Open Mathematics

Research Article

Yingfeng Zhao* and Ting Zhao

A reduced space branch and bound algorithm for a class of sum of ratios problems

https://doi.org/10.1515/math-2018-0049
Received February 8, 2018; accepted April 4, 2018.

Abstract: Sum of ratios problem occurs frequently in various areas of engineering practice and management science, but most solution methods for this kind of problem are often designed for determining local solutions. In this paper, we develop a reduced space branch and bound algorithm for globally solving sum of convex-concave ratios problem. By introducing some auxiliary variables, the initial problem is converted into an equivalent problem where the objective function is linear. Then the convex relaxation problem of the equivalent problem is established by relaxing auxiliary variables only in the outcome space. By integrating some acceleration and reduction techniques into branch and bound scheme, the presented global optimization algorithm is developed for solving these kind of problems. Convergence and optimality of the algorithm are presented and numerical examples taken from some recent literature and MINLPLib are carried out to validate the performance of the proposed algorithm.

Keywords: Sum of ratios problem, Global optimization, Branch and bound, Accelerating technique

MSC: 90C26, 90C30

1 Introduction

Fractional programming occurs frequently in a variety of economic, industrial and engineering problems [1]. It is one of the most topical and useful fields in nonconvex optimization and many intensive and systematical researches have been done on fractional programming since the seminal works by Charnes and Cooper [2, 3]. Sum of ratios problem SRP is a special case of optimization among fractional programming [4, 5], as a generalization of linear fractional programming which optimizes sum of linear ratios, the SRP has a broad range of applications. Included among those, are clustering problems [6], transportation planning [7], multi-stage stochastic shipping [8], finance and investment [9], and layered manufacturing problems [10, 11], to name but a few. The reader is referred to a survey [12] and a bibliography [13] to find many other applications. In this paper, we focus on the following sum of ratios problem:

$$\mathrm{SRP}:\begin{cases}\min f(x)=\sum\limits_{i=1}^{p}\delta_i\frac{\phi_i(x)}{\psi_i(x)}\\ s.t.\;\; h_j(x)\le 0, j=1,2,\ldots,m\\ \qquad x\in X=[\underline{x},\overline{x}],\end{cases}$$

where each coefficient δ_i is a real number, functions $\phi_i(x), -\psi_i(x), i=1,2,\ldots,p, h_j(x), j=1,2,\ldots,m$ are all convex (hence continuous). Furthermore, we assume that $\psi_i(x)\neq 0, \forall x\in X$. By the continuity of the denominators, we know that $\psi_i(x)$ must satisfy $\psi_i(x)>0$ or $\psi_i(x)<0$. Based on the discussion in [14] and

***Corresponding Author: Yingfeng Zhao:** School of Mathematical Science, Henan Institute of Science and Technology, Xinxiang 453003, China, E-mail: zhaoyingfeng6886@163.com
Ting Zhao: School of Mathematical Science, Henan Institute of Science and Technology, Xinxiang 453003, China

for the sake of simplicity, we only need to consider a special case of the SRP, that is, the numerator and the denominator of each ratio in the objective function of the SRP satisfies the following condition:

$$\phi_i(x) \geq 0, \psi_i(x) > 0, \ \forall x \in X, i = 1, 2, \cdots, p. \tag{1}$$

The SRP has attracted the interest of quite a lot of researchers and practitioners for many years which is at least in part due to the difficulty associated with the existence of multiple local solution that are not optimally global. Actually, Charnes and Cooper have proved that the optimization of a single linear ratio is equivalent to a linear program and hence it can be solved in polynomial time [2, 15]. But this is not true for the SRP where the objective function is a sum of p ($p \geq 2$) nonlinear (even linear) ratios, due to some inherent difficulties, there are many theoretical and computational challenge for finding the global optimizer of the SRP. During the past several years, some feasible algorithms have already been proposed for the SRP and its special forms, for instance, Konno et al. presented a parametric simplex method and an efficient heuristic algorithm for globally solving the sum of linear fractional problems and its special case [17, 18], but their algorithm can only solve sum of linear ratios, and the problem must have three ratios. Falk and Palocasy put forward an approach based on the image space analysis for globally solving sum of affine ratios problem [19], wherein they identify classes of nonconvex problems involving either sums or products of ratios of linear terms which may be treated by analysis in a transformed space. In each class, the image space is defined by a mapping which associates a new variable with each original ratio of linear terms. In the image space, optimization is easy in certain directions, and the overall solution may be realized by sequentially optimizing in these directions, this algorithm has good performance, but the problem they considered can only have linear constraints; Pei and Zhu present a branch and bound algorithm by converting it into a D.C programming [20], their algorithm performs well when the number of variables is not so big. In addition to this, Shen and Wang developed two kinds of branch-reduction-bound algorithms for sum of linear ratios problem [21, 22], both of these branch and bound algorithms branch in the variable space, with the increase of the number of variables, the performance of the algorithm will decline sharply; Jiao and Liu presented a practical outcome space branch and bound algorithm for globally maximizing sum of linear ratios problem [14], this algorithm can effectively solve the sum of linear ratios problem with quite a lot of variables, but the branch operation occurs in the outcome space of the reciprocal of the denominator. Despite these various contributions, however, there is still no decisive method for globally solving general sum of ratios problem and thus efficient solution method for SRP is still an open issue.

In this study, we will present a reduced space branch and bound algorithm with practical accelerating techniques according to some properties (concavity, convexity and continuity) of the objective and constraint functions in SRP. The attractive properties of this algorithm is mainly embodied in the following three aspects. First, the problem we considered is more general and extensive than that in most of the above literatures. Second, the relaxation operation we used is quite concise and practical, the adapted subdivision and range reduction technique carried out in the outcome space can sharply reduce the number of nodes in the branching tree so as the execution efficiency of the algorithm is significantly improved. Finally, the global convergence property is proved and some numerical experiment and a random test is performed to illustrate the feasibility and robust property.

The remainder of this paper is organized in the following way. The next section shows how to construct the equivalent problem EP and the convex relaxation programming of the EP according to the concavity and convexity of the objective and constraint functions in SRP. The condensing, branching and bounding operations of the new algorithm are established in Section 3. The detailed statement and the global convergence property of the presented algorithm is put forward in Section 4. Section 5 is devoted to a report of computational comparison between our algorithm and some of the other algorithms that exist in the literature. Some concluding remarks are proposed in the last section.

2 Equivalent problem and relaxation programming

In this section, we will first transform the problem SRP into an equivalent problemEP by associating each ratio in the objective function in SRP with an additional variable which we call the outcome variable, and then our focus will be shifted to find the global optimal solution of the EP. By utilizing the special structure of the EP, a concise convex relaxation programming for the EP will be introduced with which we only need to branch in a reduced outcome space and at the same time, a new upper bound and lower bound of the optimal value will be obtained simultaneously at each iteration, so as to greatly reduce the workload of the calculation.

