A Global Optimization Algorithm for Solving Generalized Linear Fractional Programming

Yonghong Zhang, Zhaolong Li, and Lixia Liu

***Abstract*—In this paper, a global optimization algorithm for solving generalized linear fractional programming problem (GLFP) is presented. Firstly, a new linear relaxation technique is proposed; then, by a sequence of linear programming problems, the initial problem GLFP is solved. Furthermore, two pruning techniques are presented to improve the convergence speed of the proposed algorithm. The convergence of the proposed algorithm is proved, and some experiments are provided to show its feasibility and efficiency.**

***Index Terms*—Linear relaxation; Global optimization; Generalized linear fractional programming; Pruning technique; Branch and bound.**

I. Introduction

THE following generalized linear fractional programming problem (GLFP) is considered in this paper :

$$\text{GLFP}\left\{\begin{array}{lll} v= & \min & \Phi(x)=\sum\limits_{i=1}^{p}\delta_i\frac{n_i(x)}{d_i(x)} \\ & \text{s.t.} & Ax\le b, \end{array}\right.$$

where $p\ge 2,\ A=(a_{ij})_{m\times n},\ b=(b_i)_{m\times 1},\ \delta_i$ are arbitrary real numbers, $n_i(x)=\sum\limits_{j=1}^{n}c_{ij}x_j+d_i,\ d_i(x)=\sum\limits_{j=1}^{n}e_{ij}x_j+f_i$ are affine functions, $D=\{x\in R^n\mid Ax\le b\}$ is bounded with $intD\ne\emptyset$, and for $\forall x\in D$, $n_i(x)\ge 0,\ d_i(x)>0,\ i=1,\cdots,p.$

The problem GLFP is a special class of fractional programming, and it frequently appears in a wide variety of applications, including financial optimization [1], portfolio optimization [2], microeconomics [3], plant layout design [4], and so on. However, as pointed out in [5,6], the problem GLFP is NP-hard, that is, it generally posses multiple local optima that are not globally optima. So it is necessary to put forward good methods to solve problem GLFP.

For solving problem GLFP, many algorithms have been developed in the past few decades. For example, under the assumption that $n_i(x)\ge 0,\ d_i(x)>0$ and $p=1$, by using variable transformation, an efficient elementary simplex method was put forward[8]. When $p=2$, based on [7], a similar parametric elementary simplex method was proposed, which can be used to solve large scale problem [8]. When $p=3$, a heuristic algorithm was developed [9]. When $p\ge 3$, a twice linearization technique was presented by using the characteristics of exponential and logarithmic functions [10]. By using an equivalent transformation and a linearization technique, a branch and bound algorithm for solving a sum of linear ratios problem with coefficients was proposed [11]. Through solving an equivalent concave minimum problem of the original problem, a new branch and bound algorithm was proposed [12]. For the case that p is fixed, a fully polynomial time approximate scheme (FPTAS) was put forward[13]. Through using suitable transformation, a method was proposed , which has a potential to solve GLFP by some well known techniques [14]. By introducing a vector parameter, this problem is reduced to a general fractional programming with d.c. functions firstly, and then it is solved by using the generalized Dinkelbach's approach [15]. For the case that $n_i(x)\ge 0,\ d_i(x)\ne 0$, a branch and bound algorithm was developed by [16]. Under the assumption that $d_i(x)>0$, for solving the sum-of-linear-ratios problem with lower dimension, a linear relaxation algorithm was proposed [17]. Under the assumption that $d_i(x)\ne 0$, three global optimization algorithms were developed [18,19,20]. Recently, by using range division and linearization technique, a global optimization algorithm was designed [21].

For solving problem GLFP, this paper proposed a reliable and effective method. In this method (1) by utilizing the characteristics of the problem GLFP, we present a new linearization technique, which can be embedded within a branch and bound algorithm without increasing new variables and constraints; (2) two pruning techniques are presented, which can be used to improve the convergence speed of the proposed algorithm; (3) compared with [10,12,16], the model considered by this paper has a more general form; (4) numerical experiments show that the proposed algorithm is feasible, and the computational advantages are indicated.

This remainder of this study is organized as follows. In Section 2, the new linear relaxation technique is presented, which can be used to obtain the linear relaxation programming problem LRP for problem GLFP. In Section 3, two pruning techniques are presented to improve the convergence speed of the proposed algorithm. The global optimization algorithm is described, and its convergence is established in Section 4. To show the feasibility and efficiency of the proposed algorithm, numerical results are reported in Section 5.

Manuscript received August 16, 2019; revised January 28, 2020. This work was supported by the National Natural Science Foundation of China(11671122, 11801430); the Key Project of Henan Educational Committee(19A110021, 19A510014); the School Research Project(20180562).

Y.H. Zhang is with the Department of Mathematics, Henan Normal University, Xinxiang, 453007, PR China; Henan Engineering Laboratory for Big Data Statistical Analysis and Optimal Control, School of Mathematics and Information Sciences, Henan Normal University, Xinxiang, 453007, PR China email:zhangyonghong09@126.com

Z.L. Li is with the Audit Office, Henan Normal University, Xinxiang, 453007, PR China.

L.X. Liu is with School of Mathematics and Statistics, Xidian University, Xi'an 710126, PR China.

II. Linear relaxation programming (LRP) problem

Through solving $2n$ linear programming problems: $l_j^0=\min\limits_{x\in D}x_j,\ u_j^0=\max\limits_{x\in D}x_j,\ j=1,\cdots,n,$ and construct a rectangle $H^0=\{x\in R^n\mid l_j^0\le x_j\le u_j^0,\ j=1,\cdots,n\},$

the problem GLFP can be rewritten as the following form:

$$\text{GLFP}\begin{cases} v = & \min \quad \Phi(x) = \sum\limits_{i=1}^{p} \delta_i \frac{n_i(x)}{d_i(x)} \\ & \text{s.t.} \quad Ax \le b, \\ & \qquad x \in H^0. \end{cases}$$

To solve problem GLFP, the principal task of our algorithm is to construct lower bounds for this problem and its partitioned subproblems. Next, we explain how to obtain these two lower bounds are.

