

Research Article

A New Linearizing Method for Sum of Linear Ratios Problem with Coefficients

Hongwei Jiao¹ and Yongqiang Chen^{2,3}

¹ School of Mathematics, Henan Institute of Science and Technology, Xinxiang 453003, China

² Department of Mathematics, Henan Normal University, Xinxiang 453007, China

³ Department of Mathematics, Beijing Jiaotong University, Beijing 100044, China

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

Received 11 December 2013; Accepted 24 February 2014; Published 26 March 2014

Academic Editor: Ferenc Hartung

Copyright © 2014 H. Jiao and Y. Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new linearizing method is presented for globally solving sum of linear ratios problem with coefficients. By using the linearizing method, linear relaxation programming (LRP) of the sum of linear ratios problem with coefficients is established, which can provide the reliable lower bound of the optimal value of the initial problem. Thus, a branch and bound algorithm for solving the sum of linear ratios problem with coefficients is put forward. By successively partitioning the linear relaxation of the feasible region and solving a series of the LRP, the proposed algorithm is convergent to the global optimal solution of the initial problem. Compared with the known methods, numerical experimental results show that the proposed method has the higher computational efficiency in finding the global optimum of the sum of linear ratios problem with coefficients.

1. Introduction

In this paper, we consider the following sum of linear ratios problem with coefficients:

$$(\text{SLRC}) : \begin{cases} \min & H(x) = \sum_{j=1}^p H_j(x) = \sum_{j=1}^p \delta_j \frac{t_j(x)}{s_j(x)} \\ \text{s.t.} & x \in D = \{x \in R^n \mid Ax \leq b, x \geq 0\}, \end{cases} \quad (1)$$

where $t_j(x)$ and $s_j(x)$ are all linear functions, δ_j are all arbitrary real coefficients, $j = 1, \dots, p$, $A \in R^{m \times n}$, $b \in R^m$, and D is a nonempty compact set. Here, symbols δ_j in the problem (SLRC) can be considered the weight of each linear ratio, since δ_j are all arbitrary real numbers; thus the sum of linear ratios with coefficients δ_j can be called the weighted sum and difference of linear ratios. Therefore, the problem (SLRC) considered in this paper popularizes the mathematical model of sum of linear ratios problem investigated in many literatures.

The sum of linear ratios problem with coefficients (SLRC) has attracted the interest of researchers and practitioners for many years. In part, this is because the problem (SLRC) and its special case have broad applications in many practical

problems, for example, profit rates and pricing decisions problem [1], cluster analysis problem [2], bond portfolio optimization problem [3], and so forth. Another reason for the strong interest in the problem (SLRC) is that from a research point of view the problem (SLRC) poses many important theoretical and computational difficulties. This is mainly because the problem (SLRC) is a global optimization problem; that is, it shows multiple local optimal solutions which are not global optimal solutions.

During the past 20 years, many algorithms have been presented for globally solving special cases of the problem (SLRC), which are only developed for solving the sum of linear ratios problem with the assumption that all coefficients $\delta_j = +1$, $j = 1, \dots, p$. For example, when feasible region D is a polyhedral set and $p = 2$, Konno et al. [4] proposed a parametric simplex method for sum of linear ratios problem without coefficients. When all numerators $t_j(x) \geq 0$ and all denominators $s_j(x) \geq 0$, Wang et al. [5] proposed a linear relaxation method for the sum of linear ratios problem without coefficients. When the numerators $t_j(x) \geq 0$ and the denominators $s_j(x) \neq 0$, Wang and Shen [6] and Ji et al. [7] presented two different branch and bound algorithms for globally solving the sum of linear ratios problem; Shen

and Wang [8] first proposed a global optimization algorithm for the sum of linear ratios problem with coefficients, which is investigated in this paper. Recently, by using the linear characteristics of convex envelope and concave envelope of double variables product function, Gao and Jin [9] proposed an effective branch and bound algorithm for solving sum of linear ratios problem; by using subgradient and convex envelope, Pei and Zhu [10] presented a global optimization method for maximizing the sum of difference of convex functions ratios over nonconvex region, which can be used to globally solve the sum of linear ratios problem. However, to our knowledge, although there exist many algorithms to solve special cases of the problem (SLRC), due to its intrinsic difficulty, less theoretical research and algorithm design have been still expanded to globally solving the sum of linear ratios problem with coefficients (SLRC).

The purpose of this paper is to develop a new linearizing method for globally solving the sum of linear ratios problem with coefficients (SLRC). The proposed new linearizing method uses more information of the objective function of the problem (SLRC); by using the method, the linear relaxation programming (LRP) of the problem (SLRC) is established, which can provide a tighter lower bound of the global optimal value of the problem (SLRC) than the previous linearizing method in branch and bound algorithm and which can be used to reduce rapidly the growth of branching tree in the branch and bound algorithm for solving the problem (SLRC); therefore it can improve the computational efficiency of the algorithm. By successively subdividing feasible region of the problem (SLRC) and solving a series of the LRP, the proposed algorithm is convergent to the global optimum of the problem (SLRC). Finally, compared with the known methods, numerical experimental results imply that the proposed new linearizing method can be used to globally solve the problem (SLRC) with the higher computational efficiency.

The remainder sections of this paper are organized as follows. In Section 2, a new linearizing method is proposed; by utilizing the method, the linear relaxation programming (LRP) of the problem (SLRC) is established. In Section 3, a branch and bound algorithm for globally solving the problem (SLRC) is presented. Several test examples in the recent literatures and their numerical results obtained are listed in Section 4. At last, some concluding remarks are described in Section 5.

2. New Linearizing Method

To globally solve the problem (SLRC), by solving $2n$ simple linear programming problems, we obtain easily the initial partitioned rectangle

$$X^0 = \left\{ x \mid \min_{x \in D} x_i = \underline{x}_i \leq x_i \leq \bar{x}_i = \max_{x \in D} x_i, \quad i = 1, \dots, n \right\}. \quad (2)$$

By the characteristic of fractional function $t_j(x)/s_j(x)$, $j = 1, \dots, p$, we have the denominator $s_j(x) \neq 0$; that is, $s_j(x) > 0$ or $s_j(x) < 0$. If $s_j(x) < 0$, by letting $t_j(x)/s_j(x) =$

$-t_j(x)/-s_j(x)$, we have $-s_j(x) > 0$. Thus, we can always assume that all denominators $s_j(x) > 0$. If $t_j(x)$ is an arbitrary function, then there always exists an enough large positive number σ_j such that $t_j(x) + \sigma_j s_j(x) > 0$; thus we can assume $t_j(x) > 0$. Therefore, we can always suppose that $t_j(x) > 0$ and $s_j(x) > 0$, for all $j = 1, \dots, p$.

