UB(H^k). \tag{6}$$

Since

$$\lim_{k\to\infty} UB(H^k) = \sum_{i=1}^{p}\left(\sum_{j\in T_i^+} c_{ij}x_j u_i^k + \sum_{j\in T_i^-} c_{ij}x_j l_i^k\right) + \sum_{i\in D^+} d_i u_i^k + \sum_{i\in D^-} d_i l_i^k = \sum_{i=1}^{p} c_i \bar{y}_i\left(\sum_{j=1}^{n} e_{ij}\bar{x}_j + f_i\right) = \varphi_0(\bar{x},\bar{y}). \tag{7}$$

From (6) and (7), we have

$$\lim_{k\to\infty} UB(H^k) = v = \varphi_0(\bar{x},\bar{y}),$$

therefore, $(\bar{x},\bar{y})$ is a global optimal solution for problem EP(H^0). By Theorem 2, this implies that $\bar{x}$ is a global optimal solution for problem GFP.For each k, since x^k is the incumbent solution for problem GFP at the end of Step k,

$$LB_k = g(x^k), \quad \text{forall } k \geqslant 1.$$

By the continuity of g, we have

$$\lim_{k\to\infty} g(x^k) = g(\bar{x}).$$

Since $\bar{x}$ is a global optimal solution for problem GFP,

$$g(\bar{x}) = v.$$

Therefore, $\lim_{k\to\infty} LB_k = v$, and the proof is complete. □

5. Numerical experiment

To verify the performance of the proposed global optimization algorithm, some test problems are implemented on micro-computer, and the convergence tolerance set to ϵ = 1.0e − 4 in our experiment. The results are summarized in Tables 1 and 2.

Example 1

$$\begin{aligned}
\min\ & \frac{4x_1+3x_2+3x_3+50}{3x_2+3x_3+50} + \frac{3x_1+4x_3+50}{4x_1+4x_2+5x_3+50} + \frac{x_1+2x_2+4x_3+50}{x_1+5x_2+5x_3+50} + \frac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50},\\
\text{s.t.}\ & 2x_1+x_2+5x_3 \leqslant 10,\\
& x_1+6x_2+2x_3 \leqslant 10,\\
& 9x_1+7x_2+3x_3 \geqslant 10,\\
& x_1,\ x_2,\ x_3 \geqslant 0,
\end{aligned}$$

Example 2

$$\begin{aligned}
\min\ & \frac{3x_1+5x_2+3x_3+50}{3x_1+4x_2+5x_3+50} + \frac{3x_1+4x_2+50}{4x_1+3x_2+2x_3+50} + \frac{4x_1+2x_2+4x_3+50}{5x_1+4x_2+3x_3+50},\\
\text{s.t.}\ & 2x_1+x_2+5x_3 \leqslant 10,\\
& x_1+6x_2+2x_3 \leqslant 10,\\
& 9x_1+7x_2+3x_3 \geqslant 10,\\
& x_1,\ x_2,\ x_3 \geqslant 0,
\end{aligned}$$

Table 1
Computational results of test problems

	Example		
	1	2	3
Iter	58	80	32
$L_{\max}$	18	64	32
Time	2.968694	8.566259	1.089285

Table 2
Optimal solution and optimal value for test problems

Example	Optimal solution			Optimal value v
	x_1	x_2	x_3	
1	0	0.625	1.875	4.0000
2	0	3.3333	0	3.0029
3	3	4		5

Example 3

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

In Table 1, the notations has been used for column headers: Iter: number of algorithm iteration; $L_{\max}$: the maximal length of the enumeration tree; time: the execution time of the algorithm in second.

From Tables 1 and 2, it is seen that our algorithm can globally solve the problem GFP effectively. Test results also indicate that the proposed algorithm can be used successfully to globally solve the problem GFP.

6. Concluding remarks

In this paper, we present a branch and bound algorithm for solving general linear fractional problem GFP. To globally solve problem GFP, we first convert it into an equivalent problem EP(H^0), then, through using linearization method, we obtain a linear relaxation programming problem of EP(H^0). In the algorithm, First, the branching process takes place in the space R^p rather than in the space R^n. This economizes the computation required to solve problem GFP. This mainly due to the fact that the number of ratios in the objective function of problem GFP is smaller than the number of decision variables n in the problem. Second, the upper bounding subproblems are linear programming problems that are quite similar to one another. These characteristics of the algorithm offer computational advantages that can enhance the efficiencies of the algorithm.

It is hoped that in practice, the proposed algorithm and ideas used in this paper will offer valuable tools for solving general linear fractional programming.

Acknowledgements

The authors are grateful to the responsible editor and the anonymous referees for their valuable comments and suggestions, which have greatly improved the earlier version of this paper.

References

[1] H. Konno, Y. Yajima, T. Matsui, Parametric simplex algorithms for solving a special class of nonconvex minimization problem, Journal of Global Optimization 1 (1991) 65–81.
[2] H. Konno, H. Yamashita, Minimizing sums and products of linear fractional functions over a polytope, Naval Research Logistics 46 (1999) 583–596.
[3] J.E. Falk, S.W. Palocsay, Image space analysis of generalized fractional programs, Journal of Global Optimization 4 (1994) 63–88.
[4] Y. Ji, K.C. Zhang, S.J. Qu, A deterministic global optimization algorithm, Applied Mathematics and Computation 185 (2007) 382–387.
[5] H. Konno, M. Inori, Bond protfolio optimization by bilinear fractional programming, Journal of the Operations Research Society of Japan 32 (1989) 143–158.
[6] R. Horst, H. Tuy, Global Optimization: Deterministic Approaches, second ed., Springer-Verlag, Berlin, Germany, 1993.