Let $H = \{x \mid l \le x \le u\}$ be the initial box H^0 or modified box as defined for some partitioned subproblem in a branch and bound scheme. Compute $\underline{\xi}_i = \sum\limits_{j=1}^{n} \min\{c_{ij}l_j, c_{ij}u_j\} + d_i, \overline{\xi}_i = \sum\limits_{j=1}^{n} \max\{c_{ij}l_j, c_{ij}u_j\} + d_i$, $\underline{\eta}_i = \sum\limits_{j=1}^{n} \min\{e_{ij}l_j, e_{ij}u_j\} + f_i$, $\overline{\eta}_i = \sum\limits_{j=1}^{n} \max\{e_{ij}l_j, e_{ij}u_j\} + f_i$, we have $\underline{\xi}_i \le n_i(x) \le \overline{\xi}_i$, $\underline{\eta}_i \le d_i(x) \le \overline{\eta}_i, \underline{\zeta}_i = \frac{1}{\overline{\eta}_i} \le \frac{1}{d_i(x)} \le \frac{1}{\underline{\eta}_i} = \overline{\zeta}_i$ $i = 1, \cdots, p$.

Consider the term $\frac{n_i(x)}{d_i(x)}$, $i = 1, \cdots, p$. Since $n_i(x) - \underline{\xi}_i \ge 0$, $\frac{1}{d_i(x)} - \underline{\zeta}_i \ge 0$, we have

$$(n_i(x) - \underline{\xi}_i)(\frac{1}{d_i(x)} - \underline{\zeta}_i) \ge 0,$$

that is

$$\frac{n_i(x)}{d_i(x)} - n_i(x)\underline{\zeta}_i - \underline{\xi}_i \frac{1}{d_i(x)} + \underline{\xi}_i\underline{\zeta}_i \ge 0.$$

Furthermore, we have

$$\frac{n_i(x)}{d_i(x)} \ge n_i(x)\underline{\zeta}_i + \underline{\xi}_i \frac{1}{d_i(x)} - \underline{\xi}_i\underline{\zeta}_i. \tag{1}$$

In addition, $n_i(x) - \underline{\xi}_i \ge 0$, $\frac{1}{d_i(x)} - \overline{\zeta}_i \le 0$, we have

$$(n_i(x) - \underline{\xi}_i)(\frac{1}{d_i(x)} - \overline{\zeta}_i) \le 0,$$

that is

$$\frac{n_i(x)}{d_i(x)} - n_i(x)\overline{\zeta}_i - \underline{\xi}_i \frac{1}{d_i(x)} + \underline{\xi}_i\overline{\zeta}_i \le 0.$$

Furthermore, we have

$$\frac{n_i(x)}{d_i(x)} \le n_i(x)\overline{\zeta}_i + \underline{\xi}_i \frac{1}{d_i(x)} - \underline{\xi}_i\overline{\zeta}_i. \tag{2}$$

Now, we consider that $\frac{1}{d_i(x)}$ over the interval $[\underline{\zeta}_i, \overline{\zeta}_i]$. Since the function $\frac{1}{d_i(x)}$ is a convex function over the interval $[\underline{\zeta}_i, \overline{\zeta}_i]$, we have

$$\frac{1}{d_i(x)} \ge -\frac{1}{\underline{\eta}_i\overline{\eta}_i} d_i(x) + \frac{2}{\sqrt{\underline{\eta}_i\overline{\eta}_i}}, \tag{3}$$

$$\frac{1}{d_i(x)} \le -\frac{1}{\underline{\eta}_i\overline{\eta}_i} d_i(x) + \frac{\underline{\eta}_i + \overline{\eta}_i}{\underline{\eta}_i\overline{\eta}_i}. \tag{4}$$

Since $\underline{\xi}_i \ge 0$, from (1) and (3), we have

$$\frac{n_i(x)}{d_i(x)} \ge n_i(x)\underline{\zeta}_i - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} d_i(x) + \frac{2\underline{\xi}_i}{\sqrt{\underline{\eta}_i\overline{\eta}_i}} - \underline{\xi}_i\underline{\zeta}_i. \tag{5}$$

From (2) and (4), we can obtain

$$\frac{n_i(x)}{d_i(x)} \le n_i(x)\overline{\zeta}_i - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} d_i(x) + \frac{(\underline{\eta}_i + \overline{\eta}_i)\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} - \underline{\xi}_i\overline{\zeta}_i. \tag{6}$$

From (5) and (6), we have the following relations:

$$\begin{aligned} \Phi(x) &= \sum_{i=1}^{p} \delta_i \tfrac{n_i(x)}{d_i(x)} = \sum_{\delta_i>0} \delta_i \tfrac{n_i(x)}{d_i(x)} + \sum_{\delta_i<0} \delta_i \tfrac{n_i(x)}{d_i(x)} \\ &\ge \sum_{\delta_i>0} \delta_i[n_i(x)\underline{\zeta}_i - \tfrac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} d_i(x) + \tfrac{2\underline{\xi}_i}{\sqrt{\underline{\eta}_i\overline{\eta}_i}} - \underline{\xi}_i\underline{\zeta}_i] \\ &\quad + \sum_{\delta_i<0} \delta_i[n_i(x)\overline{\zeta}_i - \tfrac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} d_i(x) + \tfrac{(\underline{\eta}_i+\overline{\eta}_i)\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} \\ &\quad - \underline{\xi}_i\overline{\zeta}_i] = \Phi^l(x), \end{aligned}$$

$$\begin{aligned} \Phi(x) &= \sum_{i=1}^{p} \delta_i \tfrac{n_i(x)}{d_i(x)} = \sum_{\delta_i>0} \delta_i \tfrac{n_i(x)}{d_i(x)} + \sum_{\delta_i<0} \delta_i \tfrac{n_i(x)}{d_i(x)} \\ &\le \sum_{\delta_i>0} \delta_i[n_i(x)\overline{\zeta}_i - \tfrac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} d_i(x) + \tfrac{(\underline{\eta}_i+\overline{\eta}_i)\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} - \underline{\xi}_i\overline{\zeta}_i] \\ &\quad + \sum_{\delta_i<0} \delta_i[n_i(x)\underline{\zeta}_i - \tfrac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} d_i(x) + \tfrac{2\underline{\xi}_i}{\sqrt{\underline{\eta}_i\overline{\eta}_i}} - \underline{\xi}_i\underline{\zeta}_i] \\ &= \Phi^u(x). \end{aligned}$$

Based on the above discussion, the linear relaxation programming problem LRP can be established as follows, which provides a lower bound for the optimal value of problem GLFP over H:

$$\text{LRP}\begin{cases} \min & \Phi^l(x) \\ \text{s.t.} & Ax \le b, \\ & x \in H. \end{cases}$$

Theorem 1. Consider the functions $\Phi^l(x)$, $\Phi(x)$ and $\Phi^u(x)$. For all $x \in H$, let $\Delta x = u - l$, then, we have $\lim\limits_{\Delta x \to 0}(\Phi(x) - \Phi^l(x)) = \lim\limits_{\Delta x \to 0}(\Phi^u(x) - \Phi(x)) \to 0$.