2.1 Equivalent problem

To solve the problem, we will first transform the SRP into an equivalent problem EP, where the objective function is linear and the constraint functions possess a special structure which is beneficial for constructing convex relaxing programming problems. To explain how such a reformulation is possible, we first introduce p auxiliary variables $t_i, i = 1, 2\cdots, p$, and for definiteness and without loss of generality, we assume that
i) $\delta_i > 0, i = 1, 2\cdots, p$, when $\phi_i(x)$ is convex and $\psi_i(x)$ is concave, $i = 1, 2\cdots, p$.
ii) $\delta_i > 0, i = 1, 2\cdots, T$; $\delta_i < 0, i = T+1, T+2\cdots, p$, when $\phi_i(x)$ and $\psi_i(x)$ are linear, $i = 1, 2\cdots, p$.
Then denote

$$d_i = \min_{x\in X} \psi_i(x);\ l_i^0 = \frac{\min\limits_{x\in X} \phi_i(x)}{\max\limits_{x\in X} \psi_i(x)},\ \ u_i^0 = \frac{\max\limits_{x\in X} \phi_i(x)}{\min\limits_{x\in X} \psi_i(x)}, i = 1, 2, \cdots, p,$$

note that we can obtain the values of d_i, l_i^0 and u_i^0 easily by utilizing the convexity and concavity of the numerators and denominators and clearly we know that $0 \le l_i^0 \le u_i^0$.

Next, we consider the following equivalent problem EP:

$$\text{EP}: \begin{cases} \min\ g(t) = \sum\limits_{i=1}^{p} \delta_i t_i \\ s.t.\ \ c_i(t) = \phi_i(x) - t_i(\psi_i(x) - d_i) + t_i d_i \le 0, i = 1, 2, \ldots, T, \\ \qquad c_i(t) = \phi_i(x) - t_i(\psi_i(x) - d_i) + t_i d_i \ge 0, i = T+1, T+2, \ldots, p, \\ \qquad h_j(x) \le 0, j = 1, 2, \ldots, m, \\ \qquad x \in X, t \in D^0. \end{cases}$$

where $D^0 = \{t \in R^p \mid l_i^0 \le t_i \le u_i^0, i = 1, 2, \ldots, p\}$ is called an outcome space corresponding to the feasible region of SRP, and soon we will show that problems SRP and EP are equivalent in the sense of the following theorem.

Theorem 2.1. *$x^* \in R^n$ is a global optimal solution for the SRP if and only if $(x^*, t^*) \in R^{n+p}$ is a global optimal solution for the EP, where $t_i^* = \frac{\phi_i(x^*)}{\psi_i(x^*)}, i = 1, 2, \ldots, p$.*

Proof. Assume $x^* \in R^n$ is a global optimal solution for the SRP, let $t_i^* = \frac{\phi_i(x^*)}{\psi_i(x^*)}, i = 1, 2, \ldots, p$, then we have $(x^*, t^*) \in R^{n+p}$ is a feasible solution for the EP, according the optimality of x^* in the SRP, we know that, for each $x \in F$

$$f(x) = \sum_{i=1}^{p} \delta_i \frac{\phi_i(x)}{\psi_i(x)} \ge f(x^*) = \sum_{i=1}^{p} \delta_i \frac{\phi_i(x^*)}{\psi_i(x^*)} = \sum_{i=1}^{p} \delta_i t_i^* = g(t^*). \tag{2}$$

In addition, if (x, t) is a feasible solution to the EP, we can obtain

$$g(t) = \sum_{i=1}^{T} \delta_i t_i + \sum_{i=T+1}^{p} \delta_i t_i \ge \sum_{i=1}^{T} \delta_i \frac{\phi_i(x)}{\psi_i(x)} + \sum_{i=T+1}^{p} \delta_i \frac{\phi_i(x)}{\psi_i(x)} = f(x). \tag{3}$$

From the above conclusion (2) and (3), we know that $(x^*, t^*) \in R^{n+p}$ is a global optimal solution for the EP. Conversely, if $(x^*, t^*) \in R^{n+p}$ is a global optimal solution for the EP, we first prove that

$$t_i^* = \frac{\phi_i(x^*)}{\psi_i(x^*)}, i = 1, 2, \ldots, p.$$

If otherwise, according the feasibility of (x^*, t^*), one of the following two conclusions

$$t_i^* > \frac{\phi_i(x^*)}{\psi_i(x^*)}, \text{ for some } i \in \{1, 2, \cdots, T\}, \; t_j^* \geq \frac{\phi_j(x^*)}{\psi_j(x^*)}, \text{ for all } j \neq i$$

or

$$t_i^* < \frac{\phi_i(x^*)}{\psi_i(x^*)}, \text{ for some } i \in \{T+1, T+2, \cdots, p\}, \; t_j^* \leq \frac{\phi_j(x^*)}{\psi_j(x^*)}, \text{ for all } j \neq i$$

must hold. Let $\bar{t}_i = \frac{\phi_i(x^*)}{\psi_i(x^*)}, i = 1, 2, \ldots, p$, then we know $(x^*, \bar{t})$ is a feasible solution and $g(\bar{t}) = \sum_{i=1}^{T} \delta_i \bar{t}_i + \sum_{i=T+1}^{p} \delta_i \bar{t}_i < \sum_{i=1}^{T} \delta_i t_i^* + \sum_{i=T+1}^{p} \delta_i t_i^* = g(t^*)$, it is a contradiction to the optimality of (x^*, t^*), hence we have

$$t_i^* = \frac{\phi_i(x^*)}{\psi_i(x^*)}, i = 1, 2, \ldots, p.$$

Furthermore, $\forall x \in F$, let $t_i = \frac{\phi_i(x)}{\psi_i(x)}, i = 1, 2, \ldots, p$, then we have $(x, t) \in R^{n+p}$ is a feasible solution for the EP, and by the optimality of (x^*, t^*), we have

$$f(x^*) = \sum_{i=1}^{p} \delta_i \frac{\phi_i(x^*)}{\psi_i(x^*)} = \sum_{i=1}^{p} \delta_i t_i^* \leq \sum_{i=1}^{p} \delta_i t_i = \sum_{i=1}^{p} \delta_i \frac{\phi_i(x)}{\psi_i(x)} = f(x),$$

that is to say, x^* is the global optimal solution of SRP, and this completes the proof. □

For solving problem SRP, according to the conclusion of Theorem 2.1, we only need to consider how to solve problem EP. To this end, we will make full use of the structure of the EP to establish the convex or linear relaxation problem of the EP for designing the presented algorithm. To keep things simple, we only consider the linear situation (case ii)), it can be easily extended to the nonlinear circumstances that satisfy condition i).