The important structure in the establishment of a branch and bound procedure to globally solve the problem (SLRC) is the calculation of lower bounds for this problem and for its subproblems. A lower bound of the global optimal value of the problem (SLRC) and its subproblems can be computed by solving a sequence of linear relaxation programming problems. The proposed new linearizing method for establishing linear relaxation programming of the problem (SLRC) is to underestimate or overestimate each function $H_j(x) = t_j(x)/s_j(x)$ with linear functions $L_j(x)$ or $U_j(x)$, $j = 1, 2, \dots, p$. By the former assumption, let

$$H_j(x) = \exp \left\{ \ln(t_j(x)) - \ln(s_j(x)) \right\}. \quad (3)$$

Firstly, for all $x \in X \subseteq X^0$, for each $j \in \{1, \dots, p\}$, let

$$\begin{aligned} t_j^l &= \min_{x \in X} t_j(x), & t_j^u &= \max_{x \in X} t_j(x), \\ k_j^1 &= \frac{\ln(t_j^u) - \ln(t_j^l)}{t_j^u - t_j^l}, \\ s_j^l &= \min_{x \in X} s_j(x), & s_j^u &= \max_{x \in X} s_j(x), \\ k_j^2 &= \frac{\ln(s_j^u) - \ln(s_j^l)}{s_j^u - s_j^l}. \end{aligned} \quad (4)$$

Let $L(\ln(Z))$ and $U(\ln(Z))$ be the linear underestimating function and the linear overestimating function of the $\ln(Z)$ over the interval $[Z^l, Z^u]$. By the linear characteristic of the concave function $\ln(Z)$, we can get the following inequalities:

$$\begin{aligned} L(\ln(Z)) &= k(Z - Z^l) + \ln Z^l \\ &\leq \ln(Z) \\ &\leq kZ - 1 - \ln k \\ &= U(\ln(Z)), \end{aligned} \quad (5)$$

where

$$k = \frac{\ln Z^u - \ln Z^l}{Z^u - Z^l}. \quad (6)$$

By the above inequality (5), it follows that

$$\begin{aligned} k_j^1 [t_j(x) - t_j^l] + \ln t_j^l &\leq \ln(t_j(x)) \leq k_j^1 t_j(x) - 1 - \ln k_j^1, \\ k_j^2 [s_j(x) - s_j^l] + \ln s_j^l &\leq \ln(s_j(x)) \leq k_j^2 s_j(x) - 1 - \ln k_j^2. \end{aligned} \quad (7)$$

By the above inequalities (7), for each $j \in \{1, \dots, p\}$, then we can establish the underestimating function $H_j^l(x)$

and the overestimating function $H_j^u(x)$ of the function $H_j(x)$ as follows:

$$\begin{aligned}
 H_j^l(x) &= \exp \left\{ k_j^1 [t_j(x) - t_j^l] + \ln t_j^l - [k_j^2 s_j(x) - 1 - \ln k_j^2] \right\}, \\
 H_j^u(x) &= \exp \left\{ k_j^1 t_j(x) - 1 - \ln k_j^1 - k_j^2 [s_j(x) - s_j^l] - \ln s_j^l \right\}.
 \end{aligned} \tag{8}$$

It is obvious that we have

$$H_j^l(x) \leq H_j(x) \leq H_j^u(x), \quad \text{for } \forall x \in X \subseteq X^0. \tag{9}$$

Secondly, for all $x \in X \subseteq X^0$, we let

$$\begin{aligned}
 Z_j &= k_j^1 [t_j(x) - t_j^l] + \ln t_j^l - [k_j^2 s_j(x) - 1 - \ln k_j^2], \\
 Z_j^u &= \max_{x \in X} \left\{ k_j^1 [t_j(x) - t_j^l] + \ln t_j^l - [k_j^2 s_j(x) - 1 - \ln k_j^2] \right\}, \\
 Z_j^l &= \min_{x \in X} \left\{ k_j^1 [t_j(x) - t_j^l] + \ln t_j^l - [k_j^2 s_j(x) - 1 - \ln k_j^2] \right\}, \\
 A_j^1 &= \frac{\exp(Z_j^u) - \exp(Z_j^l)}{Z_j^u - Z_j^l},
 \end{aligned}$$

$$\begin{aligned}
 Y_j &= k_j^1 t_j(x) - 1 - \ln k_j^1 - k_j^2 [s_j(x) - s_j^l] - \ln s_j^l, \\
 Y_j^u &= \max_{x \in X} \left\{ k_j^1 t_j(x) - 1 - \ln k_j^1 - k_j^2 [s_j(x) - s_j^l] - \ln s_j^l \right\}, \\
 Y_j^l &= \min_{x \in X} \left\{ k_j^1 t_j(x) - 1 - \ln k_j^1 - k_j^2 [s_j(x) - s_j^l] - \ln s_j^l \right\}, \\
 A_j^2 &= \frac{\exp(Y_j^u) - \exp(Y_j^l)}{Y_j^u - Y_j^l}.
 \end{aligned} \tag{10}$$

By the characteristic of the convex function, we can derive its linear underestimating function $L(\exp(Z_j))$ of $\exp(Z_j)$ over the interval $[Z_j^l, Z_j^u]$ as follows:

$$L_j(\exp(Z_j)) = A_j^1 (1 + Z_j - \ln A_j^1), \tag{11}$$

such that

$$L_j(\exp(Z_j)) \leq \exp(Z_j). \tag{12}$$

Based on the above discussion, for each $j \in \{1, \dots, p\}$, by (11)-(12), then finally we can derive the linear underestimating function $L_j(x)$ of $H_j^l(x)$, which underestimates the value of the function $H_j^l(x)$ as follows:

$$\begin{aligned}
 L_j(x) &= A_j^1 \left\{ 1 + k_j^1 [t_j(x) - t_j^l] + \ln t_j^l \right. \\
 &\quad \left. - [k_j^2 s_j(x) - 1 - \ln k_j^2] - \ln A_j^1 \right\},
 \end{aligned} \tag{13}$$

such that

$$L_j(x) \leq H_j^l(x), \quad \text{for } \forall x \in X \subseteq X^0. \tag{14}$$

