AIMS Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 9(9): 25396–25412.
DOI: 10.3934/math.20241240
Received: 19 July 2024
Revised: 07 August 2024
Accepted: 27 August 2024
Published: 30 August 2024

Research article

A nonlinear relaxation-strategy-based algorithm for solving sum-of-linear-ratios problems

Bo Zhang[1,2,*], Yuelin Gao[2,3], Ying Qiao[1] and Ying Sun[1]

[1] School of Mathematics and Information Sciences, North Minzu University, Yinchuan, 750021, China

[2] Ningxia province key laboratory of intelligent information and data processing, North Minzu University, Yinchuan, 750021, China

[3] Ningxia mathematics basic discipline research center, North Minzu University, Yinchuan, 750021, China

* **Correspondence:** Email: zbsdx121@163.com; Tel: +8618709602837.

Abstract: This paper mainly studies the sum-of-linear-ratios problems, which have important applications in finance, economy and computational vision. In this process, we first propose a new method to re-represent the original problem as an equivalent problem (EP). Secondly, by relaxing these constraints, a nonlinear relaxation subproblem is constructed for EP. In view of the special structure of the relaxation, it is reconstructed as a second-order cone programming (SOCP) problem, which is essentially a SOCP relaxation of EP. Thirdly, through the structural characteristics of the objective function of EP, a region reduction technique is designed to accelerate the termination of the algorithm as much as possible. By integrating the SOCP relaxation and acceleration strategy into the branch and bound framework, a new global optimization algorithm is developed. Further, the theoretical convergence and computational complexity of the algorithm are analyzed. Numerical experiment results reveal that the algorithm is effective and feasible.

Keywords: global optimization; fractional program; branch and bound; SOCP relaxation
Mathematics Subject Classification: 90C32, 90C26

1. Introduction

Consider the following sum of linear ratios problem

$$
(\text{SLR})\begin{cases}\min\ \varphi(x)=\displaystyle\sum_{i=1}^{p}\frac{c_i^\top x+d_i}{e_i^\top x+f_i}\\ \text{s.t.}\ \ x\in \mathrm{X}:=\{x\in\mathbb{R}^n|Ax\le b\}\end{cases}
$$

where $p\ge 2$, $c_i\in\mathbb{R}^n$, $d_i\in\mathbb{R}$, $e_i\in\mathbb{R}^n$, $f_i\in\mathbb{R}$, $A\in\mathbb{R}^{m\times n}$, $b\in\mathbb{R}^m$, X is a nonempty and bounded set. Besides, for any $x\in\mathrm{X}$, it is assumed that $e_i^\top x+f_i>0$ ($i\in\{1,2,\cdots,p\}$), which is without loss of generality (see [1,2]).

Over the years, many scholars have paid special attention to and studied algorithms for solving problem SLR. There are two main reasons for this. One is in the application, this problem has been widely appeared in the fields of economics [3], financial investment [4], portfolio [5,6] and system engineering [7], biodiversity conservation [8], network data envelopment analysis [9] and computer vision [10]. The other is in the theoretical research, except for some special cases [11], SLR is usually NP-hard [12], and its multiple local non-global solutions seriously interfere with the process of finding global optimal solutions. This property aggravates the difficulty of global optimization, so it is of great practical and theoretical significance to develop a new global optimization algorithm for SLR.