2.2 Relaxation technique

In this part, we concentrate on how to construct the linear relaxation programming problem of the EP on assumption that all functions appeared in the SRP are linear. Note that the objective function of problem EP is already a linear function, so we only need to consider the bilinear constraints. For simplicity, we denote D as the rectangle region generated by the branching operation, where $D = D_1 \times D_2 \times \cdots \times D_p$ with $D_i = \{t_i \in R | l_i \leq t_i \leq u_i\}$, and F is a subset of the feasible region which appears in the branch operation, then we can put forward an approach for generating a linear underestimating function of the constraint function for problem EP, which is given by the following Theorem 2.1.

Theorem 2.2. *For any $(x, t) \in F \times D$, denote*

$$\tilde{c}_i(t) = (n^i)^T x - \sum_{j=1}^{n} \theta_{ij} d_j^i x_j + \beta_i + t_i \gamma_i, i = 1, 2, \cdots, p, \tag{4}$$

where

$$\theta_{ij} = \begin{cases} u_i, d_j \geq 0 \\ l_i, d_j < 0 \end{cases}, i = 1, 2 \cdots T \; ; \; \theta_{ij} = \begin{cases} l_i, d_j \geq 0 \\ u_i, d_j < 0 \end{cases}, i = T+1, T+2 \cdots p,$$

then we have:
(i) The function $\tilde{c}_i(t)$ is a lower bounding function for $c_i(t)$ over the region $F \times D$.
(ii) Function $\tilde{c}_i(t)$ will approximate each $c_i(t), (i = 1, 2, \cdots, p)$ as $|u_i - l_i| \to 0$ that is $|\tilde{c}_i(t) - c_i(t)| \to 0$, as $|u_i - l_i| \to 0$.

Proof. (i) For any $(x, t) \in F \times D$, by the definition of $c_i(t)$ and $\tilde{c}_i(t)$, we have

$$\begin{aligned} c_i(t) &= \phi_i(x) - t_i\psi_i(x) = (n^i)^T x + \beta_i - t_i(d^i x - \gamma_i) \\ &= (n^i)^T x - \sum_{j=1}^{n} t_i d_j^i x_j + t_i\gamma_i + \beta_i \\ &\geq (n^i)^T x - \sum_{j=1}^{n} \theta_{ij} d_j^i x_j + \beta_i + t_i\gamma_i = \tilde{c}_i(t). \end{aligned}$$

So the conclusion (i) holds.

(ii) For any $(x, t) \in F \times D$, by the definition of $c_i(t)$ and $\tilde{c}_i(t)$, we have

$$\begin{aligned} |\tilde{c}_i(t) - c_i(t)| &= |n^i x - \sum_{j=1}^{n} t_i d_j^i x_j + t_i\gamma_i + \beta_i - (n^i x - \sum_{j=1}^{n} \theta_{ij} d_j^i x_j + \beta_i + t_i\gamma_i)| \\ &= |\sum_{j=1}^{n} \theta_{ij} d_j^i x_j - \sum_{j=1}^{n} t_i d_j^i x_j| = |\sum_{j=1}^{n} (\theta_{ij} - t_i) d_j^i x_j| \\ &\leq |\sum_{j=1}^{n} d_j^i x_j||u_i - l_i| \leq M_i|u_i - l_i|. \end{aligned} \tag{5}$$

Where M_i is an upper bound of $|\sum_{j=1}^{n} d_j^i x_j|$, it existence can be easily obtained by the continuity of affine function $\sum_{j=1}^{n} d_j^i x_j$ over a compact region. From conclusion (5), it is easy to see that

$$|\tilde{c}_i(t) - c_i(t)| \to 0, \text{ as } |u_i - l_i| \to 0.$$

Thus the proof is complete. □

Therefore, according to the above discussion, we can obtain the linear relaxation programming problem $\mathrm{REP_D}$ corresponding to the outcome space D of $\mathrm{EP_D}$ as follows:

$$\mathrm{REP_D}: \begin{cases} \min\ g(t) = \sum_{i=1}^{p} \delta_i t_i \\ s.t.\ \ -\tilde{c}_i(t) \geq 0, i = 1, 2, \ldots, T, \\ \qquad \tilde{c}_i(t) \geq 0, i = T + 1, T + 2, \cdots, p, \\ \qquad x \in F, t \in D, \end{cases}$$

where $\tilde{c}_i(t)$ is defined by (5), and from now on, we will use the symbol $\mathrm{EP_D}$ to express the problem EP corresponding to the outcome space D, and in the rest of this paper, any symbol similar to this should be understood in the same meaning.

Based on the construct process of the $\mathrm{REP_D}$, it is not hard to find that every feasible solution for $\mathrm{EP_D}$ is also a feasible solution of the $\mathrm{REP_D}$, but its optimal value is not less than that of the $\mathrm{REP_D}$, thus the $\mathrm{REP_D}$ can provide a valid lower bound for the optimal value of problem $\mathrm{EP_D}$ and problem $\mathrm{REP_D}$ will approximate the $\mathrm{EP_D}$ as $\max_{1 \leq i \leq p} |u_i - l_i| \to 0$ which is indicated by Theorem 2.2.

3 Key operations for algorithm design

To present the reduced space branch and bound algorithm for solving the SRP, we will describe three fundamental operations: branching, condensing and bounding, in this section.

3.1 Branching operation

In this paper, we adopt the so-called adapted partition technique to subdivide the initial box D^0 into subboxes. The adapted partition operation performs in a reduced outcome space associated with problem EP other than $n-$dimensional variable space, this is the place where is different from the general branch and bound algorithm performed in variable space. For any subset $D = \{t \in R^p | l_i \le t_i \le u_i\} \subset D^0$, the specific division procedure is given as follows.

Partition regulation

1. Let $r \in \operatorname{argmax}\{u_i - l_i | i = 1, 2, \cdots, p\}$.
2. Let $\theta_r = (1 - \alpha) l_r + \alpha u_r$.
3. Subdivide D_r into two intervals D_r^1 and D_r^2, where $D_r^1 = [l_r, \theta_r]$ and $D_r^2 = [\theta_r, u_r]$, then let

$$D' = D_1 \times D_2 \times \cdots \times D_{r-1} \times D_r^1 \times D_{r+1} \times \cdots \times D_p,$$

and

$$D'' = D_1 \times D_2 \times \cdots \times D_{r-1} \times D_r^2 \times D_{r+1} \times \cdots \times D_p.$$

Thus, region D is divided into two new hyper-rectangles D' and D''.

It can be seen from the above partition regulation that only the $p-$dimensional outcome space is partitioned in the algorithm, the $n-$dimensional variable space was never divided, this is just the place where our algorithm is different from the usual branch and bound algorithm, and immediately, we will see that this operation will make the algorithm quite efficient for special scaled problem where the number of the ratios in the objective function is far less than that of the variables.