Proof. We first prove $\lim\limits_{\Delta x \to 0}(\Phi(x) - \Phi^l(x)) \to 0$. By the definitions $\Phi(x)$ and $\Phi^l(x)$, we have

$$\begin{aligned} &\mid \Phi(x) - \Phi^l(x) \mid \\ &= \mid \sum_{\delta_i>0} \delta_i(\tfrac{n_i(x)}{d_i(x)} - (n_i(x)\underline{\zeta}_i - \tfrac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} d_i(x) + \tfrac{2\underline{\xi}_i}{\sqrt{\underline{\eta}_i\overline{\eta}_i}} - \underline{\xi}_i\underline{\zeta}_i)) \\ &+ \sum_{\delta_i<0} -\delta_i((n_i(x)\overline{\zeta}_i - \tfrac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} d_i(x) + \tfrac{(\underline{\eta}_i+\overline{\eta}_i)\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} - \underline{\xi}_i\overline{\zeta}_i) - \tfrac{n_i(x)}{d_i(x)}) \mid \\ &\le \sum_{\delta_i>0} \delta_i \mid \tfrac{n_i(x)}{d_i(x)} - (n_i(x)\underline{\zeta}_i - \tfrac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} d_i(x) + \tfrac{2\underline{\xi}_i}{\sqrt{\underline{\eta}_i\overline{\eta}_i}} - \underline{\xi}_i\underline{\zeta}_i) \mid \\ &+ \sum_{\delta_i<0} \mid \delta_i \mid\mid (n_i(x)\overline{\zeta}_i - \tfrac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} d_i(x) + \tfrac{(\underline{\eta}_i+\overline{\eta}_i)\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} - \underline{\xi}_i\overline{\zeta}_i) - \tfrac{n_i(x)}{d_i(x)} \mid \\ &= \sum_{\delta_i>0} \delta_i\Delta_{i,1} + \sum_{\delta_i<0} \mid \delta_i \mid \Delta_{i,2}, \end{aligned} \tag{7}$$

where $\Delta_{i,1} = \mid \frac{n_i(x)}{d_i(x)} - (n_i(x)\underline{\zeta}_i - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} d_i(x) + \frac{2\underline{\xi}_i}{\sqrt{\underline{\eta}_i\overline{\eta}_i}} - \underline{\xi}_i\underline{\zeta}_i) \mid$, $\Delta_{i,2} = \mid (n_i(x)\overline{\zeta}_i - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} d_i(x) + \frac{(\underline{\eta}_i+\overline{\eta}_i)\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} - \underline{\xi}_i\overline{\zeta}_i) - \frac{n_i(x)}{d_i(x)} \mid$. To prove $\lim\limits_{\Delta x \to 0}(\Phi(x) - \Phi^l(x)) \to 0$, we only need to prove $\Delta_{i,1} \to 0$ and $\Delta_{i,2} \to 0$ as $\Delta x \to 0$.

Now, consider $\Delta_{i,1}$, we have

$$\begin{aligned}\Delta_{i,1} =& \mid \frac{n_i(x)}{d_i(x)} - (n_i(x)\underline{\zeta}_i - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i}d_i(x) + \frac{2\underline{\xi}_i}{\sqrt{\underline{\eta}_i\overline{\eta}_i}} - \underline{\xi}_i\underline{\zeta}_i) \mid \\ =& \mid \frac{n_i(x)}{d_i(x)} - (n_i(x)\underline{\zeta}_i + \underline{\xi}_i\frac{1}{d_i(x)} - \underline{\xi}_i\underline{\zeta}_i) + (n_i(x)\underline{\zeta}_i + \underline{\xi}_i\frac{1}{d_i(x)} \\ & -\underline{\xi}_i\underline{\zeta}_i) - (n_i(x)\underline{\zeta}_i - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i}d_i(x) + \frac{2\underline{\xi}_i}{\sqrt{\underline{\eta}_i\overline{\eta}_i}} - \underline{\xi}_i\underline{\zeta}_i) \mid \\ \le& \mid \frac{n_i(x)}{d_i(x)} - (n_i(x)\underline{\zeta}_i + \underline{\xi}_i\frac{1}{d_i(x)} - \underline{\xi}_i\underline{\zeta}_i) \mid \\ & + \mid (n_i(x)\underline{\zeta}_i + \underline{\xi}_i\frac{1}{d_i(x)} - \underline{\xi}_i\underline{\zeta}_i) - (n_i(x)\underline{\zeta}_i - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i}d_i(x) \\ & + \frac{2\underline{\xi}_i}{\sqrt{\underline{\eta}_i\overline{\eta}_i}} - \underline{\xi}_i\underline{\zeta}_i) \mid \\ =& \mid (n_i(x) - \underline{\xi}_i)(\frac{1}{d_i(x)} - \underline{\zeta}_i) \mid + \mid \underline{\xi}_i \mid\mid \frac{1}{d_i(x)} + \frac{1}{\underline{\eta}_i\overline{\eta}_i}d_i(x) \\ & - \frac{2}{\sqrt{\underline{\eta}_i\overline{\eta}_i}} \mid . \end{aligned} \tag{8}$$