Similarly, for each convex function $\exp(Y_j)$ over the interval $[Y_j^l, Y_j^u]$, we can derive its linear overestimating function $U(\exp(Y_j))$ as follows:

$$U_j(\exp(Y_j)) = A_j^2 (Y_j - Y_j^l) + \exp(Y_j^l), \tag{15}$$

such that

$$U_j(\exp(Y_j)) \geq \exp(Y_j). \tag{16}$$

According to the above discussion, for each $j \in \{1, \dots, p\}$, by (15)-(16), then finally we derive the linear overestimating function $U_j(x)$ of $H_j^u(x)$, which overestimates the optimal value of the function $H_j^u(x)$ as follows:

$$\begin{aligned}
 U_j(x) &= A_j^2 \left\{ k_j^1 t_j(x) - 1 - \ln k_j^1 \right. \\
 &\quad \left. - k_j^2 [s_j(x) - s_j^l] - \ln s_j^l - Y_j^l \right\} + \exp(Y_j^l),
 \end{aligned} \tag{17}$$

such that

$$U_j(x) \geq H_j^u(x), \quad \text{for } \forall x \in X \subseteq X^0. \tag{18}$$

By (9), (14), and (18), we get

$$L_j(x) \leq H_j^l(x) \leq H_j(x) \leq H_j^u(x) \leq U_j(x), \tag{19}$$

for $\forall x \in X \subseteq X^0$.

Let

$$H^L(x) = \sum_{j=1, \delta_j > 0}^p \delta_j L_j(x) + \sum_{j=1, \delta_j < 0}^p \delta_j U_j(x). \tag{20}$$

Then, by (19), we have

$$\begin{aligned}
 H^L(x) &= \sum_{j=1, \delta_j > 0}^p \delta_j L_j(x) \\
 &\quad + \sum_{j=1, \delta_j < 0}^p \delta_j U_j(x) \\
 &\leq \sum_{j=1}^p \delta_j H_j(x) \\
 &\leq H(x).
 \end{aligned} \tag{21}$$

According to the above linearizing method, for $\forall X \subseteq X^0$, we can construct the linear relaxation programming (LRP) of the problem (SLRC) over X as follows:

$$\text{(LRP)} : \begin{cases} \min & H^L(x) \\ \text{s.t.} & Ax \leq b, \quad x \geq 0, \\ & x \in X. \end{cases} \tag{22}$$

According to the construction method of the linear relaxation programming (LRP), for $\forall X \subseteq X^0$, the problem (LRP) provides a valid lower bound for the global optimal value of the problem (SLRC).

The following theorem ensures that the linear function $H^L(x)$ will approximate infinitely the corresponding function $H(x)$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$.

Theorem 1. For all $x \in X = [\underline{x}, \bar{x}] \subseteq X^0$, then one can follow that the error $\Delta = H(x) - H^L(x) \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$.

Proof. Let

$$\Delta_j^1 = \frac{t_j(x)}{s_j(x)} - L_j(x), \quad \Delta_j^2 = U_j(x) - \frac{t_j(x)}{s_j(x)}. \quad (23)$$

Then, we have

$$\begin{aligned} \Delta &= H(x) - H^L(x) \\ &= \left(\sum_{j=1}^p \delta_j \frac{t_j(x)}{s_j(x)} \right) \\ &\quad - \left(\sum_{j=1, \delta_j > 0} \delta_j L_j(x) + \sum_{j=1, \delta_j < 0} \delta_j U_j(x) \right) \\ &= \sum_{j=1, \delta_j > 0} \delta_j \left(\frac{t_j(x)}{s_j(x)} - L_j(x) \right) \\ &\quad - \sum_{j=1, \delta_j < 0} \delta_j \left(U_j(x) - \frac{t_j(x)}{s_j(x)} \right) \\ &= \sum_{j=1, \delta_j > 0} \delta_j \Delta_j^1 - \sum_{j=1, \delta_j < 0} \delta_j \Delta_j^2. \end{aligned} \quad (24)$$

Firstly, in the following, we will prove that $\Delta_j^1 \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$. Since

$$\Delta_j^1 = [H_j(x) - H_j^l(x)] + [H_j^l(x) - L_j(x)] = \Delta_{j1}^1 + \Delta_{j2}^1, \quad (25)$$

then we only need to prove that $\Delta_{j1}^1 \rightarrow 0$ and $\Delta_{j2}^1 \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$.

Let

$$\begin{aligned} f_j(x) &= \ln(t_j(x)) - \ln(s_j(x)), \\ g_j(x) &= k_j^1 t_j(x) - 1 - \ln k_j^1 - k_j^2 [s_j(x) - s_j^l] - \ln s_j^l, \\ h_j(x) &= k_j^1 [t_j(x) - t_j^l] + \ln t_j^l - [k_j^2 s_j(x) - 1 - \ln k_j^2]. \end{aligned} \quad (26)$$

Then, we have

$$\begin{aligned} \Delta_{j1}^1 &= \exp \{ \ln(t_j(x)) - \ln(s_j(x)) \} \\ &\quad - \exp \{ k_j^1 [t_j(x) - t_j^l] \\ &\quad \quad + \ln t_j^l - [k_j^2 s_j(x) - 1 - \ln k_j^2] \} \end{aligned} \quad (27)$$

$$\begin{aligned} &= \exp(f_j(x)) - \exp(h_j(x)) \\ &\leq \|f_j(x) - h_j(x)\| \sup_{\xi_j \in L(f_j(x), h_j(x))} \exp(\xi_j), \end{aligned}$$

where

$$\begin{aligned} L(f_j(x), h_j(x)) &= \alpha f_j(x) + (1 - \alpha) h_j(x) \\ &\quad \text{with } \alpha \in [0, 1]. \end{aligned} \quad (28)$$

Let

$$\begin{aligned} f_j(x) - h_j(x) &= \{ \ln(t_j(x)) - \ln(s_j(x)) \} \\ &\quad - \{ k_j^1 [t_j(x) - t_j^l] + \ln t_j^l \\ &\quad \quad - [k_j^2 s_j(x) - 1 - \ln k_j^2] \} \\ &= \{ \ln(t_j(x)) - [k_j^1 [t_j(x) - t_j^l] + \ln t_j^l] \} \\ &\quad + \{ [k_j^2 s_j(x) - 1 - \ln k_j^2] - \ln(s_j(x)) \} \\ &= \Delta_{j1.1}^1 + \Delta_{j1.2}^1. \end{aligned} \quad (29)$$

Since $\Delta_{j1.1}^1 = \ln(t_j(x)) - [k_j^1 [t_j(x) - t_j^l] + \ln t_j^l]$ is a concave function about $t_j(x)$, we know that $\Delta_{j1.1}^1$ can attain the maximum $\Delta_{j1.1}^{1, \max}$ at the point $t_j(x) = 1/k_j^1$. Let $u_j = t_j^u/t_j^l$; then through computing, we can derive

$$\Delta_{j1.1}^{1, \max} = \frac{\ln u_j}{u_j - 1} - 1 - \ln \frac{\ln u_j}{u_j - 1}. \quad (30)$$

Since $u_j \rightarrow 1$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$, then we have $\Delta_{j1.1}^{1, \max} \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$.