3.2 Condensing and bounding technique

For any rectangle $D^k \subset D^0$ generated by the branching operation in the k-th iteration, the condensing operation consists in reducing the current partition still of interest by incising the part which does not contain the global optimal solution for problem $\mathrm{SARP}_{\mathrm{D}^0}$; The bounding operation aims at estimating an upper and (or) lower bound of the optimal objective value of the EP and removing the subregion which doesn't have further research value.

In the k-th iteration, first, we solve the linear relaxation programming problem $\mathrm{REP}_{\mathrm{D}^k}$, assume the optimal solution is $(\overline{x}^k, \overline{t}^k)$, then let $\widetilde{t}_i^k = \frac{\phi_i(\overline{x}^k)}{\psi_i(\overline{x}^k)}$, we can obtain a feasible solution $(\overline{x}^k, \widetilde{t}^k)$ for problem $(\mathrm{EP}_{\mathrm{D}^0})$, and of course, the objective value of $(\overline{x}^k, \widetilde{t}^k)$ is an upper bound for the optimal value of the $(\mathrm{EP}_{\mathrm{D}^0})$; Further more, the optimal value of the $\mathrm{REP}_{\mathrm{D}^k}$ is a lower bound of the objective value of $(\mathrm{EP}_{\mathrm{D}^k})$, and the smallest optimal value of all subproblem in the k-th iteration is a lower bound for the optimal value of the $(\mathrm{EP}_{\mathrm{D}^0})$. Assume that $\overline{f}$ is the best upper bound of the optimum of the $\mathrm{REP}_{\mathrm{D}^0}$ known so far, then the condensing technique can be described in the form of the following theorem:

Theorem 3.1. *For any sub-region $D^k \subset D^0$, assume f_{min}^k is the optimal value of problem $\mathrm{REP}_{\mathrm{D}^k}$, then the following two conclusions hold:*
(i) If $f_{min}^k > \overline{f}$, then D^k doesn't contain the optimal solution of problem $(\mathrm{EP}_{\mathrm{D}^0})$, so it can be removed.
(ii) If $f_{min}^k \le \overline{f}$, then for each $j \in \{1, 2, \cdots, T\}$, the region $\overline{D}^k \bigcap D^k$ can be incised; For each $j \in \{T+1, T+2, \cdots, p\}$, the region $\overline{\overline{D}}^k \bigcap D^k$ can be incised, where

$$\overline{D}^k = D_1^k \times D_2^k \times \cdots \times D_{j-1}^k \times \overline{D}_j^k \times D_{j+1}^k \times \cdots \times D_T^k \times D_{T+1}^k \times \cdots \times D_p^k,$$

and

$$\overline{\overline{D}}^k = D_1^k \times D_2^k \times \cdots \times D_T^k \times D_{T+1}^k \times \cdots \times D_{j-1}^k \times \overline{\overline{D}}_j^k \times D_{j+1}^k \times \cdots \times D_p^k,$$

with

$$r_j^k = \begin{cases} \frac{1}{\delta_j}\left(\overline{f} - f_{min}^k\right) + u_j^k, j = 1, 2, \cdots, T, \\ \frac{1}{\delta_j}\left(\overline{f} - f_{min}^k\right) + l_j^k, j = T+1, T+2, \cdots, p, \end{cases}$$

and

$$\overline{D}_j^k = [l_j^k, r_j^k] \cap D_j^k,\ \overline{\overline{D}}_j^k = [r_j^k, u_j^k] \cap D_j^k.$$

Proof. (i)The conclusion is obvious, here is omitted.

(ii) For simplicity's sake, we denote

$$\overline{M}^k = \left\{ (x,t) \in F \times \bar{D}^k \middle| \begin{array}{l} \tilde{c}_i(t) \le 0, i = 1, 2, \ldots, T \\ \tilde{c}_i(t) \ge 0, i = T+1, T+2, \cdots, p \end{array} \right\},$$

and

$$\overline{\overline{M}}^k = \left\{ (x,t) \in F \times \bar{\bar{D}}^k \middle| \begin{array}{l} \tilde{c}_i(t) \le 0, i = 1, 2, \ldots, T \\ \tilde{c}_i(t) \ge 0, i = T+1, T+2, \cdots, p \end{array} \right\},$$

Since $f_{min}^k \le \overline{f}$, then for each $j \in \{1, 2, \cdots, T\}$, and $(x,t) \in \overline{M}^k$, we have

$$\min_{(x,t)\in\overline{M}^k} \sum_{i=1}^p \delta_i t_i \ge f_{min}^k + \min_{(x,t)\in\overline{M}^k} \delta_j t_j - \min_{(x,t)\in\overline{M}^k} \delta_j t_j > f_{min}^k + \delta_j r_j^k - \delta_j u_j^k = \overline{f},$$

therefore, $\overline{D}^k$ doesn't contain the global optimal solution for the $(\mathrm{EP}_{\mathrm{D}^0})$, and it can be incised.

In the same way, when $j \in \{T+1, T+2, \cdots, p\}$, we have

$$\min_{(x,t)\in\overline{M}^k} \sum_{i=1}^p \delta_i t_i \ge f_{min}^k + \mathop{min}_{(x,t)\in\overline{\overline{M}}^k} \delta_j t_j - \min_{(x,t)\in\overline{\overline{M}}^k} \delta_j t_j > f_{min}^k + \delta_j r_j^k - \delta_j l_j^k = \overline{f},$$

similarly, $\overline{\overline{D}}^k$ doesn't contain the global optimal solution of the $(\mathrm{EP}_{\mathrm{D}^0})$, and it will be incised in the algorithm. □

By Theorem 3.1, the condensing operation can cut away a large part of current region in which the optimal solution doesn't exist, so the rapid growth of the branching node can be suppressed from iteration to iteration. Additionally, unlike a normal branch and bound algorithm, the branching method used in this study can adjust the ratio of the partitions measurement by adopting different ratios, and thus the convergent speed of the algorithm can be enhanced.

4 Algorithm statement and convergence analysis

Based upon the above results and technique, the basic steps of the reduced space branch and bound algorithm associated with efficient accelerating techniques for globally solving the SRP will be summarized in this section.

4.1 Algorithm statement

By integrating the condensing technique and partition skills into the reduced space branch and bound scheme, the presented algorithm for the SRP can be described as follows.

Step 0 (Initialization). Set the convergence precision $\epsilon \ge 0$, iteration counter $k = 0$ and the partition ratio $\alpha \in [0, 1]$. Compute the values of l_i^0, u_i^0, for each $i = 1, 2, \cdots, p$, then detemine the optimal solution (x^0, t^0) and optimal value f_{min} by solving the linear relaxation programming problem $\mathrm{REP}_{\mathrm{D}^0}$. Let

$$\underline{f} = f_{min}, t_i^* = \frac{\phi_i(x^*)}{\psi_i(x^*)}, i = 1, 2, \cdots, p, x^* = x^0,$$

clearly, (x^*, t^*) is a feasible solution for the $(\mathrm{EP}_{\mathrm{D}^0})$. Let $\overline{f} = g(t^*) = \sum_{i=1}^{p} \delta_i t_i^*$, if $\overline{f} - \underline{f} \leq \epsilon$, stop, and $x^*, \overline{f}$ are the optimal solution and the optimal vale of the SRP, respectively; otherwise, set $F = \phi$, $k = 1$, $D^1 = D^0$, the set of all partitions still of interest $\Theta_k = \{D^1\}$, then turn to step 1.