Since $\frac{1}{d_i(x)} + \frac{1}{\underline{\eta}_i\overline{\eta}_i}d_i(x) - \frac{2}{\sqrt{\underline{\eta}_i\overline{\eta}_i}}$ is convex function about $d_i(x)$ over $[\underline{\eta}_i, \overline{\eta}_i]$, it can obtain the maximum $\frac{(\sqrt{\overline{\eta}_i}-\sqrt{\underline{\eta}_i})^2}{\underline{\eta}_i\overline{\eta}_i}$ at the point $\underline{\eta}_i$ or $\overline{\eta}_i$. From (8), we have

$$\Delta_{i,1} \le \mid \overline{\xi}_i - \underline{\xi}_i \mid\mid \overline{\zeta}_i - \underline{\zeta}_i \mid + \mid \underline{\xi}_i \mid\mid \frac{(\sqrt{\overline{\eta}_i} - \sqrt{\underline{\eta}_i})^2}{\underline{\eta}_i\overline{\eta}_i} \mid . \tag{9}$$

By the definitions of $\underline{\eta}_i$, $\overline{\eta}_i$, $\underline{\xi}_i$, $\overline{\xi}_i$, $\underline{\zeta}_i$ and $\overline{\zeta}_i$, we know that, $\overline{\eta}_i - \underline{\eta}_i \to 0$, $\overline{\xi}_i - \underline{\xi}_i \to 0$ and $\overline{\zeta}_i - \underline{\zeta}_i \to 0$ as $\Delta x \to 0$. From (9), we have $\Delta_{i,1} \to 0$ as $\Delta x \to 0$.

Next, consider $\Delta_{i,2}$, we have

$$\begin{aligned}\Delta_{i,2} =& \mid (n_i(x)\overline{\zeta}_i - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i}d_i(x) + \frac{(\underline{\eta}_i+\overline{\eta}_i)\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} - \underline{\xi}_i\overline{\zeta}_i) - \frac{n_i(x)}{d_i(x)} \mid \\ =& \mid (n_i(x)\overline{\zeta}_i - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i}d_i(x) + \frac{(\underline{\eta}_i+\overline{\eta}_i)\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} - \underline{\xi}_i\overline{\zeta}_i) \\ & -(n_i(x)\overline{\zeta}_i + \underline{\xi}_i\frac{1}{d_i(x)} - \underline{\xi}_i\overline{\zeta}_i) + (n_i(x)\overline{\zeta}_i + \underline{\xi}_i\frac{1}{d_i(x)} - \underline{\xi}_i\overline{\zeta}_i) \\ & - \frac{n_i(x)}{d_i(x)} \mid \\ \le& \mid \underline{\xi}_i \mid\mid -\frac{1}{\underline{\eta}_i\overline{\eta}_i}d_i(x) + \frac{\underline{\eta}_i+\overline{\eta}_i}{\underline{\eta}_i\overline{\eta}_i} - \underline{\xi}_i\frac{1}{d_i(x)} \mid \\ & + \mid (n_i(x) - \underline{\xi}_i)(\overline{\zeta}_i - \frac{1}{d_i(x)}) \mid . \end{aligned} \tag{10}$$

Since $-\frac{1}{\underline{\eta}_i\overline{\eta}_i}d_i(x) + \frac{\underline{\eta}_i+\overline{\eta}_i}{\underline{\eta}_i\overline{\eta}_i} - \underline{\xi}_i\frac{1}{d_i(x)}$ is a concave function about $d_i(x)$ over $[\underline{\eta}_i, \overline{\eta}_i]$, it has the maximum $\frac{(\sqrt{\overline{\eta}_i}-\sqrt{\underline{\eta}_i})^2}{\underline{\eta}_i\overline{\eta}_i}$ at the point $\sqrt{\underline{\eta}_i\overline{\eta}_i}$. By (8), we have

$$\Delta_{i,2} \le \mid \overline{\xi}_i - \underline{\xi}_i \mid\mid \overline{\zeta}_i - \underline{\zeta}_i \mid + \mid \underline{\xi}_i \mid\mid \frac{(\sqrt{\overline{\eta}_i} - \sqrt{\underline{\eta}_i})^2}{\underline{\eta}_i\overline{\eta}_i} \mid . \tag{11}$$

By the definitions of $\underline{\eta}_i$, $\overline{\eta}_i$, $\underline{\xi}_i$, $\overline{\xi}_i$, $\underline{\zeta}_i$ and $\overline{\zeta}_i$, we know that, $\overline{\eta}_i - \underline{\eta}_i \to 0$, $\overline{\xi} - \underline{\xi}_i \to 0$ and $\overline{\zeta}_i - \underline{\zeta}_i \to 0$ as $\Delta x \to 0$. From (1), we have $\Delta_{i,2} \to 0$ as $\Delta x \to 0$.

From (7), it follows that $\lim_{\Delta x \to 0}(\Phi(x) - \Phi^l(x)) \to 0$.

Similarly, we can prove $\lim_{\Delta x \to 0}(\Phi^u(x) - \Phi(x)) = 0$, and the proof is complete.

Theorem 1 implies that $\Phi^l(x)$ and $\Phi^u(x)$ will approximate the function $\Phi(x)$ as $\Delta x \to 0$.

III. Pruning technique

To improve the convergence speed of the proposed algorithm, two pruning techniques is proposed in this section. By using these techniques, the region in which the global optimal solution of problem GLFP does not exist can be pruned.

For $i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, n$, $k = 1, \cdots, n$, introduce the following notations:

$$\begin{aligned} \alpha_j =& \sum_{\delta_i>0}\delta_i(\underline{\zeta}_i c_{ij} - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i}e_{ij}) + \sum_{\delta_i<0}\delta_i(\overline{\zeta}_i c_{ij} - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i}e_{ij}), \\ \Lambda_1 =& \sum_{\delta_i>0}\delta_i(\underline{\zeta}_i d_i - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i}f_i + \frac{2\underline{\xi}_i}{\sqrt{\underline{\eta}_i\overline{\eta}_i}} - \underline{\xi}_i\underline{\zeta}_i) \\ & + \sum_{\delta_i<0}\delta_i(\overline{\zeta}_i d_i - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i}f_i + \frac{(\underline{\eta}_i+\overline{\eta}_i)\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} - \underline{\xi}_i\overline{\zeta}_i), \\ \gamma_k =& UB - \sum_{j=1,j\neq k}^{n}\min\{\alpha_j l_j, \alpha_j u_j\} - \Lambda_1, \\ \beta_j =& \sum_{\delta_i>0}\delta_i(\overline{\zeta}_i c_{ij} - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i}e_{ij}) + \sum_{\delta_i<0}\delta_i(\underline{\zeta}_i c_{ij} - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i}e_{ij}), \\ \Lambda_2 =& \sum_{\delta_i>0}\delta_i(\overline{\zeta}_i d_i - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i}f_i + \frac{(\underline{\eta}_i+\overline{\eta}_i)\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i} - \underline{\xi}_i\overline{\zeta}_i) \\ & + \sum_{\delta_i<0}\delta_i(\underline{\zeta}_i d_i - \frac{\underline{\xi}_i}{\underline{\eta}_i\overline{\eta}_i}f_i + \frac{2\underline{\xi}_i}{\sqrt{\underline{\eta}_i\overline{\eta}_i}} - \underline{\xi}_i\underline{\zeta}_i), \\ \rho_k =& LB - \sum_{j=1,j\neq k}^{n}\max\{\beta_j l_j, \beta_j u_j\} - \Lambda_2, \end{aligned}$$

and assume that UB and LB are the current known upper bound and lower bound of the optimal value v of the problem GLFP, respectively.

Theorem 2. For any subrectangle $H \subseteq H^0$ with $H_j = [l_j, u_j]$, if there exists some index $k \in \{1, 2, \cdots, n\}$ such that $\alpha_k > 0$ and $\gamma_k < \alpha_k u_k$, then there is no globally optimal solution of problem GLFP over H^1; if $\alpha_k < 0$ and $\gamma_k < \alpha_k l_k$, for some k, then there is no globally optimal solution of problem GLFP over H^2, where

$$H^1 = (H_j^1)_{n\times 1} \subseteq H, \text{ with } H_j^1 = \begin{cases} H_j, & j \neq k, \\ (\frac{\gamma_k}{\alpha_k}, u_k] \bigcap H_k, & j = k, \end{cases}$$

$$H^2 = (H_j^2)_{n\times 1} \subseteq H, \text{ with } H_j^2 = \begin{cases} H_j, & j \neq k, \\ [l_k, \frac{\gamma_k}{\alpha_k}) \bigcap H_k, & j = k. \end{cases}$$

Proof. For the kth component x_k of x, since $x_k \in (\frac{\gamma_k}{\alpha_k}, u_k]$, we have

$$\frac{\gamma_k}{\alpha_k} < x_k \le u_k.$$

Furthermore, since $\alpha_k > 0$, we have $\gamma_k < \alpha_k x_k$. For all $x \in H^1$, by the above inequality and the definition of γ_k, it implies that

$$UB - \sum_{j=1,j\neq k}^{n}\min\{\alpha_j l_j, \alpha_j u_j\} - \Lambda_1 < \alpha_k x_k,$$

i.e.

$$\begin{aligned} UB &< \sum_{j=1,j\neq k}^{n}\min\{\alpha_j l_j, \alpha_j u_j\} + \alpha_k x_k + \Lambda_1 \\ &\le \sum_{j=1}^{n}\alpha_j x_j + \Lambda_1 = \Phi^l(x). \end{aligned}$$

Thus, for all $x \in H^1$, we have $\Phi(x) \ge \Phi^l(x) > UB \ge v$, that is, for all $x \in H^1$, $\Phi(x)$ is always greater than the optimal value v of the problem GLFP. Therefore, there can not exist globally optimal solution of problem GLFP over H^1.

Similarly, for all $x \in H^2$, if there exists some k such that $\alpha_k < 0$ and $\gamma_k < \alpha_k l_k$, it can be derived that there is no globally optimal solution of problem GLFP over H^2.

Theorem 3. For any subrectangle $H \subseteq H^0$ with $H_j = [l_j, u_j]$, if there exists some index $k \in \{1, 2, \cdots, n\}$ such that $\beta_k > 0$ and $\rho_k > \beta_k l_k$, then there is no globally optimal solution of problem GLFP over H^3; if $\beta_k < 0$ and $\rho_k > \beta_k u_k$, for some k, then there is no globally optimal solution of problem GLFP over H^4, where

$$H^3 = (H_j^3)_{n\times 1} \subseteq H, \text{ with } H_j^3 = \begin{cases} H_j, & j \neq k, \\ [l_k, \frac{\rho_k}{\beta_k}) \bigcap H_k, & j = k, \end{cases}$$

$$H^4 = (H_j^4)_{n\times 1} \subseteq H, \text{ with } H_j^4 = \begin{cases} H_j, & j \neq k, \\ (\frac{\rho_k}{\beta_k}, u_k] \bigcap H_k, & j = k. \end{cases}$$

Proof. We first show that for all $x \in H^3$, $\Phi(x) < LB$. Consider the kth component x_k of x. By the assumption and the definitions of β_k and ρ_k, we have

$$l_k \leq x_k < \frac{\rho_k}{\beta_k}.$$

Note that $\beta_k > 0$, we have $\rho_k > \beta_k x_k$. For all $x \in H^3$, by the above inequality and the definition of ρ_k, it implies that

$$\begin{aligned} LB &> \sum_{j=1, j\neq k}^{n} \max\{\beta_j l_j, \beta_j u_j\} + \beta_k x_k + \Lambda_2 \\ &\geq \sum_{j=1}^{n} \beta_j x_j + \Lambda_2 = \Phi^u(x) \geq \Phi(x). \end{aligned}$$

Thus, for all $x \in H^3$, we have $v \geq LB > \Phi(x)$. Therefore, there can not exist globally optimal solution of problem GLFP over H^3.

Similarly, for all $x \in H^4$, if there exists some k such that $\beta_k < 0$ and $\rho_k > \beta_k u_k$, it can be derived that there is no globally optimal solution of problem GLFP over H^4.

IV. Algorithm and its convergence

In this section, based on the former results, we present the branch and bound algorithm, and prove its convergence.

A. Branching rule

In a branch and bound algorithm, the branching process plays an important role, which can create a more refined partition that cannot yet be excluded from further consideration in searching for a global optimal solution for problem GLFP. In this paper, a simple and standard bisection rule is chosen to partition rectangle, which is sufficient to ensure convergence since it drives the intervals shrinking to a singleton for all the variables along any infinite branch of the branch and bound tree.