In general, there is a positive correlation between the difficulty of SLR and the magnitude of p. For a single linear fractional programming problem with $p=1$ and the quasi-concavity (quasi-convexity) property of the objective function, Ozkok [13] proposes an iterative algorithm based on the (ϵ,δ)-definition of continuity; Charnes and Cooper adopts an ingenious method to transform the problem into a linear program [14]. When $p=2$, Konno et al. [15] developed a parameter-based simplex method to solve the problem, while the branch-and-bound (B&B) algorithm based on wave-curve bounds proposed by Xia et al. [16] can also solve such problem cases. For decades, there have been many effective algorithms for the SLR problem with $p>2$, such as the interior point method [17], heuristic method [18], concave minimization method [19], polynomial time approximation algorithm [20], image-space analysis method [21], monotone method [22], outer approximation algorithm [23] and B&B algorithms [24–28]. Among these algorithms, the B&B algorithm is a classical global optimization method, which is often adopted to solve many difficult optimization problems. Most B&B algorithms for solving SLR are implemented by employing various equivalent transformations and establishing corresponding relaxation strategies. Also, other strategies are sometimes combined to design algorithms. For instance, Benson [24] proposed a B&B algorithm for SLR by combining simplex-based branching operations with Lagrangian dual-bound strategies. By combining the two techniques of B&B and plane cutting, Benson [25] also proposed a global algorithm, which adds linear cuts but does not compute newly generated vertices. Kuno and Masaki [10] and Carlsson and Shi [26] constructed two B&B algorithms based on the linear relaxation technique, but their branching operations are all performed in the n-dimensional decision space, so that the computational efficiency decreases with the increase of the number of variables. For solving generalized sum-of-ratio programming problems, Ashtiani and Paulo [29] presented a cut-plane algorithm combining a B&B technique with linear relaxation, whose branching operations are performed in the $2p$-dimensional output space. By transforming the SLR problem into an equivalent bilinear programming problem with bilinear constraints, Jiao and Liu [30], Liu and Ge [31] and Liu et al. [32] respectively proposed different linear relaxation strategies to simultaneously relax the objective and constraint functions, and designed different B&B algorithms whose branching operations take place in p-dimensional outer space. Recently, Jiao and Ma [33] proposed a new linear relaxation strategy based on the essential structure of each fractional function in ten different cases, from which a B&B algorithm was designed by combining the corresponding acceleration techniques and branching operations in p-dimensional outer space. To reduce the dimension of the outer space where branching operations are performed, Zhang et al. [1, 2] designed two B&B algorithms based on Charnes–Cooper (CC) transformation [14] and different linear relaxation strategies, both of which branching operations occur in the $(p-1)$-dimensional outer space. Similarly, Shen et al. [34] also designed a B&B algorithm with branching operations in a $(p-1)$-dimensional space, but it incorporated a second-order cone programming (SOCP) relaxation technique. For a more detailed introduction of SLR and its generalized form, it is recommended to refer to the fractional programming bibliography [35].

In this paper, a novel B&B algorithm is developed for solving the SLR problem globally. Our main work is to reformulate the SLR problem into a new equivalent problem (EP) with an non-convex objective function and $(p-1)$ non-convex constraints by introducing intermediate variables. Thus, a new nonlinear relaxation strategy is proposed based on the relaxation of these non-convex constraints. According to the characteristics of the objective function of the relaxation subproblem, a corresponding acceleration technique is designed. It is noted that Zhang et al. [1, 2] and Shen et al. [34] firstly employed CC transformation to reduce the number of linear fractions in SLR from p to $(p-1)$, and thus proposed new B&B algorithms that execute branching processes in $(p-1)$-dimensional space. Differently, based on the new equivalent problem, we study the problem from another point of view, and finally propose a nonlinear relaxation. Through an artful transformation, the nonlinear relaxation subproblem is finally reconstructed into a SOCP problem. Compared with some existing B&B algorithms, our algorithm does not perform branching operations in p-, $2p$- or n-dimensional space, so it may greatly save computation when solving some SLR instances. Furthermore, the SLR problem we studied only assumes that the denominator of each fractional function is positive, instead of the positive denominator and non-negative numerator in [20, 30, 36]. Of course, the computational complexity of the proposed algorithm is derived in detail, and its analysis method is different from those in the existing literature (e.g., [1, 33, 34]). This estimates the maximum number of iterations for our algorithm as lowly as possible. Finally, the numerical results show that the algorithm is feasible and effective. Overall, our algorithm can solve all small and medium-sized SLR problems well, and is quite stable and effective for large-scale problems.

The rest of this paper is organized as follows: In Section 2, we give the relevant theories of the algorithm, which mainly includes the equivalent problem, bounding operation, branching operation, rectangle-region reduction technique, detailed steps and computational complexity. Section 3 introduces some test examples in the existing literature, and gives the computation results and numerical analysis. Finally, a positive review and outlook are given for the research of this paper.

2. Theoretical framework

To solve the problem of SLR globally, this section mainly presents a new B&B algorithm based on a nonlinear relaxation strategy.