Step 1 (Condensing). For each rectangle $D^k \subset D$, incising the invalid part by the condensing technique described in section 3.2, substitute D^k with the remaining partition.

Step 2 (Branching). Subdivide region D^k into two new regions D^{k1} and D^{k2} according to the ratio partition rule, express the collection of new partitions as $\widetilde{D}^k$.

Step 3 (Bounding). Obtain optimal solutions $(x^{k\nu}, t^{k\nu})$ and optimal values $f_{max}^{k\nu}$, with $\nu \in \{1, 2\}$, respectively, by solving problems $\mathrm{REP}_{\mathrm{D}^k}$. Then let

$$\overline{t}_i^{*\nu} = \frac{\phi_i(x^{k\nu})}{\psi_i(x^{k\nu})}, i = 1, 2, \cdots, p, \nu \in \{1, 2\},$$

and update the upper bound by setting $\overline{f} = \min\{\overline{f}, g(\overline{t}^{*\nu})\}$, and let x^* be the feasible solution with the best objective value currently known. If $f_{min}^{k\nu} > \overline{f}$, delete the node associated with $D^{k\nu}$ from Θ^k, if $\Theta_k = \phi$, stop, $x^*, \overline{f}$ are the optimal solution and the optimal vale of the SRP, respectively; else, update the lower bound by setting $f_{min} = \min\{f_{min}^{k\nu}\}$.

Step 4 (Optimality test). If $\overline{f} - \underline{f} \leq \epsilon$, the algorithm can stop, at the same time, we can conclude that x^* and $\overline{f}$ are the $\epsilon-$global minimizer and minimum for the SRP, respectively. Otherwise, let $k = k + 1$ and return to step 1.

4.2 Convergence analysis

In this section, we illustrate the convergence property of the algorithm by the following theorem.

Theorem 4.1. *The reduced space branch and bound algorithm described above either terminates within finitely many iterations and yield an $\epsilon-$global solution of the SRP, or generates an infinite sequence of feasible solutions with an accumulation as an $\epsilon-$global solution for the SRP.*

Proof. If the algorithm terminates at the k-th iteration, upon termination criteria, it follows that $\overline{f} - \underline{f} \leq \epsilon$. From step 0 and step 3 in the algorithm, a feasible solution x^k can be found to satisfy the following relation $f(x^k) - \underline{f} \leq \epsilon$. At the same time, we have $\underline{f} \leq f_{opt} \leq f(x^k)$, where f_{opt} is the optimal value for the $(\mathrm{EP}_{\mathrm{D}^0})$. Thus, taken the above relations together, it implies that

$$f_{opt} \leq f(x^k) \leq \underline{f} + \epsilon \leq f_{opt} + \epsilon,$$

we can conclude that x^k is the optimal solution for the $(\mathrm{EP}_{\mathrm{D}^0})$, and of course also for the SRP.

If the algorithm is infinite and via solving the $\mathrm{REP}_{\mathrm{D}^k}$, generates an infinite feasible solution sequence $\{(x^k, t^k)\}$. Let $\overline{t}_i^k = \frac{\phi_i(x^k)}{\psi_i(x^k)}$, then $\{(x^k, \overline{t}^k)\}$ is a feasible solution sequence for the $(\mathrm{EP}_{\mathrm{D}^0})$. Since the sequence $\{x^k\}$ is bounded, it must have accumulations assuming $\lim_{k\to\infty} x^k = x^*$ without loss of generality. On the other hand, we get

$$\lim_{k\to\infty} \overline{t}_i^k = \lim_{k\to\infty} \frac{\phi_i(x^k)}{\psi_i(x^k)} = \frac{\phi_i(x^*)}{\psi_i(x^*)}, \tag{6}$$

by the continuity of $\phi_i(x)$ and $\psi_i(x)$.

Also, according to the branching regulation described before, we know that

$$\lim_{k\to\infty} l_i^k = \lim_{k\to\infty} u_i^k = t_i^*, \tag{7}$$

what'more, note that $l_i^k \le \frac{\phi_i(x^k)}{\psi_i(x^k)} \le u_i^k$, we conclude that

$$t_i^* = \lim_{k\to\infty} \frac{\phi_i(x^k)}{\psi_i(x^k)} = \frac{\phi_i(x^*)}{\psi_i(x^*)} = \lim_{k\to\infty} \bar{t}^k$$

by (6) and (7), therefore $(x^*, \bar{t}^*)$ is also a feasible solution for the $(\mathrm{EP}_{\mathrm{D}^0})$. Further more, since the lower bound sequence $\underline{f}^k$ for the optimal value is increasing and lower bounded by the optimal value f_{opt} in the algorithm, so combining the continuity of $g(t)$, we have

$$\lim_{k\to\infty} \underline{f}^k = g(t^*) \le f_{opt} \le \lim_{k\to\infty} g(\bar{t}^k) = g(t^*).$$

That is, (x^*, t^*) is an optimal solution for the $(\mathrm{EP}_{\mathrm{D}^0})$, and of course x^* is an optimal solution for the SRP according to the equality of problems SRP and $(\mathrm{EP}_{\mathrm{D}^0})$, therefore completing the proof. □

5 Numerical experiments

To test the proposed algorithm in efficiency and solution quality, we performed some computational examples on a personal computer containing an Intel Core i5 processor of 2.40 GHz and 4GB of RAM. The code base is written in Matlab 2014a and interfaces LINPROG for the linear relaxation subproblems and CVX for the convex relaxation subproblems.

We consider some numerical examples in recent literatures [14, 20–26], and a randomly generated test problem to verify the performance of the algorithm. The numerical test and results are listed as follows.