For a rectangle $H = \{x \in R^n \mid l_j \leq x_j \leq u_j,\ j = 1, \cdots, n\} \subseteq H^0$ to be partitioned, the branching rule is as follows:

(i) let $k = \text{argmax}\{u_j - l_j \mid j = 1, \cdots, n\}$;

(ii) let $\pi_k = (l_k + u_k)/2$;

(iii) let

$$\begin{aligned} H^1 &= \{x \in R^n \mid l_j \leq x_j \leq u_j,\ j \neq k, l_k \leq x_k \leq \pi_k\}, \\ H^2 &= \{x \in R^n \mid l_j \leq x_j \leq u_j,\ j \neq k, \pi_k \leq x_k \leq u_k\}. \end{aligned}$$

Through using this branching rule, the rectangle H is partitioned into two subrectangles H^1 and H^2. Obviously, we have $H = H^1 \bigcup H^2$ and $intH^1 \bigcap intH^2 = \emptyset$.

B. Branch and bound algorithm

Let $LB(H^k)$ be the optimal function value of LRP over the subrectangle $H = H^k$, and $x^k = x(H^k)$ be an element of the corresponding argmin. The basic steps of the proposed algorithm are summarized as follows.

Algorithm statement

Step 1. Choose $\epsilon \geq 0$. Find an optimal solution $x^0 = x(H^0)$ and the optimal value $LB(H^0)$ for problem LRP with $H = H^0$. Set $LB_0 = LB(H^0)$, and $UB_0 = \Phi(x^0)$. If $UB_0 - LB_0 \leq \epsilon$, then stop: x^0 is an ϵ-optimal solution of problem GLFP. Otherwise, set $Q_0 = \{H^0\},\ k = 1$, and go to Step 2.

Step 2. Set $UB_k = UB_{k-1}$. Subdivide H^{k-1} into two subrectangles via the branching rule, and denote the set of new partition rectangles as $\overline{H}^k$.

Step 3. For each new rectangle $H \in \overline{H}^k$, utilizing the pruning techniques of Theorems 2 and 3 to prune rectangle H. For $i = 1, \cdots, m$, if there exists some i such that $\sum_{j=1}^{n} \min\{a_{ij} l_j, a_{ij} u_j\} \geq b_i$ over rectangle H, then remove the rectangle H from $\overline{H}^k$, i.e. $\overline{H}^k = \overline{H}^k \setminus H$.

Step 4. If $\overline{H}^k \neq \emptyset$, solve LRP to obtain $LB(H)$ and $x(H)$ for each $H \in \overline{H}^k$. If $LB(H) > UB_k$, set $\overline{H}^k = \overline{H}^k \setminus H$. Otherwise, let $UB_k = \min\{UB_k, \Phi(x(H))\}$. If $UB_k = \Phi(x(H))$, set $x^k = x(H)$.

Step 5. Set

$$Q_k = \{Q_{k-1} \setminus H^{k-1}\} \bigcup \overline{H}^k.$$

Step 6. Set $LB_k = \min\{LB(H) \mid H \in Q_k\}$. Let H^k be the subrectangle which satisfies that $LB_k = LB(H^k)$. If $UB_k - LB_k \leq \epsilon$, then stop: x^k is a global ϵ-optimal solution of problem GLFP. Otherwise, set $k = k + 1$, and go to Step 2.

C. Convergence analysis

In this subsection, we give the global convergence properties of the above algorithm.

Theorem 4. The above algorithm either terminates finitely with a globally ϵ-optimal solution, or generates an infinite sequence $\{x^k\}$ of iteration such that along any infinite branch of the branch and bound tree, which any accumulation point is a globally optimal solution of problem GLFP.

Proof. When the algorithm terminates finitely, that is, it terminates at some step $k \geq 0$. Upon termination, it follows that

$$UB_k - LB_k \leq \epsilon.$$

From Step 1 and Step 6 in the algorithm, a feasible solution x^k for the problem GLFP can be found, and the following relation holds

$$\Phi(x^k) - LB_k \leq \epsilon.$$

By Section 2, we have

$$LB_k \leq v.$$

Since x^k is a feasible solution of problem GLFP, $\Phi(x^k) \geq v$. Taken together above, it implies that

$$v \leq \Phi(x^k) \leq LB_k + \epsilon \leq v + \epsilon,$$

and so x^k is a global ϵ-optimal solution to the problem GLFP over H^0 in the sense that

$$v \leq \Phi(x^k) \leq v + \epsilon.$$

When the algorithm terminates infinitely, then an infinite sequence $\{x^k\}$ will be generated. Since the feasible region of GLFP is bounded, the sequence $\{x^k\}$ must be has a convergence subsequence. Without loss of generality, set $\lim_{k\to\infty} x^k = x^*$. By the algorithm, we have

$$\lim_{k\to\infty} LB_k \leq v.$$

Since x^* is a feasible solution of problem GLFP, $v \leq \Phi(x^*)$. Taken together, we have

$$\lim_{k\to\infty} LB_k \leq v \leq \Phi(x^*).$$

On the other hand, by the algorithm and the continuity of $\Phi^l(x)$, we have

$$\lim_{k\to\infty} LB_k = \lim_{k\to\infty} \Phi^l(x^k) = \Phi^l(x^*).$$

From Theorem 1, it follows that

$$\Phi(x^*) = \Phi^l(x^*).$$

Therefore, we have $v = \Phi(x^*)$, that is x^* is a global optimal solution of problem GLFP.

V. Numerical experiments

In this section, to verify the performance of the proposed algorithm, some numerical experiments are reported, and compared with several latest algorithms[17,10,11,22,23]. The algorithm is implemented by Matlab 7.1, and all test problems are carried out on a Pentium IV (3.06 GHZ) microcomputer. The simplex method is applied to solve the linear relaxation programming problems.

In Table I, the results of problems 1-6 are summarized, where the following notations have been used in row headers: ϵ: convergence error; Iter: number of algorithm iterations.

For Examples 1-6, we also used two algorithms to solve them, which are the algorithm (named Algorithm 1) proposed by this paper and the algorithm proposed by this paper but without using pruning techniques(named Algorithm 2), respectively. For this test, ϵ is set to $1e-6$. The comparison results are given in Table II. In Table II, $Time$ denotes execution time in seconds.