2.1. Equivalent problem and its analysis

In this section, in order to solve the problem SLR based on the B&B algorithm, we need to propose a new equivalent problem for it. To this end, we reformulate the SLR as the following form:

$$
(\text{EP})\begin{cases} \min\ \phi(x,\mu) = \displaystyle\sum_{i=1}^{p-1}\mu_i - \sum_{i=1}^{p-1}\alpha_i\left(\frac{e_i^\top x + f_i}{e_p^\top x + f_p}\right) + \frac{c_p^\top x + d_p}{e_p^\top x + f_p} \\ \text{s.t. } \dfrac{c_i^\top x + d_i}{e_i^\top x + f_i} + \alpha_i\left(\dfrac{e_i^\top x + f_i}{e_p^\top x + f_p}\right) \le \mu_i, i = 1,\cdots,p-1, \\ \quad x \in \mathrm{X} \end{cases}
$$

by introducing a vector $\mu = (\mu_i, \mu_2, \cdots, \mu_{p-1})^\top \in \mathbb{R}^{p-1}$, $\alpha_i > 0$.

The equivalence between problems EP and SLR is given by the following theorem.

Theorem 1. *A point $x^* \in \mathbb{R}^n$ is global optimal for the problem* SLR *if and only if $(x^*, \mu^*) \in \mathbb{R}^{n+p-1}$ is global optimal for the problem* EP *with $\mu_i^* = \frac{c_i^\top x^* + d_i}{e_i^\top x^* + f_i} + \alpha_i\left(\frac{e_i^\top x^* + f_i}{e_p^\top x^* + f_p}\right)$, $i = 1, \cdots, p-1$.*

Proof. For each $i = 1, \cdots, p-1$, the univariate function μ_i is monotonically increasing as μ_i increases, so the conclusion of the theorem clearly holds. □

Theorem 1 illustrates that the problem ESLR can be addressed by solving the EP. At the same time, a partial component x^* of the optimal solution (x^*, μ^*) of EP becomes the optimal solution of SLR. Thus, the study of problem EP will be focused.

It can be found that the objective function and constraints

$$
\frac{c_i^\top x + d_i}{e_i^\top x + f_i} + \alpha_i\left(\frac{e_i^\top x + f_i}{e_p^\top x + f_p}\right) \le \mu_i \tag{2.1}
$$

of EP are non-convex, so that the non-convexity of the problem is currently reflected in these places. However, we only investigate the relaxation of these non-convex constraints, while the objective function will be simply linearized later. To do so, it is necessary to know the initial upper bound $\overline{\mu}_i^0$ and lower bound $\underline{\mu}_i^0$ of μ_i^* such that $\underline{\mu}_i^0 \le \mu_i^* := \frac{c_i^\top x^* + d_i}{e_i^\top x^* + f_i} + \alpha_i\left(\frac{e_i^\top x^* + f_i}{e_p^\top x^* + f_p}\right) \le \overline{\mu}_i^0$, and to construct an initial box $H_0 := \prod_{i=1}^{p-1}\left[\underline{\mu}_i^0, \overline{\mu}_i^0\right]$. Hence, for each $i = 1, 2, \cdots, p-1$, the following problems need to be resolved:

$$
\underline{\eta}_i = \min_{x\in \mathrm{X}} \frac{c_i^\top x + d_i}{e_i^\top x + f_i},\ \ \overline{\eta}_i = \max_{x\in \mathrm{X}} \frac{c_i^\top x + d_i}{e_i^\top x + f_i}, \tag{2.2}
$$

$$
\underline{\varsigma}_i = \min_{x\in \mathrm{X}} \frac{e_i^\top x + f_i}{e_p^\top x + f_p},\ \ \overline{\varsigma}_i = \max_{x\in \mathrm{X}} \frac{e_i^\top x + f_i}{e_p^\top x + f_p}. \tag{2.3}
$$

It is imperative to note that subsequent to the adoption of CC transformation, the problems in Eqs (2.2) and (2.3) can be reformulated into corresponding linear programming problems, thereby facilitating their resolution. Then let

$$
\underline{\mu}_i^0 := \underline{\eta}_i + \alpha_i\underline{\varsigma}_i,\ \ \overline{\mu}_i^0 := \overline{\eta}_i + \alpha_i\overline{\varsigma}_i, \tag{2.4}
$$

which satisfies $\underline{\mu}_i^0 \le \mu_i^* \le \overline{\mu}_i^0$ for $i = 1, 2, \cdots, p-1$.