Since $\Delta_{j1.2}^1 = [k_j^2 s_j(x) - 1 - \ln k_j^2] - \ln(s_j(x))$ is a convex function about $s_j(x)$, it can attain the maximum $\Delta_{j1.2}^{1, \max}$ at the points s_j^u or s_j^l . Let $v_j = s_j^u/s_j^l$; then through computing, we can derive

$$\Delta_{j1.2}^{1, \max} = \frac{\ln v_j}{v_j - 1} - 1 - \ln \frac{\ln v_j}{v_j - 1}. \quad (31)$$

Since $v_j \rightarrow 1$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$, then we have $\Delta_{j1.2}^{1,\max} \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$.

Therefore, we have

$$\|f_j(x) - h_j(x)\| = \|\Delta_{j1.1}^1 + \Delta_{j1.2}^1\| \leq \|\Delta_{j1.1}^1\| + \|\Delta_{j1.2}^1\|. \quad (32)$$

By the above proof, we have

$$\|f_j(x) - h_j(x)\| \rightarrow 0, \quad \text{as } \|\bar{x} - \underline{x}\| \rightarrow 0. \quad (33)$$

Since $\exp(\xi_j)$ is a continuous and bounded function about variable x and by the above (27) and (33), we have

$$\Delta_{j1}^1 \rightarrow 0, \quad \text{as } \|\bar{x} - \underline{x}\| \rightarrow 0. \quad (34)$$

Next, we consider the difference Δ_{j2}^1 , and it follows that

$$\begin{aligned} \Delta_{j2}^1 &= H_j^l(x) - L_j(x) \\ &= \exp\{k_j^1[t_j(x) - t_j^l] + \ln(t_j^l) - [k_j^2 s_j(x) - 1 - \ln k_j^2]\} \\ &\quad - A_j^1\{1 + k_j^1[t_j(x) - t_j^l] + \ln(t_j^l) \\ &\quad - [k_j^2(s_j(x) - 1 - \ln k_j^2) - \ln A_j^1]\} \\ &= \exp(Z_j) - A_j^1\{1 + Z_j - \ln A_j^1\}. \end{aligned} \quad (35)$$

Since Δ_{j2}^1 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 Δ_{j2}^1 can obtain the maximum $\Delta_{j2}^{1,\max}$ at the points Z_j^l or Z_j^u . Let

$$R_j = \frac{\exp(Z_j^u - Z_j^l) - 1}{Z_j^u - Z_j^l}. \quad (36)$$

Then, through computing, we can follow that

$$\begin{aligned} \Delta_{j2}^{1,\max} &= \Delta_{j2}^1(Z_j^l) \\ &= \Delta_{j2}^1(Z_j^u) \\ &= \exp(Z_j^l)(1 - R_j + R_j \ln R_j). \end{aligned} \quad (37)$$

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

Secondly, we will prove that $\Delta_j^2 \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$. Since

$$\Delta_j^2 = [U_j(x) - H_j^u(x)] + [H_j^u(x) - H_j(x)] = \Delta_{j1}^2 + \Delta_{j2}^2, \quad (38)$$

then we only need to prove that $\Delta_{j1}^2 \rightarrow 0$ and $\Delta_{j2}^2 \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$. Thus,

$$\begin{aligned} \Delta_{j1}^2 &= U_j(x) - H_j^u(x) \\ &= A_j^2\{k_j^1 t_j(x) - 1 - \ln k_j^1 - k_j^2[s_j(x) - s_j^l] - \ln s_j^l - Y_j^l\} \\ &\quad + \exp(Y_j^l) - \exp\{k_j^1 t_j(x) - 1 - \ln k_j^1 \\ &\quad - k_j^2[s_j(x) - s_j^l] - \ln s_j^l\} \\ &= A_j^2(Y_j - Y_j^l) + \exp(Y_j^l) - \exp(Y_j). \end{aligned} \quad (39)$$

Since Δ_{j1}^2 is a concave function about Y_j , for any $Y_j \in [Y_j^l, Y_j^u]$ defined in the former, then it follows that Δ_{j1}^2 can obtain the maximum $\Delta_{j1}^{2,\max}$ at the points $\ln(A_j^2)$. Let

$$W_j = \frac{\exp(Y_j^u - Y_j^l) - 1}{Y_j^u - Y_j^l}. \quad (40)$$

Then, through computing, we can get

$$\Delta_{j1}^{2,\max} = \exp(Y_j^l)(1 - W_j + W_j \ln W_j). \quad (41)$$

Since $W_j \rightarrow 1$ as $|Y_j^u - Y_j^l| \rightarrow 0$ and $|Y_j^u - Y_j^l| \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$, it is obvious that $\Delta_{j1}^{2,\max} \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$. Therefore, we have $\Delta_{j1}^2 \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$.