Example 5.1 ([23]).

$$\begin{cases} \max \frac{-x_1+2x_2+2}{3x_1-4x_2+5} + \frac{4x_1-3x_2+4}{-2x_1+x_2+3} \\ \text{s.t.} \quad x_1 + x_2 \le 1.50, x_1 - x_2 \le 0, 0 \le x_1 \le 1, 0 \le x_2 \le 1. \end{cases}$$

Example 5.2 ([14, 22, 23]).

$$\begin{cases} \max \frac{4x_1+3x_2+3x_3+50}{3x_2+3x_3+50} + \frac{3x_1+4x_2+50}{4x_1+4x_2+5x_3+50} + \frac{x_1+2x_2+5x_3+50}{x_1+5x_2+5x_3+50} + \frac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50} \\ \text{s.t.} \quad 2x_1 + x_2 + 5x_3 \le 10, x_1 + 6x_2 + 3x_3 \le 10, \\ \qquad 5x_1 + 9x_2 + 2x_3 \le 10, 9x_1 + 7x_2 + 3x_3 \le 10, x_1 \ge 0, x_2 \ge 0, x_3 \ge 0. \end{cases}$$

Example 5.3 ([14]).

$$\begin{cases} \max 0.9 \times \frac{-x_1+2x_2+2}{3x_1-4x_2+5} - 0.1 \times \frac{4x_1-3x_2+4}{-2x_1+x_2+3} \\ \text{s.t.} \quad x_1 + x_2 \le 1.50, x_1 - x_2 \le 0, 0 \le x_1 \le 1, 0 \le x_2 \le 1. \end{cases}$$

Example 5.4 ([21]).

$$\begin{cases} \max \frac{3x_1+4x_2+50}{3x_1+5x_2+4x_3+50} - \frac{3x_1+5x_2+3x_3+50}{5x_1+5x_2+4x_3+50} - \frac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50} - \frac{4x_1+3x_2+3x_3+50}{3x_2+3x_3+50} \\ \text{s.t.} \quad 6x_1 + 3x_2 + 3x_3 \le 10, 10x_1 + 3x_2 + 8x_3 \le 10, x_1 \ge 0, x_2 \ge 0, x_3 \ge 0. \end{cases}$$

Example 5.5 ([14, 24]).

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

Example 5.6 ([20, 23]).

$$\begin{cases} \max \frac{3x_1+5x_2+3x_3+50}{3x_1+4x_2+5x_3+50} + \frac{3x_1+4x_2+50}{4x_1+3x_2+2x_3+50} + \frac{4x_1+2x_2+4x_3+50}{5x_1+4x_2+3x_3+50} \\ \text{s.t.} \quad 6x_1 + 3x_2 + 3x_3 \le 10, 10x_1 + 3x_2 + 8x_3 \le 10, x_1 \ge 0, x_2 \ge 0, x_3 \ge 0. \end{cases}$$

Example 5.7 ([14, 24]).

$$\begin{cases} \max \sum\limits_{i=1}^{5} \frac{c^i x + r_i}{d^i x + s_i} \\ \text{s.t.} \quad Ax \le b, \\ \qquad\quad x \ge 0. \end{cases}$$

where

$b = (15.7, 31.8, -36.4, 38.5, 40.3, 10.0, 89.8, 5.8, 2.7, -16.3, -14.6, -72.7, 57.7, -34.5, 69.1)^T$

$$A = \begin{pmatrix} -1.8 & -2.2 & 0.8 & 4.1 & 3.8 & -2.3 & -0.8 & 2.5 & -1.6 & 0.2 & -4.5 & -1.8 \\ 4.6 & -2.0 & 1.4 & 3.2 & -4.2 & -3.3 & 1.9 & 0.7 & 0.8 & -4.4 & 4.4 & 2.0 \\ 3.7 & -2.8 & -3.2 & -2.0 & -3.7 & -3.3 & 3.5 & -0.7 & 1.5 & -3.1 & 4.5 & -1.1 \\ -0.6 & -0.6 & -2.5 & 4.1 & 0.6 & 3.3 & 2.8 & -0.1 & 4.1 & -3.2 & -1.2 & -4.3 \\ 1.8 & -1.6 & -4.5 & -1.3 & 4.6 & 3.3 & 4.2 & -1.2 & 1.9 & 2.4 & 3.4 & -2.9 \\ -0.5 & -4.1 & 1.7 & 3.9 & -0.1 & -3.9 & -1.5 & 1.6 & 2.3 & -2.3 & -3.2 & 3.9 \\ 0.3 & 1.7 & 1.3 & 4.7 & 0.9 & 3.9 & -0.5 & -1.2 & 3.8 & 0.6 & -0.2 & -1.5 \\ 0.5 & -4.2 & 3.6 & -0.6 & -4.8 & 1.5 & -0.3 & 0.6 & -3.6 & 0.2 & 3.8 & -2.8 \\ 0.1 & 3.3 & -4.3 & 2.4 & 4.1 & 1.7 & 1.0 & -3.3 & 4.4 & -3.7 & -1.1 & -1.4 \\ -0.6 & 2.2 & 2.5 & 1.3 & -4.3 & -2.9 & -4.1 & 2.7 & -0.8 & -2.9 & 3.5 & 1.2 \\ 4.3 & 1.9 & -4.0 & -2.6 & 1.8 & 2.5 & 0.6 & 1.3 & -4.3 & -2.3 & 4.1 & -1.1 \\ 0.0 & 0.4 & -4.5 & -4.4 & 1.2 & -3.8 & -1.9 & 1.2 & 3.0 & -1.1 & -0.2 & 2.5 \\ -0.1 & -1.7 & 2.9 & 1.5 & 4.7 & -0.3 & 4.2 & -4.4 & -3.9 & 4.4 & 4.7 & -1.0 \\ -3.8 & 1.4 & -4.7 & 1.9 & 3.8 & 3.5 & 1.5 & 2.3 & -3.7 & -4.2 & 2.7 & -0.1 \\ 0.2 & -0.1 & 4.9 & -0.9 & 0.1 & 4.3 & 1.6 & 2.6 & 1.5 & -1.0 & 0.8 & 1.6 \end{pmatrix}$$

$$\begin{aligned} c^1 &= (0.0, -0.1, -0.3, 0.3, 0.5, 0.5, -0.8, 0.4, -0.4, 0.2, 0.2, -0.1), & r_1 &= 14.6 \\ c^2 &= (0.2, 0.5, 0.0, 0.4, 0.1, -0.6, -0.1, -0.2, -0.2, 0.1, 0.2, 0.3), & r_2 &= 7.1 \\ c^3 &= (-0.1, 0.3, 0.0, 0.1 - 0.1, 0.0, 0.3, -0.2, 0.0, 0.3, 0.5, 0.3), & r_3 &= 1.7 \\ c^4 &= (-0.1, 0.5, 0.1, 0.1 - 0.2, -0.5, 0.6, 0.7, 0.5, 0.7, -0.1, 0.1), & r_4 &= 4.0 \\ c^5 &= (0.7, -0.5, 0.1, 0.2 - 0.1, -0.3, 0.0, -0.1, -0.2, 0.6, 0.5, -0.2), & r_5 &= 6.8 \\ d^1 &= (-0.3, -0.1, -0.1, -0.1, 0.1, 0.4, 0.2, -0.2, 0.4, 0.2, -0.4, 0.3), & s_1 &= 14.2 \\ d^2 &= (0.0, 0.1, -0.1, 0.3, 0.3 - 0.2, 0.3, 0.0, -0.4, 0.5, -0.3, 0.1), & s_2 &= 1.7 \\ d^3 &= (0.8, -0.4, 0.7, -0.4, -0.4, 0.5, -0.2, -0.8, 0.5, 0.6, -0.2, 0.6), & s_3 &= 8.1 \\ d^4 &= (0.0, 0.6, -0.3, 0.3, 0.0, 0.2, 0.3, -0.6, -0.2, -0.5, 0.8, -0.5), & s_4 &= 26.9 \\ d^5 &= (0.4, 0.2, -0.2, 0.9, 0.5, -0.1, 0.3, -0.8, -0.2, 0.6, -0.2, -0.4), & s_5 &= 3.7 \end{aligned}$$