Based on the above discussion, it can be concluded that (x^*, μ^*) must satisfy $x^* \in X$ and $\underline{\mu}_i^0 \le \mu_i^* \le \overline{\mu}_i^0$ $(i = 1, 2, \cdots, p-1)$. Next, for each $i = 1, 2, \cdots, p-1$, we multiply $\frac{e_i^\top x + f_i}{e_p^\top x + f_p}$ onto both sides of the constraint (2.1) to obtain the following equivalent nonlinear constraint:

$$\frac{c_i^\top x + d_i}{e_p^\top x + f_p} + \alpha_i \left(\frac{e_i^\top x + f_i}{e_p^\top x + f_p} \right)^2 \le \frac{\mu_i (e_i^\top x + f_i)}{e_p^\top x + f_p}. \tag{2.5}$$

By adding $\frac{(\mu_i)^2}{4\alpha_i}$ to both sides of Eq (2.5) and doing a simple arrangement, the inequality can be rewritten as:

$$\left(\frac{\sqrt{\alpha_i}(e_i^\top x + f_i)}{e_p^\top x + f_p} - \frac{\mu_i}{2\sqrt{\alpha_i}} \right)^2 + \frac{c_i^\top x + d_i}{e_p^\top x + f_p} \le \frac{(\mu_i)^2}{4\alpha_i}, \quad i = 1, 2, \cdots, p-1. \tag{2.6}$$

Accordingly, EP is clearly equivalent to the following problem:

$$\mathrm{EP}(H_0) \begin{cases} \min \ \phi(x, \mu) = \sum_{i=1}^{p-1} \mu_i + \dfrac{c_p^\top x + d_p - \sum_{i=1}^{p-1} \alpha_i (e_i^\top x + f_i)}{e_p^\top x + f_p} \\ \text{s.t.} \ \ Eq.(2.6) \ \ x \in X, \ \mu \in H_0. \end{cases}$$

Suppose that $H = \prod_{i=1}^{p-1} [\underline{\mu}_i, \overline{\mu}_i]$ denotes any sub-rectangle of H_0, i.e., $H \subseteq H_0$, then the subproblem of EP over H can be formulated as follows:

$$\mathrm{EP}(H) \ \begin{cases} \min \ \phi(x, \mu) = \sum_{i=1}^{p-1} \mu_i + \dfrac{c_p^\top x + d_p - \sum_{i=1}^{p-1} \alpha_i (e_i^\top x + f_i)}{e_p^\top x + f_p} \\ \text{s.t.} \ \ Eq.(2.6) \ \ x \in X, \ \mu \in H. \end{cases}$$

Based on the characteristics of the B&B algorithm, we will propose a nonlinear relaxation strategy of the subproblem EP(H) for the execution of the bounding operation.

2.2. Nonlinear relaxation strategy

In this section, we mainly relax EP(H) as a nonlinear relaxation programming (NLRP) problem that can provide a lower bound for the optimal value of EP(H) in the proposed algorithm.

Now, after linearizing a term on the right-hand side of each non-convex constraint (2.6), the corresponding nonlinear relaxation subproblem can be obtained. It is well known that the concave envelope of a univariate convex function $(\mu_i)^2$ over the interval $[\underline{\mu}_i, \overline{\mu}_i]$ can be formulated as

$$\vartheta_i(\mu_i) = (\underline{\mu}_i + \overline{\mu}_i)\mu_i - \underline{\mu}_i \overline{\mu}_i, \ i = 1, 2, \cdots, p, \tag{2.7}$$

which necessarily satisfy

$$\vartheta_i(\mu_i) \ge (\mu_i)^2 \text{ for any } \mu_i \in [\underline{\mu}_i, \overline{\mu}_i], \ i = 1, 2, \cdots, p. \tag{2.8}$$

According to Eq (2.8), each of the above constraints (2.6) can be relaxed into:

$$\left(\frac{\sqrt{\alpha_i}(e_i^\top x+f_i)}{e_p^\top x+f_p}-\frac{\mu_i}{2\sqrt{\alpha_i}}\right)^2+\frac{c_i^\top x+d_i}{e_p^\top x+f_p}\le\frac{1}{4\alpha_i}\vartheta_i(\mu_i),\ i=1,2,\cdots,p-1. \tag{2.9}$$

Thus, EP(H) is finally relaxed as the following nonconvex program:

$$\text{NLRP}(H)\ \begin{cases}\min\ \psi(x,\mu)=\sum_{i=1}^{p-1}\mu_i+\dfrac{c_p^\top x+d_p-\sum_{i=1}^{p-1}\alpha_i(e_i^\top x+f_i)}{e_p^\top x+f_p}\\ \text{s.t. } Eq.(2.9),\ x\in \mathrm{X},\ \mu\in H.\end{cases}$$

From Eq (2.8), it can be known that $\vartheta_i(\mu_i)$ is actually a linear upper approximation function of $(\mu_i)^2$ over $[\underline{\mu}_i,\overline{\mu}_i]$, which implies that the feasible region of EP(H) always does not exceed that of problem NLRP(H), so that the optimal value of the latter is never greater than that of the former. This clarifies that solving the NLRP(H) yields an effective lower bound on the optimal value of EP(H).