Thus,

$$\begin{aligned} \Delta_{j2}^2 &= H_j^u(x) - H_j(x) \\ &= \exp\{k_j^1 t_j(x) - 1 - \ln k_j^1 - k_j^2[s_j(x) - s_j^l] - \ln s_j^l\} \\ &\quad - \exp\{\ln(t_j(x)) - \ln(s_j(x))\} \\ &= \exp(g_j(x)) - \exp(f_j(x)) \\ &\leq \|g_j(x) - f_j(x)\| \sup_{\eta_j \in L(g_j(x), f_j(x))} \exp(\eta_j), \end{aligned} \quad (42)$$

where

$$L(g_j(x), f_j(x)) = \beta g_j(x) + (1 - \beta) f_j(x) \quad \text{with } \beta \in [0, 1]. \quad (43)$$

Let

$$\begin{aligned} g_j(x) - f_j(x) &= \{k_j^1 t_j(x) - 1 - \ln k_j^1 \\ &\quad - k_j^2[s_j(x) - s_j^l] - \ln s_j^l\} \\ &\quad - \{\ln(t_j(x)) - \ln(s_j(x))\} \end{aligned}$$

$$\begin{aligned}
&= \{ [k_j^1 t_j(x) - 1 - \ln k_j^1] - \ln(t_j(x)) \} \\
&\quad + \{ \ln(s_j(x)) - [k_j^2 [s_j(x) - s_j^l] + \ln s_j^l] \} \\
&= \Delta_{j2.1}^2 + \Delta_{j2.2}^2.
\end{aligned} \tag{44}$$

Since $\Delta_{j2.1}^2 = [k_j^1 t_j(x) - 1 - \ln k_j^1] - \ln(t_j(x))$ is a convex function about $t_j(x)$, it can attain the maximum $\Delta_{j2.1}^{2,\max}$ at the points t_j^u or t_j^l . Then, through computing, we derive

$$\Delta_{j1.1}^{2,\max} = \frac{\ln u_j}{u_j - 1} - 1 - \ln \frac{\ln u_j}{u_j - 1}. \tag{45}$$

Since $u_j \rightarrow 1$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$, then we have $\Delta_{j2.1}^{2,\max} \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$.

Since $\Delta_{j2.2}^2 = \ln(s_j(x)) - [k_j^2 [s_j(x) - s_j^l] + \ln s_j^l]$ is a concave function about $s_j(x)$, we can get the fact that $\Delta_{j2.2}^2$ can attain the maximum $\Delta_{j2.2}^{2,\max}$ at the point $s_j(x) = 1/k_j^2$. Then, through computing, we derive

$$\Delta_{j2.2}^{2,\max} = \frac{\ln v_j}{v_j - 1} - 1 - \ln \frac{\ln v_j}{v_j - 1}. \tag{46}$$

Since $v_j \rightarrow 1$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$, then we have $\Delta_{j2.2}^{2,\max} \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$.

Therefore, we have

$$\|g_j(x) - f_j(x)\| = \|\Delta_{j2.1}^2 + \Delta_{j2.2}^2\| \leq \|\Delta_{j2.1}^2\| + \|\Delta_{j2.2}^2\|. \tag{47}$$

By the above proof, we have

$$\|g_j(x) - f_j(x)\| \rightarrow 0, \quad \text{as } \|\bar{x} - \underline{x}\| \rightarrow 0. \tag{48}$$

Since $\exp(\eta_j)$ is a continuous and bounded function about variable x and by (42) and (48), we have

$$\Delta_{j2}^2 \rightarrow 0, \quad \text{as } \|\bar{x} - \underline{x}\| \rightarrow 0. \tag{49}$$

By the above proof, we can follow that

$$\begin{aligned}
\Delta_j^2 &= [U_j(x) - H_j^u(x)] + [H_j^u(x) - H_j(x)] \\
&= \Delta_{j1}^2 + \Delta_{j2}^2, \rightarrow 0 \quad \text{as } \|\bar{x} - \underline{x}\| \rightarrow 0.
\end{aligned} \tag{50}$$

Therefore,

$$\begin{aligned}
\Delta &= H(x) - H^L(x) \\
&= \sum_{j=1, \delta_j > 0}^p \delta_j \Delta_j^1 - \sum_{j=1, \delta_j < 0}^p \delta_j \Delta_j^2 \rightarrow 0 \quad \text{as } \|\bar{x} - \underline{x}\| \rightarrow 0.
\end{aligned} \tag{51}$$

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

3. Algorithm and Its Convergence

In this section, based on the former new linearizing method, we present an effective branch and bound algorithm for globally solving the sum of linear ratios problem with coefficients (SLRC). The critical construction in ensuring that the proposed branch and bound algorithm is convergent to the global optimum of the problem (SLRC) is the selection of a reasonable branching rule. In this paper, we use a standard branching rule which is called bisection. The selected branching rule is described as follows.

Assume that the hyperrectangle $X^k = [\underline{x}^k, \bar{x}^k] \subseteq X^0$ is a node subproblem identified, select $\rho \in \operatorname{argmax} \{\bar{x}_i^k - \underline{x}_i^k : i = 1, \dots, n\}$, and subdivide X^k by partitioning the maximum edge of the interval $[\underline{x}_\rho^k, \bar{x}_\rho^k]$ into the two subintervals $[\underline{x}_\rho^k, (\underline{x}_\rho^k + \bar{x}_\rho^k)/2]$ and $[(\underline{x}_\rho^k + \bar{x}_\rho^k)/2, \bar{x}_\rho^k]$.

Assume that $LB(X^k)$ is the optimal value of the LRP(X^k) and $x^k = x(X^k)$ is the corresponding optimal solution of the LRP(X^k). The steps of the proposed algorithm are as follows.

Step 1. Set the initial convergence tolerance $\varepsilon > 0$, the initial number of iteration $k := 0$, and the set of initial active node $\Xi_0 = X^0$. Let the initial upper bound $UB_0 = +\infty$ and the initial feasible point set $\Theta := \emptyset$.

Calculate $LB_0 := LB(X^0)$ and $x^0 = x(X^0)$ by solving the LRP over X^0 ; if x^0 is feasible to the problem (SLRC), update the feasible point set Θ and the upper bound UB_0 , if necessary. If $UB_0 - LB_0 \leq \varepsilon$, then the algorithm stops with x^0 being the global optimal solution of the problem (SLRC); otherwise, continue the following Step 2.

Step 2. Subdivide hyperrectangle X^k into two new sub-hyper-rectangles by utilizing the proposed branching rule and denote the set of new partitioned subrectangles as \bar{X}^k .

Step 3. For every $X \in \bar{X}^k$, calculate $LB(X)$ and $x(X)$ by solving the LRP over X . If $LB(X) > UB_k$, then let $\bar{X}^k := \bar{X}^k \setminus X$; if $x(X)$ is feasible to the problem (SLRC), then update feasible point set Θ , the upper bound UB_k , and the best known feasible point x^k , if necessary, and let $\Xi_k = (\Xi_k \setminus X) \cup \bar{X}^k$ and update the lower bound $LB_k = \inf_{X \in \Xi_k} LB(X)$.