Example 5.8 ([25]).

$$\begin{cases} \min \ x_1 + x_2 + x_3 \\ \text{s.t.} \quad \frac{833.33252x_4}{x_1x_6} + \frac{100}{x_6} \le 1 \\ \qquad\ \ \frac{1250x_5 - 1250x_4}{x_2x_7} + \frac{x_4}{x_7} \le 1 \\ \qquad\ \ \frac{1250000 - 2500x5}{x_3x_8} + \frac{x_5}{x_8} \le 1 \\ \qquad\ \ 0.0025x_4 + 0.0025x_6 \le 1 \\ \qquad\ \ -0.0025x_4 + 0.0025x_5 + 0.0025x_7 \le 1 \\ \qquad\ \ 0.01x_8 - 0.01x_5 \le 1 \\ \qquad\ \ 100 \le x_1 \le 10000 \\ \qquad\ \ 1000 \le x_2, x_3 \le 10000 \\ \qquad\ \ 10 \le x_i \le 1000, i = 4, 5 \ldots, 8. \end{cases}$$

Example 5.9 ([26]).

$$\begin{cases} \min \dfrac{0.5(x_1-10)}{x_2} - x_1 \\ \text{s.t. } \frac{x_2}{x_3} + x_1 + 0.5x_1x_3 \le 100 \\ \qquad 1 \le x_i \le 100,\ i = 1, 2, 3. \end{cases}$$

Example 5.10 (Random test).

$$\begin{cases} \max \sum\limits_{i=1}^{p} \delta_i \frac{(n^i)^T x + \beta_i}{(d^i)^T x + \gamma_i} \\ \text{s.t. } \ Ax \le b, \\ \qquad x \ge 0. \end{cases}$$

where the elements of the matrix $A \in R^{m\times n}, b \in R^m, n^i, d^i \in R^n$ and the elements of constant terms of denominators and numerators β_i and $\gamma_i \in R$ are randomly generated in the interval [0,1], this agrees with the way random numbers are generated in [14], while in our experiment $\delta_i \in R$ is randomly generated in the interval [-1,1] rather than in interval [0,1], this is much more challenging to test the performance of the algorithm. The results of the contrast experiments and the random tests are shown in Tables 1-3, and each symbol used in the table has the following meaning: p, m and n represent the number of affine ratios in the objective function, the number of the constraints and the number of constrained variable respectively; Ave.time, Ave.Nod and Ave.Ite stand for the average CPU time in seconds, average number of the subproblem and iteration in the algorithm; ϵ express the error precision used in the algorithm, and α refer to the split ratio used in the branching operations.

Table 1. Results of the numerical contrast test 1-7.

Example	Methods	ϵ	α	Optimal value	Optimal solution	Iter
1	[21]	1e-8	-	(0.0, 0.283935547)	1.623183358	71
	our	1e-9	0.5	(0.0, 0.2840)	1.62318	10
2	[18]	1e-9	-	(1.1111,0.0000,0.0000)	4.0907	1289
	[20]	1e-6	-	(1.1111,1.365e-5,1.351e-5)	4.081481	39
	[21]	1e-5	-	(0.0013,0.0000,0.0000)	4.087412	1640
	our	1e-9	0.5	(1.1111,0.0000,0.0000)	4.09070	1
3	[18]	1e-6	-	(0,1)	3.575	1
	our	1e-9	0.5	(0.00000,1.00000)	3.57500	1
4	[19]	1e-6	-	(-1.838e-16,3.3333,0.0)	1.9	8
	our	1e-9	0.5	(0.00000,3.33333,0.00000)	-1.9	1
5	[18]	1e-6	-	(3,4)	5	1
	[22]	1e-2	-	(3,4)	5	11
	our	1e-9	0.3	(3.00000,4.00000)	5.0000	2
6	[17]	1e-6	-	(0.00,1.6725,0.0000)	3.0009	1033
	[21]	1e-2	-	(0.00,3.3333,0.0)	3.00292	119
	our	1e-9	0.5	(0.00,3.33333,0.00000)	3.00292	1
7	[18]	1e-3	-	$x^{(18)}$	16.2619	927
	[22]	1e-2	-	$x^{(22)}$	16.077978	620
	our	1e-3	0.65	$x^{(our1)}$	16.26283	626
8	[23]	1e-6	-	$x^{(23)}$	7049.24682	-
	our	1e-6	0.50	$x^{(our2)}$	6944.248031	35
9	[24]	1e-6	-	$x^{(24)}$	-83.249728	-
	our	1e-6	0.50	$x^{(our3)}$	-85.68859	58

while

$$\begin{aligned}
x^{(18)} &= (6.24409, 20.0249, 3.79672, 5.93972, 0, 7.43852,\\
&\quad 0, 23.2833, 0.515015, 40.9896, 0, 3.14363)^T\\
x^{(22)} &= (6.223689, 20.060317, 3.774684, 5.947841, 0, 7.456686,\\
&\quad 0, 23.312579, 0.000204, 41.031824, 0, 3.171106)^T\\
x^{(23)} &= (578.973143, 1359.572730, 5110.701048, 181.9898,\\
&\quad 295.5719, 218.0101, 286.4179, 395.5719)\\
x^{(24)} &= (87.614446, 8.754375, 1.413643, 19.311410)\\
x^{(our1)} &= (6.22442, 20.05821, 3.77441, 5.94859, 0.00001, 7.45691,\\
&\quad 0.00002, 23.31133, 0.00012, 41.03002, 0.00001, 3.17225)^T\\
x^{(our2)} &= (579.326059, 1359.9445, 5109.977472, 182.019317,\\
&\quad 295.600901, 217.980682, 286.418416, 395.600901)\\
x^{(our3)} &= (87.614446, 8.754375, 1.413643, 19.311410).
\end{aligned}$$

The computational results in Table 2 and Table 3 indicate that our algorithm has good performance, and is effective for special relatively large-scale optimization problems where the number of ratios in the objective function is not so large. Meanwhile, we find that, the average number of iterations and subproblems that need to be solved by the algorithm and the average CPU time do not substantially increase as the size of the problem becomes large. Based on the result of the above numerical examples, our algorithm is quite robust and efficient and so it can be used successfully to solve the sum of affine ratios problem SRP.