For each $i=1,2,\cdots,p-1$, if the positive fractional function $\frac{e_p^\top x+f_p}{e_i^\top x+f_i}$ is multiplied by the left and right sides of Eq (2.9), the following inequality can be obtained:

$$\frac{c_i^\top x+d_i}{e_i^\top x+f_i}+\alpha_i\left(\frac{e_i^\top x+f_i}{e_p^\top x+f_p}\right)-\mu_i\le\frac{(e_p^\top x+f_p)(\vartheta_i(\mu_i)-(\mu_i)^2)}{4\alpha_i(e_i^\top x+f_i)}\le\frac{(\overline{\mu}_i-\underline{\mu}_i)^2}{16\alpha_i\underline{\varsigma}_i}. \tag{2.10}$$

After comparing Eqs (2.1) and (2.10), the relaxation process essentially magnifies 0 to $\frac{(e_p^\top x+f_p)(\vartheta_i(\mu_i)-(\mu_i)^2)}{4\alpha_i(e_i^\top x+f_i)}$, thus relaxing the feasible region of the problem EP(H). From the above relaxation process, it can be found that different values of α_i will lead to relaxation problems NLRP(H) with different compactness. When $H=H^0$, by the definition of $\underline{\mu}_i^0$ and $\overline{\mu}_i^0$ in Eq (2.4), the upper limit of constraint error in Eq (2.10) is closely related to the value of α_i, i.e.,

$$\frac{(\overline{\mu}_i^0-\underline{\mu}_i^0)^2}{16\alpha_i\underline{\varsigma}_i}=\frac{(\overline{\eta}_i-\underline{\eta}_i+\alpha_i(\overline{\varsigma}_i-\underline{\varsigma}_i))^2}{16\alpha_i\underline{\varsigma}_i}=:\xi(\alpha_i).$$

Over the interval $(0,+\infty)$, the minimum value of the univariate function $\xi(\alpha_i)$ is reached at the point $\frac{\overline{\eta}_i-\underline{\eta}_i}{\overline{\varsigma}_i-\underline{\varsigma}_i}$. Consequently, under the condition of $\alpha_i=\frac{\overline{\eta}_i-\underline{\eta}_i}{\overline{\varsigma}_i-\underline{\varsigma}_i}$, when each non-convex constraint (2.1) is relaxed, the upper bound of the error between the optimal values of EP(H^0) and NLRP(H^0) can be minimized as much as possible. Therefore, throughout this paper, the value of each parameter α_i will always be $\frac{\overline{\eta}_i-\underline{\eta}_i}{\overline{\varsigma}_i-\underline{\varsigma}_i}$.

Also, it can be observed from Eqs (2.7), (2.8) and (2.10) that the rectangles corresponding to the variable μ can be directly branching and thinning, which forces the optimal values of NLRP(H) and EP(H) to gradually approach each other in the limiting sense.

2.3. *Solving problem* NLRP*(H)*

Given a rectangle $H\subseteq H_0$, if the problem NLRP(H) is solvable, we notice that the problem cannot be solved directly by using the existing convex optimization solver, which is extremely inconvenient. However, NLRP(H) is implicitly convex and can be revealed as follows:

It is observed that the linear fractional factors in NLRP(H) have the same denominator $e_p^\top x + f_p$. By introducing the CC transformation: $t = \frac{1}{e_p^\top x + f_p}$ and $y = tx$, the problem can be transformed into the following convex problem:

$$\mathrm{CP}(H)\quad \begin{cases} \min \omega(y,t,\mu) = \sum_{i=1}^{p-1} \mu_i + c_p^\top y + d_p t - \sum_{i=1}^{p-1} \alpha_i (e_i^\top y + f_t) \\ \text{s.t. } \left(\sqrt{\alpha_i}(e_i^\top y + f_i t) - \frac{\mu_i}{2\sqrt{\alpha_i}}\right)^2 + c_i^\top y + d_i t \le \frac{1}{4\alpha_i}\vartheta_i(\mu_i),\ i = 1,2,\cdots,p-1, \\ \qquad e_p^\top y + f_p t = 1,\ Ay - bt \le 0,\ t > 0,\ \mu \in H. \end{cases}$$