Step 4. Let $\Xi_{k+1} = \Xi_k \setminus \{X : UB_k - LB(X) \leq \varepsilon, X \in \Xi_k\}$. If $\Xi_{k+1} = \emptyset$, then the algorithm stops with UB_k being the global ε -optimal value for the problem (SLRC) and x^k being a global optimization solution for the problem (SLRC). Otherwise, let $k := k + 1$, select subrectangle X^k satisfying $X^k = \operatorname{arg min}_{X \in \Xi_k} LB(X)$, and return to Step 2.

Theorem 2. *The above proposed algorithm either stops finitely with the global optimal solution for the problem (SLRC) or generates an infinite sequence of iterations $\{x^k\}$, the limitation point of which will be the global optimal solution for the problem (SLRC).*

Proof. If the proposed algorithm stops finitely at iteration k , then, when the algorithm stops, we obtain $UB_k = v^* = LB_k$, where v^* is the global optimal value of the problem (SLRC). Thus, by the structure of the proposed algorithm, we know that x^k must be a global optimal solution for the problem (SLRC).

If the proposed algorithm generates an infinite sequence of partitioned subrectangles $\{X^k\}$, then, by exhaustiveness of the branching method, we know that the sequence $\{X^k\}$ must shrink to a singleton. By the design of the proposed algorithm we can follow that the sequence $\{UB_k\}$ is nonincreasing, and the sequence $\{LB_k\}$ is nondecreasing; thus, the sequence $\{UB_k - LB_k\}$ is nonincreasing. In conclusion, we can get the fact that the bounding operation is consistent and that the selection operation is bound improving; thus, by Theorem IV.3. in [11], we can follow easily that the sequence $\{UB_k - LB_k\}$ must converge to zero. Since $LB_k \leq v^* \leq UB_k$, this shows that $\lim_{k \rightarrow \infty} UB_k = v^*$. Since $UB_k = H(x^k)$, any cluster point x^* of the sequence $\{x^k\}$ is a feasible point for the problem (SLRC) with the objective function value $H(x^*)$. Therefore, the conclusion is followed. \square

4. Numerical Examples

To verify the reliability and effectiveness of the proposed new linearizing method, several test examples that appeared in the recent literatures are implemented on an Intel(R) Core(TM)2 Duo CPU (1.58 GHZ) microcomputer. The proposed algorithm using the new linearizing method is coded in C++ and every linear relaxation programming problem is solved by simplex method. These test examples and their numerical results are described as follows.

Example 1 (see [9]). We have the following:

$$\begin{aligned} \min \quad & \frac{37x_1 + 73x_2 + 13}{13x_1 + 13x_2 + 13} + \frac{63x_1 - 18x_2 + 39}{13x_1 + 26x_2 + 13} \\ \text{s.t.} \quad & 5x_1 - 3x_2 = 3, \\ & 1.5 \leq x_1 \leq 3. \end{aligned} \tag{52}$$

With $\epsilon = 10^{-4}$, using the method proposed in this paper, the optimal solution is $(x_1, x_2) = (1.500000000, 1.500000000)$, the global ϵ -optimal value is 4.912587413, the number of algorithm iterations is 29, the maximal number of algorithm active nodes necessary is 9, and the computational time is 0.12715 seconds.

Using the method in [9], with $\epsilon = 10^{-4}$, the optimal solution is $(x_1, x_2) = (1.5000, 1.5000)$, the global ϵ -optimal value is 4.9125, the number of algorithm iterations is 113, and the computational time is 201.626020 seconds.

Example 2 (see [6]). We have the following:

$$\begin{aligned} \max \quad & \frac{37x_1 + 73x_2 + 13}{13x_1 + 13x_2 + 13} + \frac{63x_1 - 18x_2 + 39}{13x_1 + 26x_2 + 13} \\ \text{s.t.} \quad & 5x_1 - 3x_2 = 3, \\ & 1.5 \leq x_1 \leq 3. \end{aligned} \tag{53}$$

With $\epsilon = 10^{-4}$, using the method proposed in this paper, the optimal solution is $(x_1, x_2) = (3.000000000, 4.000000000)$, the global ϵ -optimal value is 5.000000000, the number of algorithm iterations is 59, the maximal number of algorithm active nodes necessary is 19, and the computational time is 0.25843 seconds.

But, using the method in [6], with $\epsilon = 10^{-4}$, the optimal solution is $(x_1, x_2) = (3, 4)$, the global ϵ -optimal value is 5, the number of algorithm iterations is 32, the maximal number of algorithm active nodes necessary is 32, and the computational time is 1.089285 seconds.

Example 3 (see [10]). We have the following:

$$\begin{aligned} \max \quad & \frac{3x_1 + 5x_2 + 3x_3 + 50}{3x_1 + 4x_2 + 5x_3 + 50} + \frac{3x_1 + 5x_2 + 50}{3x_1 + 5x_2 + 3x_3 + 50} \\ & + \frac{4x_1 + 2x_2 + 4x_3 + 50}{5x_1 + 4x_2 + 3x_3 + 50} \\ \text{s.t.} \quad & 6x_1 + 3x_2 + 3x_3 \leq 10, \\ & 10x_1 + 3x_2 + 8x_3 \leq 10, \\ & x_1, x_2, x_3 \geq 0. \end{aligned} \tag{54}$$

With $\epsilon = 10^{-5}$, using the proposed method in this paper, the optimal solution is $(x_1, x_2, x_3) = (0.000000000, 0.000000000, 0.000000000)$, the global ϵ -optimal value is 3.000000000, the number of algorithm iterations is 37, the maximal number of algorithm active nodes necessary is 10, and the computational time is 0.109402 seconds.

Using the method in [10], with $\epsilon = 10^{-5}$, the optimal solution is $(x_1, x_2, x_3) = (0.0013, 1.6725, 0.0000)$, the global ϵ -optimal value is 3.0009, the number of algorithm iterations is 1033, the maximal number of algorithm active nodes necessary is 200, and the computational time is 99.3570 seconds.