Table 2. Computational results of random test 8 corresponding to the variation of the number of variable n.

(p, m, n)	Ave.Ite	Ave.Nod	Ave.Time(s)
(2,10,10)	23	31	0.507
(2,10,20)	30	42	0.729
(2,10,30)	36	53	1.327
(2,10,40)	47	66	1.465
(2,10,50)	74	105	2.323
(2,10,60)	63	88	1.942
(2,10,70)	57	79	1.799
(2,10,80)	54	77	1.707
(2,10,90)	63	91	2.125
(2,10,100)	175	269	5.567

6 Concluding remarks

In this paper, a new kind of branch and bound optimization algorithm is presented for globally solving a class of sum of ratios problem. The algorithm is divided into three steps. First, the original problem is tactfully reformulated into an equivalent problem coupled with an outcome space, then the convex relaxation programming is established by utilizing the lower and upper bound of the auxiliary variables. At last, a new condensing operation based on the lower bound of the optimal value is presented for inciting the whole or a

Table 3. Computational results of random test 8 corresponding to the variation of the number of ratio p.

(p, m, n)	Ave.Ite	Ave.Nod	Ave.Time(s)
(2,10,100)	175	269	5.567
(3,10,90)	297	511	9.056
(4,10,80)	376	660	10.211
(5,10,70)	598	973	18.937
(6,10,60)	4235	7508	128.986
(7,10,50)	4654	8274	130.364
(8,10,40)	5420	9318	150.624
(9,10,30)	7338	11782	217.592
(10,10,20)	11844	20473	318.62

part of the investigated region in which there does not contain the global optimal solution of the equivalent problem. By combining the adapted partition rule with the accelerating technique into the reduced space branch and bound scheme, the presented algorithm is developed. Numerical results show that the proposed algorithm can suppress the rapid growth of the branching tree during the algorithm search process, and several random examples illustrate the high efficiency and stability of the algorithm.

Competing interests

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

Author's contributions

Both authors contributed equally to the manuscript, and they read and approved the final manuscript.

Acknowledgement: This paper is supported by the Science and Technology Key Project of Education Department of Henan Province (14A110024) and (15A110023);the National Natural Science Foundation of Henan Province (152300410097), the Science and Technology Projects of Henan Province (182102310941) the Cultivation Plan of Young Key Teachers in Colleges and Universities of Henan Province (2016GGJS-107), the Higher School Key Scientific Research Projects of Henan Province (18A110019, 17A110021), the Major Scientific Research Projects of Henan Institute of Science and Technology (2015ZD07), and thanks for all the references authors.

References

[1] Stancu-Minasian I.M., Fractional Programming, Kluwer Academic Publishers, Boston, 1997.

[2] Charnes A., Cooper W.W., Programming with linear fractional functionals, Nav. Res. Log. Q., 1962, 9, 181-186.

[3] Host R., Pardalos P.M., Handbook of Global Optimization, Kluwer Acdemic Publishers, Dordrecht, 1995, 495-608.

[4] Jiao H.W., Liu S.Y., Range division and compression algorithm for quadratically constrained sum of quadratic ratios, Comput. Appl. Math., 2017, 36(1), 225-247.

[5] Jiao H.W., Liu S.Y., Yin J., Zhao Y., Outcome space range reduction method for global optimization of sum of affine ratios problem, Open Math., 2016, 14, 736-746.

[6] Rao M.R., Cluster analysis and mathematical programming, J. Am. Stat. Assoc., 1971, 66, 622-626.

[7] Flak J.E., Palocsay S.W., Optimizing the sum of linear fractional functions, Recent advances in global optimization, Princeton Univerisity Press, Princeton, New Jersey, 1992.

[8] Almogy Y., Levin O., Parametric analysis of a multi-stage stochastic shipping problem, Proc. of the fifth IFORS Conf., 1964, 359-370.

[9] Konno H., Watanabe H., Bond portfolio optimization problems and their applications to rex tracking, J. Oper. Res. Soc. Jpn., 1996, 39, 295-306.

[10] Majihi J., Janardan R., Smid M., Gupta P., On some geometric optimization problems in layered manufacturing, Comp. Geom., 1999, 12, 219-239.

[11] Schwerdt J., Smid M., Janardan R., Johnson E., Majihi J., Protecting critical facets in layered manufacturing, Comp. Geom., 2000, 16, 187-210.

[12] Schaible S., Shi J., Fractional programming: the sum-of-ratios case, Optim. Method Softw., 2003, 18, 219-229.

[13] Stancu-Minasian I.M., A sixth bibliography of fractional programming, Optimization, 2006, 55, 405-428.

[14] Jiao H.W., Liu S.Y., A practicable branch and bound algorithm for sum of linear ratios problem, Eur. J. Oper. Res., 2015, 243, 723-730.

[15] Karmarkar N., A new polynomial-time algorithm for linear programming, Combinatorica, 1984, 4, 373-395.

[16] Cambini A., Martein L., Schaible S., On Maximizing a sum of ratios, J. Info. Optim. Scie., 1989, 10, 65-79.

[17] Konno H., Abe N., Minimization of the sum of three linear fractional functions, J. Global Optim., 1999, 15, 419-432.

[18] Konno H., Yajima Y., Matsui T., Parametric simplex algoriyhm for solving a special class of nonconvex minimization problems, J. Global Optim., 1991, 1, 65-81.

[19] Falk J.E., Palocsay S.W., Image space analysis of generalized fractional programs, J. Global Optim., 1994, 4, 63-88.

[20] Pei Y.G., Zhu D.T., Global optimization method for maximizing the sum of difference of convex functions ratios over nonconvex region, J. Appl. Math. Comput., 2013, 41, 153-169.

[21] Shen P.P., Wang C.F., Global optimization for sum of linear ratios problem with coefficients, Appl. Math. Comp., 2006, 176, 219-229.

[22] Wang Y.J., Shen P.P., Liang Z.A., A branch-and-bound algorithm to globally solve the sum of several linear ratios, Appl. Math. Comp., 2005, 168, 89-101.

[23] Jiao H.W., Liu S.Y., An Efficient Algorithm for Quadratic Sum-of-Ratios Fractional Programs Problem, Numer. Funct. Anal. Optim., 2017, 38(11), 1426-1445.

[24] Phuong N.T.H., Tuy H., A unified monotonic approach to generalized linear fractional programming, J. Global Optim., 2003, 26, 229-259.

[25] Lin M.H., Tsai J.F., Range reduction techniques for improving computational efficiency in global optimization of signomial geometric programming problems, Eur. J. Oper. Res., 2012, 216(1), 17-25.

[26] Dembo R.S., Avriel M., Optimal design of a membrane separation process using signomial programming, Math. Prog., 1978, 15(1), 12-25.