Remark 1. *Let* $\mathrm{W} = \{(y,t) | e_p^\top y + f_p t = 1,\ Ay - bt \le 0,\ t > 0\}$*, for any* $(y,t) \in \mathrm{W}$*, it holds that* $t > 0$ *and* $y/t \in \mathrm{X}$*. Besides, for any* $x \in \mathrm{X}$*, let* $t = \frac{1}{e_p^\top x + f_p}$*,* $y = tx$*, it can be verified that* $(y,t) \in \mathrm{W}$ *and* $\omega(y,t,\mu) = \psi(x,\mu)$*. Therefore, we can obtain the optimal solution* $(\frac{\hat{y}}{\hat{t}}, \hat{\mu})$ *of* NLRP*(*H*) from the optimal solution* $(\hat{y}, \hat{t}, \hat{\mu})$ *of* CP*(*H*).*

The problem CP(H) is well defined, as is stated in [1, 14]. As a result, the conclusions in Remark 1 are obvious. Although CP(H) can be directly handled by some existing convex optimization solvers, since the quadratic term of each convex constraint is the square of a linear factor, the problem can be essentially rewritten as the following SOCP problem:

$$\mathrm{SOCP}(H)\quad \begin{cases} \min\ \sum_{i=1}^{p-1} \mu_i + c_p^\top y + d_p t - \sum_{i=1}^{p-1} \alpha_i (e_i^\top y + f_t) \\ \text{s.t. } \left\|(4\alpha_i(e_i^\top y + f_i t) - 2\mu_i, \beta_i - 1)\right\|_2 \le \beta_i + 1,\ i = 1,2,\cdots,p-1, \\ \qquad \beta_i = \vartheta_i(\mu_i) - 4\alpha_i(c_i^\top y + d_i t),\ i = 1,2,\cdots,p-1, \\ \qquad e_p^\top y + f_p t = 1,\ Ay - bt \le 0,\ t > 0,\ \mu \in H. \end{cases}$$

This problem can be solved directly by the solver coneprog in MATLAB(2023a), which can also be called a SOCP relaxation of EP(H).

2.4. Branching rule of rectangle

Branching operation is essential in B&B algorithm.

In our algorithm, the branching rule of $H = \prod_{i=1}^{p-1} [\underline{\mu}_i, \overline{\mu}_i] \subseteq H_0$ can be summarized as follows:

(i) Let $\overline{\mu}_\kappa - \underline{\mu}_\kappa = \max\{\overline{\mu}_i - \underline{\mu}_i : i = 1,2,\cdots,p-1\}$, $z_\kappa = \frac{1}{2}(\underline{\mu}_\kappa + \overline{\mu}_\kappa)$;

(ii) By adopting z_κ, the interval $[\underline{\mu}_\kappa, \overline{\mu}_\kappa]$ corresponding to the κ-edge of H is divided into two intervals $[\underline{\mu}_\kappa, z_\kappa]$ and $[z_\kappa, \overline{\mu}_\kappa]$, and then H is subdivided into two sub-rectangles

$$H_1 = \prod_{i=1}^{\kappa-1} [\underline{\mu}_i, \overline{\mu}_i] \times [\underline{\mu}_\kappa, z_\kappa] \times \prod_{i=\kappa+1}^{p-1} [\underline{\mu}_i, \overline{\mu}_i],\ H_2 = \prod_{i=1}^{\kappa-1} [\underline{\mu}_i, \overline{\mu}_i] \times [z_\kappa, \overline{\mu}_\kappa] \times \prod_{i=\kappa+1}^{p-1} [\underline{\mu}_i, \overline{\mu}_i].$$

Based on the above discussion, we can know that $H_1 \cap H_2 = \{\mu \in \mathbb{R}^{p-1} | \mu_\kappa = z_\kappa\}$ and $H_1 \cup H_2 = H$.

2.5. Region reduction technique

In this subsection, a simple region reduction technique is mainly derived, so as to delete some or all regions in a sub-rectangle H that cannot obtain the global optimal solution, which can reduce the explored child nodes or compress the search domain, thus accelerating the convergence speed of the B&B algorithm.