Table III summarizes our computational results of Example 7. For this test problem, ϵ is set to $1e-2$. In Table 2, $\mathrm{Ave.Iter}$ represents the average number of iterations; $\mathrm{Ave.Time}$ stands for the average CPU time of the algorithm in seconds, which are obtained by randomly running our algorithm for 10 test problems.

Example 1[18]

$$\begin{array}{ll} \max & 0.9 \times \dfrac{-x_1+2x_2+2}{3x_1-4x_2+5} + (-0.1) \times \dfrac{4x_1-3x_2+4}{-2x_1+x_2+3} \\ \text{s.t.} & x_1+x_2 \leq 1.5, \\ & x_1 - x_2 \leq 0, \\ & 0 \leq x_1 \leq 1,\ 0 \leq x_2 \leq 1. \end{array}$$

Example 2[18,10,23]

$$\begin{array}{ll} \max & \frac{4x_1+3x_2+3x_3+50}{3x_2+3x_3+50} + \frac{3x_1+4x_3+50}{4x_1+4x_2+5x_3+50} + \frac{x_1+2x_2+5x_3+50}{x_1+5x_2+5x_3+50} \\ & + \frac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50} \\ \text{s.t.} & 2x_1+x_2+5x_3 \leq 10, \\ & x_1+6x_2+3x_3 \leq 10, \\ & 5x_1+9x_2+2x_3 \leq 10, \\ & 9x_1+7x_2+3x_3 \leq 10, \\ & x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0,\ x_4 \geq 0. \end{array}$$

Example 3[23]

$$\begin{array}{ll} \min & \dfrac{-x_1+2x_2+2}{3x_1-4x_2+5} + \dfrac{4x_1-3x_2+4}{-2x_1+x_2+3} \\ \text{s.t.} & x_1+x_2 \leq 1.5, \\ & x_1-x_2 \leq 0, \\ & 0 \leq x_1 \leq 1,\ 0 \leq x_2 \leq 1. \end{array}$$

Example 4[22,23]

$$\begin{array}{ll} \max & \dfrac{3x_1+5x_2+3x_3+50}{3x_1+4x_2+5x_3+50} + \dfrac{3x_1+4x_2+50}{4x_1+3x_2+2x_3+50} \\ & + \dfrac{4x_1+2x_2+4x_3+50}{5x_1+4x_2+3x_3+50} \\ \text{s.t.} & 6x_1+3x_2+3x_3 \leq 10, \\ & 10x_1+3x_2+8x_3 \leq 10, \\ & x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0. \end{array}$$

Example 5[11]

$$\begin{array}{ll} \max & \dfrac{63x_1-18x_2+39}{13x_1+26x_2+13} + \dfrac{13x_1+26x_2+13}{37x_1+73x_2+13} \\ & + \dfrac{37x_1+73x_2+13}{13x_1+13x_2+13} + \dfrac{13x_1+13x_2+13}{63x_1-18x_2+39} \\ \text{s.t.} & 5x_1-3x_2 = 3, \\ & 1.5 \leq x_1 \leq 3. \end{array}$$

Example 6[11]

$$\begin{array}{ll} \max & \frac{3x_1+4x_2+50}{3x_1+5x_2+4x_3+50} - \frac{3x_1+5x_2+3x_3+50}{5x_1+5x_2+4x_3+50} - \frac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50} \\ & - \frac{4x_1+3x_2+3x_3+50}{3x_2+3x_3+50} \\ \text{s.t.} & 6x_1+3x_2+3x_3 \leq 10, \\ & 10x_1+3x_2+8x_3 \leq 10, \\ & x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0. \end{array}$$

From Table I, it can be seen that, except for Examples 5 and 6, our algorithm can determine the global optimal solution effectively than that of the corresponding references.

The comparison results of Table II show that the pruning techniques are very good at improving the convergence speed of our algorithm.

Example 7

$$\begin{array}{ll} \min & \sum_{i=1}^{p} \delta_i \frac{\sum_{j=1}^{n} c_{ij}x_j + d_i}{\sum_{j=1}^{n} e_{ij}x_j + f_i} \\ \text{s.t.} & x \in D = \{x \in R^n \mid Ax \leq b\}, \end{array}$$

where the elements of the matrix $A \in R^{m\times n}$, $b \in R^{m\times 1}$ $c_{ij},\ e_{ij},\ d_i,\ f_i \in R$ are randomly generated in the interval [0,1], $\delta_i (i = 1, \cdots, p)$ are randomly generated in the interval [-1,1].

From Table III, the computational results show that our algorithm performs well on the test problems, and can solve them in a reasonable amount of time.

The results in Tables I-III show that our algorithm is both feasible and efficient.

REFERENCES

[1] C.D. Maranas, I.P. Androulakis, C.A. Floudas, A.J. Berger, J.M. Mulvey, "Solving long-term financial planning problems via global optimization", *Journal of Economic Dynamics and Control*, vol. 21, pp. 1405-1425, 1997.

[2] H.M. Markowitz, *Portfolio Selection*, 2nd edition, Basil Blackwell Inc., Oxford,1991.

[3] J.M. Henderson, R.E. Quandt, *Microeconomic Theory*, 2n dedition, Mc Graw-Hill, New York, 1971.

[4] I. Quesada, I.E. Grossmann, "Alternative bounding approximations for the global optimization of various engineering design problems", in: I.E. Grossmann (Ed.), in: Global Optimization in Engineering Design, Nonconvex Optimization and Its Applications, vol. 9, Kluwer AcademicPublishers, Norwell, MA, 1996, pp.309-331.

[5] R.W. Freund, F. Jarre, " Solving the sum-of-ratios problem by an interior-point method", *Journal of Global Optimization*, vol. 19, pp. 83-102, 2001.

[6] T. Matsui, "NP-hardness of linear multiplicative programming and related problems", *Journal of Global Optimization*, vol. 9, pp. 113-119, 1996.

[7] A. Charnes, W.W. Cooper, "Programming with linear fractional functionk", *Naval Research Logistick Quarterly*, vol. 9, pp. 181-186, 1962.