Example 4 (see [10]). We have the following:

$$\begin{aligned} \max \quad & \frac{4x_1 + 3x_2 + 3x_3 + 50}{3x_2 + 3x_3 + 50} + \frac{3x_1 + 4x_2 + 50}{4x_1 + 4x_2 + 5x_3 + 50} \\ & + \frac{x_1 + 2x_2 + 5x_3 + 50}{x_1 + 5x_2 + 5x_3 + 50} \\ & + \frac{x_1 + 2x_2 + 4x_3 + 50}{5x_2 + 4x_3 + 50} \\ \text{s.t.} \quad & 2x_1 + x_2 + 5x_3 \leq 10, \\ & x_1 + 6x_2 + 3x_3 \leq 10, \\ & 5x_1 + 9x_2 + 2x_3 \leq 10, \\ & 9x_1 + 7x_2 + 3x_3 \leq 10, \\ & x_1, x_2, x_3 \geq 0. \end{aligned} \tag{55}$$

With $\epsilon = 10^{-5}$, using the proposed method in this paper, the optimal solution is $(x_1, x_2, x_3) = (1.111111111, 0.000000000, 0.000000000)$, the global ϵ -optimal value is 4.090702948, the number of algorithm

iterations is 21, the maximal number of algorithm active nodes necessary is 3, and the computational time is 0.0831093 seconds.

Using the method in [10], with $\epsilon = 10^{-5}$, the optimal solution is $(x_1, x_2, x_3) = (0.0013, 0.0000, 0.0000)$, the global ϵ -optimal value is 4.0001, the number of algorithm iterations is 1640, the maximal number of algorithm active nodes necessary is 233, and the computational time is 120.6355 seconds.

By substituting verification, we know that the global optimal solutions of Examples 3 and 4 obtained using our algorithm are feasible and correct and the global optimal values of Examples 3 and 4 obtained using our algorithm are correct. From the above comparison, we know that the optimal solutions and optimal values for Examples 3 and 4 using our method are much better than those in the literature [10]; that is, using our algorithm, we can obtain the more accurate global optimal solution.

From numerical results for Examples 1–4, the proposed new linearizing method can be used to globally solve the problem (SLRC) with the higher computational efficiency.

5. Some Extensions

The new linearizing method used for solving the problem (SLRC) can be extended to seek a global optimal solution of the sum of linear ratios problem with coefficients whose domain is not linear. Some extensions are given as follows.

When domain D is convex set, by solving $2n$ simple convex programming problems, we obtain easily the initial rectangle X^0 . For any $X \in X^0$, utilize the presented linearizing method to construct linear underestimating function of the sum of linear fractional function, so that we can establish a convex relaxation programming problem of the problem (SLRC) over X . Use the algorithm proposed in Section 3, at each iteration, and solve convex relaxation programming subproblems rather than linear relaxation programming subproblems; we can globally solve the problem (SLRC) over convex set.

When constraint functions of the problem (SLRC) are also sum of linear fractional functions with coefficients, whose mathematical modeling can be reformulated as follows:

$$\begin{cases} \min & H_0(x) = \sum_{q=1}^{T_0} \delta_{0q} \frac{t_{0q}(x)}{s_{0q}(x)} \\ \text{s.t.} & H_\sigma(x) = \sum_{q=1}^{T_\sigma} \delta_{\sigma q} \frac{t_{\sigma q}(x)}{s_{\sigma q}(x)} \leq B_\sigma, \\ & \sigma = 1, 2, \dots, M, \\ & x \in D = \{x \in R^n \mid Ax \leq b, x \geq 0\}, \end{cases} \tag{56}$$

where $t_{\sigma q}(x)$ and $s_{\sigma q}(x)$ are all linear affine functions, $\delta_{\sigma q}$ are all arbitrary real coefficients, $\sigma = 1, 2, \dots, M$, $A \in R^{m \times n}$, $b \in R^m$, $B \in R^M$, and D is a nonempty compact set.

Using the linearizing method proposed in Section 2 to construct the linear underestimating function $H_\sigma^L(x)$ of each

function $H_\sigma(x)$, $\sigma = 1, 2, \dots, M$, we can establish the linear relaxation programming (LRP1) of the problem (SLRC1) over X as follows:

$$\text{(LRP1)} : \begin{cases} \min & H_0^L(x) \\ \text{s.t.} & H_\sigma^L(x) \leq B_\sigma, \quad \sigma = 1, 2, \dots, M, \\ & Ax \leq b, \quad x \geq 0, \\ & x \in X \subseteq X^0. \end{cases} \tag{57}$$

Using the same algorithm step in Section 3, at each iteration, and solving linear relaxation programming subproblem (LRP1), we can globally solve the sum of linear ratios problem (SLRC1) with sum of linear ratios constraints. To verify the feasibility of the proposed new linearizing method, two test examples are implemented on a microcomputer; test examples and their computational results are given as follows.

Example 5. We have the following:

$$\begin{aligned} \min & \quad 1.6 \times \frac{\sum_{i=1}^N x_i + 1}{\sum_{i=1}^N x_i + 2} + 2.3 \times \frac{\sum_{i=1}^N x_i + 2}{\sum_{i=1}^N x_i + 3} - 3.1 \\ & \quad \times \frac{\sum_{i=1}^N x_i + 5}{\sum_{i=1}^N x_i + 4} - 4.2 \times \frac{\sum_{i=1}^N x_i + 6}{\sum_{i=1}^N x_i + 5} \\ \text{s.t.} & \quad \sum_{j=2}^5 \frac{\sum_{i=1}^N x_i + j}{\sum_{i=1}^N x_i + j + 1} \leq 3.61, \\ & \quad \sum_{j=3}^6 \frac{\sum_{i=1}^N x_i + j}{\sum_{i=1}^N x_i + j + 1} \leq 3.76, \\ & \quad \sum_{j=4}^7 \frac{\sum_{i=1}^N x_i + j}{\sum_{i=1}^N x_i + j + 1} \leq 3.83, \\ & \quad \sum_{j=5}^8 \frac{\sum_{i=1}^N x_i + j}{\sum_{i=1}^N x_i + j + 1} \leq 3.86, \\ & \quad 1.0 \leq x_i \leq 3.0, \quad i = 1, 2, \dots, N, \end{aligned} \tag{58}$$

where $N = 4$.

With $\epsilon = 10^{-8}$, using the proposed method in this paper, the global optimal solution is $(x_1, x_2, x_3, x_4) = (1.000000000, 1.000000000, 1.000000000, 1.000000000)$, the global ϵ -optimal value is -4.849404762 , the number of algorithm iterations is 181, and the maximal number of algorithm active nodes necessary is 64.