Without loss of generality, let $H = \prod_{i=1}^{p-1}[\underline{\mu}_i, \overline{\mu}_i] \subseteq H_0$. Let $\overline{UB}$ denote the currently known best objective function value of problem EP.

For any feasible solution (x, μ) of EP(H) worth being considered, it must satisfy

$$\phi(x,\mu) = \sum_{i=1}^{p-1} \mu_i + \frac{c_p^\top x + d_p - \sum_{i=1}^{p-1} \alpha_i(e_i^\top x + f_i)}{e_p^\top x + f_p} \leq \overline{UB},$$

then for every $\iota = 1, 2, \cdots, p-1$, we define

$$\gamma = \hat{\mu} + \sum_{i=1}^{p-1} \underline{\mu}_i, \quad \overline{\alpha}_\iota = \overline{UB} - \gamma + \underline{\mu}_\iota,$$

where $\hat{\mu} = \min_{x \in X} \frac{c_p^\top x + d_p - \sum_{i=1}^{p-1} \alpha_i(e_i^\top x + f_i)}{e_p^\top x + f_p}$.

If the global optimal solution (x^*, μ^*) of EP can be obtained by employing H, there must be a necessary condition:

$$\phi(x^*, \mu^*) \leq \phi(x,\mu) \leq \overline{UB},\ for\ some\ (x,\mu) \in X \times H, \tag{2.11}$$

which is the key to the following rectangular reduction theorem.

Theorem 2. *If $\overline{\alpha}_\iota < \underline{\mu}_\iota$ for a $\iota \in \{1, 2, \cdots, p-1\}$, the problem* EP *cannot obtain the global optimal solution over the rectangle H; otherwise, if $\overline{\alpha}_\iota < \overline{\mu}_\iota$ for some $\iota \in \{1, 2, \cdots, p-1\}$, the global optimal solution cannot be obtained from $\overline{H}_\iota$, where*

$$\overline{H}_\iota = \prod_{i=1}^{\iota-1}[\underline{\mu}_i, \overline{\mu}_i] \times (\overline{\alpha}_\iota, \overline{\mu}_\iota] \times \prod_{i=\iota+1}^{p-1}[\underline{\mu}_i, \overline{\mu}_i] \subseteq H.$$

Proof. If there is a $\iota \in \{1, 2, \cdots, p-1\}$ such that $\overline{\alpha}_\iota < \underline{\mu}_\iota$, it follows that

$$\phi(x^*, \mu^*) \leq \overline{UB} = \overline{\alpha}_\iota + \gamma - \underline{\mu}_\iota < \gamma = \hat{\mu} + \sum_{i=1}^{p-1} \underline{\mu}_i \leq \phi(x,\mu),\ \forall x \in X,\ \mu \in H,$$

which contradicts Eq (2.11), so that the former conclusion of the theorem holds. Further, if $\underline{\mu}_\iota \leq \overline{\alpha}_\iota < \overline{\mu}_\iota$ for some $\iota \in \{1, 2, \cdots, p-1\}$, it follows from $\mu = (\mu_1, \mu_2, \cdots, \mu_{p-1})^\top \in \overline{H}_\iota$ and the definition of $\overline{\alpha}_\iota$ that $\overline{\alpha}_\iota < \mu_\iota \leq \overline{\mu}_\iota$ and

$$\phi(x^*, \mu^*) \leq \overline{UB} = \overline{\alpha}_\iota + \gamma - \underline{\mu}_\iota < \mu_\iota + \gamma - \underline{\mu}_\iota \leq \mu_\iota + \hat{\mu} + \sum_{i=1, i\neq\iota}^{p-1} \mu_i \leq \phi(x,\mu)$$

for all $x \in X$, $\mu \in \overline{H}_\iota$. This means that no element μ in $\overline{H}_\iota$ can be a component of the optimal solution (x^*, μ^*), i.e., $\mu^* \neq \mu$ for any $\mu \in \overline{H}_\iota$. So the proof of the theorem is complete. □

2.6. SOCP relaxation based Branch-and-Bound Reduction Algorithm

To find the global optimal solution to the lifted problem EP, we construct a SOCP relaxation based branch-and-bound reduction algorithm (SOCPRBBRA) by adding the proposed convex relaxation, rectangular branching technique and rectangle-reduction rule to the B&B framework.