[8] H. Konno, Y. Yajima, T. Maktkui, "Parametric simples algorithms for solving a special class of nonconvex minimization problems", *Journal of Global Optimization*, vol. 1, pp. 65-81, 1991.

[9] H. Konno, N. Abe, "Minimization of the sum of three linear fractional functions", *Journal of Global Optimization*, vol. 15, no. 4, pp. 419-432, 1999.

[10] Y.J. Wang, P.P. Shen, Z.A. Liang, "A branch-and-bound algorithm to globally solve the sum of several linear ratios", *Applied Mathematics and Computation*, vol. 168, no. 1, pp. 89-101, 2005.

[11] P.P. Shen, C.F. Wang, "Global optimization for sum of linear ratios problem with coeicients",*Applied Mathematics and Computation*, vol. 176, no. 1, pp. 219-229, 2006.

[12] H.P. Benson, "Solving sum of ratios fractional programs via concave minimization", *Journal of Optimization Theory and Application*, vol. 135, no. 1, pp. 1-17, 2007.

[13] D. Depetrini, M. Locatelli, "Approximation of linear fractional-multiplicative problems", *Mathematical Programming*, vol. 128, pp. 437-443, 2011.

[14] H.P. Benson, "On the global optimization of sums of linear fractional functions over a convex set", *Journal of Optimization Theory and Application*, vol. 121, pp. 19-39, 2004.

[15] T.V. Gruzdeva, A.S. Strekalovsky, " On solving the sum-of-ratios problem", *Applied Mathematics and Computation*, 318:260-269, 2018.

[16] Y. Ji, K.C. Zhang, S.J. Qu, "A deterministic global optimization algorithm",*Applied Mathematics and Computation*, vol. 185, pp. 382-387, 2007.

[17] J.G. Carlsson, J.M. Shi," A linear relaxation algorithm for solving the sum-of-linear-ratios problem with lower dimension", *Operations Research Letter*, vol. 41, pp. 381-389, 2013.

[18] H.W. Jiao , S.Y. Liu, A practicable branchand bound algorithm forsum of linear ratios problem, *European Journal of Operational Research*, vol. 243, pp. 723-730, 2015.

[19] C.F. Wang, P.P. Shen, "A global optimization algorithm for linear fractional programming", *Applied Mathematics and Computation*, vol. 204, pp. 281-287, 2008.

[20] H.P. Benson, "A simplicial branch and bound duality-bounds algorithm for the linear sum-of-ratios problem", *European Journal of Operational Research*, vol. 182, pp. 597-611, 2007.

[21] P.P. Shen, B.D. Huang, L.F. Wang, "Range division and linearization algorithm for a class of linear ratios optimization problems", *Journal of Computational and Applied Mathmatics*, vol. 305, pp. 324-342.

[22] Y.G. Pei, D.T. Zhu, "Global optimization method for maximizing the sum of difference of convex functions ratios over nonconvex region", *Journal of Applied Mathematics and Computing*, vol. 41, pp. 153-169, 2013.

[23] H.W. Jiao, "A branch and bound algorithm for globally solving a class of nonconvex programming problems", *Nonlinear Analysis: Theory, Methods and Applications*, vol. 70, pp. 1113-1123, 2009.

TABLE I
COMPUTATIONAL RESULTS OF EXAMPLES 1-6

Example	ϵ	Methods	Optimal solution	Optimal value	Iter
1	1e-9	[1]	(0.0, 1.0)	3.575	1
	1e-9	$ours$	(0.0, 1.0)	3.575	1
2	1e-9	[1]	(1.1111, 0.0, 0.0)	4.0907	1289
	1e-6	[11]	(1.1111, 1.365e-5, 1.351e-5)	4.081481	39
	1e-5	[22]	(0.0013, 0.0, 0.0)	4.087412	1640
	1e-9	$ours$	(1.1111, 0.0, 0.0)	4.0907	6
3	1e-8	[22]	(0.0, 0.283935547)	1.623183358	71
	1e-8	$ours$	(0.0, 0.283935547)	1.623183358	19
4	1e-5	[21]	(0.0, 1.6725, 0.0)	3.0009	1033
	1e-8	[22]	(0.0, 3.3333, 0.0)	3.00292	119
	1e-8	$ours$	(0.0, 3.3333, 0.0)	3.00292	21
5	1e-6	[12]	(3.0, 4.0)	3.2917	9
	1e-6	$ours$	(3.0, 4.0)	3.2917	16
6	1e-6	[12]	(-1.838e-16, 3.3333, 0.0)	1.9	8
	1e-6	$ours$	(0.0, 3.3333, 0.0)	1.9	16

TABLE II
COMPUTATIONAL RESULTS OF ALGORITHM 1 AND ALGORITHM 2 FOR EXAMPLES 1-6

Example	Methods	Optimal solution	Optimal value	Iter	$Time$
1	Algorithm 1	(0.0, 1.0)	3.575	1	0.016
	Algorithm 2	(0.0, 1.0)	3.575	13	0.344
2	Algorithm 1	(1.1111, 0.0, 0.0)	4.0907	6	0.187
	Algorithm 2	(1.1111, 0.0, 0.0)	4.0907	25	0.875
3	Algorithm 1	(0.0, 0.283935547)	1.623183358	16	1.0
	Algorithm 2	(0.0, 0.283935547)	1.623183358	50	1.719
4	Algorithm 1	(0.0, 3.3333, 0.0)	3.00092	20	0.625
	Algorithm 2	(0.0, 3.3333, 0.0)	3.00292	83	2.797
5	Algorithm 1	(3.0, 4.0)	3.2917	16	0.375
	Algorithm 2	(3.0, 4.0)	3.2917	41	1.118
6	Algorithm 1	(0, 3.3333, 0.0)	1.9	16	0.375
	Algorithm 2	(0.0, 3.3333, 0.0)	1.9	28	0.984

TABLE III
COMPUTATIONAL RESULTS OF EXAMPLE 7

(p, m, n)	Ave.Time	Ave.Iter
(2,20,20)	0.0264	1
(2,20,30)	0.029	1
(5,20,20)	0.6154	9.7
(5,30,20)	0.6389	10.22
(7,20,20)	2.2765	14.5
(7,30,20)	4.3327	18.8
(10,20,20)	31.1031	61.8
(10,30,20)	38.4108	95.3