Example 6. We have the following:

$$\begin{aligned} \min & \quad \frac{2x_1 + x_2 - x_3 + x_4 + 1}{x_1 + 2x_2 - x_3 + x_4 + 2} \\ & \quad + \frac{x_1 + 2x_2 - x_3 + x_4 + 2}{2x_1 + 2x_2 + x_3 + x_4 + 3} \end{aligned}$$

$$\begin{aligned}
& - \frac{x_1 + x_2 + 3x_3 + x_4 + 5}{x_1 + 2x_2 + 3x_3 + x_4 + 4} \\
& - \frac{1.5x_1 - x_2 + 1.5x_3 + x_4 + 6}{1.4x_1 - x_2 + 2.5x_3 + 1.6x_4 + 5} \\
\text{s.t. } & \frac{2x_1 - x_2 + 3x_3 + x_4 + 2}{x_1 + 3x_2 + x_3 + 3x_4 + 3} \\
& + \frac{2x_1 + x_2 + 4x_3 + x_4 + 3}{x_1 + 4x_2 - x_3 + 5x_4 + 4} \\
& - \frac{x_1 + 2x_2 + x_3 + 3x_4 + 4}{1.5x_1 + x_2 + 3.5x_3 + x_4 + 5} \\
& + \frac{x_1 + 2x_2 + x_3 + x_4 + 5}{2x_1 - x_2 + 1.3x_3 + x_4 + 6} \leq 1.55, \\
& \frac{x_1 + 2x_2 + 2x_3 - x_4 + 3}{2x_1 - x_2 + 2x_3 + x_4 + 4} \\
& + \frac{3x_1 + x_2 + 4x_3 + x_4 + 4}{x_1 + 4x_2 - x_3 + 5x_4 + 5} \\
& + \frac{1.4x_1 - x_2 + 1.5x_3 + x_4 + 5}{1.5x_1 + x_2 + 1.7x_3 + 2x_4 + 6} \\
& - \frac{2x_1 - x_2 + 2x_3 + 2x_4 + 7}{x_1 + 3x_2 + 4x_3 + 3x_4 + 6} \leq 3.15, \\
& \frac{2x_1 - x_2 + 2x_3 + 2x_4 + 4}{3x_1 + 2x_2 + x_3 + 4x_4 + 5} \\
& - \frac{x_1 - x_2 + 2x_3 + 3x_4 + 5}{x_1 + x_2 + 3x_3 + 4x_4 + 6} \\
& + \frac{x_1 + x_2 + x_3 + x_4 + 6}{2x_1 + 2x_2 + 4x_3 + 3x_4 + 7} \\
& + \frac{x_1 + x_2 + 2x_3 + x_4 + 7.6}{x_1 + 2x_2 + x_3 + x_4 + 8.5} \leq 3.85, \\
& \frac{x_1 - 3x_2 + x_3 + 2x_4 + 5}{x_1 - 2x_2 + 2x_3 + 3x_4 + 6} \\
& + \frac{2.1x_1 - x_2 + x_3 + x_4 + 6}{x_1 + 2x_2 + 2x_3 + x_4 + 7} \\
& + \frac{x_1 + 2x_2 + 2x_3 + 3x_4 + 7}{x_1 + 2x_2 + 3x_3 + 3x_4 + 8} \\
& + \frac{1.5x_1 - x_2 + x_3 + x_4 + 8}{1.6x_1 + x_2 - x_3 + 2x_4 + 9} \leq 3.88, \\
& 1.0 \leq x_1, x_2, x_3, x_4 \leq 3.0.
\end{aligned} \tag{59}$$

With $\epsilon = 10^{-8}$, using the proposed method in this paper, the global optimal solution is $(x_1, x_2, x_3, x_4) = (1.000000000, 1.000000000, 3.000000000, 1.000000000)$, the global ϵ -optimal value is -0.888192268 , the number of algorithm iterations is 98, and the maximal number of algorithm active nodes necessary is 18.

From numerical results for Examples 5–6, the proposed new linearizing method can be extended to globally solve the problem (SLRC) with the weighted sum of linear fractional functions constraints.

It should also be noted that our approach could be extended to solve more general generalized linear fractional programming problems; this will constitute a subject for future research.

6. Concluding Remarks

In this paper, by utilizing the linear approximation of exponential and logarithmic functions, a new linearizing method is presented. Combining the linearizing method within the branch and bound scheme, a branch and bound algorithm is constructed for solving the problem (SLRC). By subsequently partitioning linear relaxation of the feasible region and solving a series of linear programming problems, the proposed algorithm is convergent to a global optimal solution of the problem (SLRC). Compared with the known methods, numerical experimental results show that the proposed new linearizing method can be used to globally solve the problem (SLRC) with the higher computational efficiency.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China under Grant no. 11171094 and the Science and Technology Key Project of Education Department of Henan Province no. 14A110024.

References

- [1] C. S. Colantoni, R. P. Manes, and A. Whinston, "Programming, profit rates, and pricing decisions," *Accounting Review*, vol. 44, pp. 467–481, 1969.
- [2] M. R. Rao, "Cluster analysis and mathematical programming," *Journal of the American Statistical Association*, vol. 66, pp. 622–626, 1971.
- [3] H. Konno and H. Watanabe, "Bond portfolio optimization problems and their applications to index tracking: a partial optimization approach," *Journal of the Operations Research Society of Japan*, vol. 39, no. 3, pp. 295–306, 1996.
- [4] H. Konno, Y. Yajima, and T. Matsui, "Parametric simplex algorithms for solving a special class of nonconvex minimization problems," *Journal of Global Optimization*, vol. 1, no. 1, pp. 65–81, 1991.
- [5] Y. J. Wang, P. P. Shen, and 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.
- [6] C.-F. Wang and P.-P. Shen, "A global optimization algorithm for linear fractional programming," *Applied Mathematics and Computation*, vol. 204, no. 1, pp. 281–287, 2008.

- [7] Y. Ji, K.-C. Zhang, and S.-J. Qu, "A deterministic global optimization algorithm," *Applied Mathematics and Computation*, vol. 185, no. 1, pp. 382–387, 2007.
- [8] P.-P. Shen and C.-F. Wang, "Global optimization for sum of linear ratios problem with coefficients," *Applied Mathematics and Computation*, vol. 176, no. 1, pp. 219–229, 2006.
- [9] Y. Gao and S. Jin, "A global optimization algorithm for sum of linear ratios problem," *Journal of Applied Mathematics*, vol. 2013, Article ID 276245, 7 pages, 2013.
- [10] Y. Pei and D. 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, no. 1-2, pp. 153–169, 2013.
- [11] R. Horst and H. Tuy, *Global Optimization, Deterministic Approaches*, Springer, Berlin, Germany, 1990.