Algorithm (SOCPRBBRA)
Step 0. (Initialization).
Given a tolerance $\epsilon > 0$. Calculate $\underline{\mu}_i^0$ and $\overline{\mu}_i^0$ by Eqs (2.2)–(2.4).
Initialize the rectangle $H_0 = \prod_{i=1}^{p-1}[\underline{\mu}_i^0, \overline{\mu}_i^0]$.
Solve the relaxation problem NLRP(H_0) to obtain its optimal value $LB(H_0)$ and optimal solution $(\hat{x}, \hat{\mu})$.
Set $LB^0 = LB(H_0)$, $LB^0 = \varphi(\hat{x})$, $x^v = \hat{x}$, $\Xi = \{[H_0, LB(H_0)]\}$, $k := 0$.
Step 1. (Termination).
If $UB^k - LB^k \leq \epsilon$, terminate and output x^v.
Step 2. (Rectangular subdivision).
By using the bisection method shown in Sect. 2.4, the rectangle H_k is divided into two sub-rectangles H_{k1} and H_{k2}. Set $\Xi := \Xi \backslash \{[H_k, UB(H_k)]\}$.
Step 3. (Region reduction operation).
For each $\varsigma = 1, 2$, remove or reduce $H_{k\varsigma}$ with the help of the region reduction technique in Sect. 2.5. Put all the reduced rectangles into the set Q and denote the number of elements in Q as $|Q|$, i.e. $|Q| = 0, 1$ or 2. If $|Q| \neq 0$, proceed to the next step; otherwise, go to Step 5.
Step 4. (Pruning operation, update the upper bound).
For each $H \in Q$, solve the relaxation problem NLRP(H) to obtain its optimal value $LB(H)$ and optimal solution $(\hat{x}, \hat{\mu})$; if $UB^k - LB(H) > \epsilon$, set $\Xi := \Xi \cup \{[H, LB(H)]\}$ and $U = \varphi(\hat{x})$, if $U < UB^k$, set $UB^k = U$, $x^v = \hat{x}$.
Step 5. (Determine the lower bound).
If $\Xi \neq \emptyset$, set $LB^k := \min\{LB(H) : [\cdot, LB(H)] \in \Xi\}$ and goto Step 6, otherwise stop and output x^v, $k = k + 1$.
Step 6. (Select a rectangle).
Choose an element $\{[H_k, \cdot]\} \in \Xi$ such that $LB(H_k) = LB^k$.
Set $k := k + 1$ and return to Step 1.

Remark 2. *In this algorithm, the optimal solution* $(\hat{x}, \hat{\mu})$ *of the relaxation problem* NLRP*(H) is not necessarily feasible for* EP*, but it can be verified that* $(\hat{x}, \mu^{\hat{x}})$ *is feasible for* EP *when* $\mu^{\hat{x}} = (\mu_1^{\hat{x}}, \mu_2^{\hat{x}}, \cdots, \mu_{p-1}^{\hat{x}})^\top$ *with* $\mu_i^{\hat{x}} = \frac{c_i^\top \hat{x} + d_i}{e_i^\top \hat{x} + f_i} + \frac{\alpha_i(e_i^\top \hat{x} + f_i)}{e_p^\top \hat{x} + f_p}$ *is set, in which case* $\phi(\hat{x}, \mu^{\hat{x}}) = \varphi(\hat{x})$ *is found. Therefore, we can directly choose* $\varphi(\hat{x})$ *(if possible) to update the upper bound instead of* $\phi(\hat{x}, \mu^{\hat{x}})$*.*

In the above algorithm, k is adopted as the iteration index. At each iteration, one subproblem is selected and up to two new subproblems are created to replace the old one. The optimal value of the new subproblem will gradually improve compared with the old subproblem, and the corresponding upper and lower bounds will be updated, so that the gap between the upper and lower bounds will gradually decrease. Indeed, SOCPRBBRA can output a global ϵ-optimal solution for the problem EP or SLR with a given tolerance $\epsilon > 0$. Here, we call $x^v \in X$ a global ϵ-optimal solution to SLR if $\varphi(x^v) \leq V(\text{SLR}) + \epsilon$, where $V(\text{SLR})$ denotes the optimal value of SLR.

Theorem 3. *Given a tolerance $\epsilon > 0$, when* SOCPRBBRA *runs to Step 1 of the kth iteration, if the subproblem* $\{[H_k, UB(H_k)]\}$ *satisfies* $\overline{\mu}_\kappa - \underline{\mu}_\kappa \leq 4\sqrt{\frac{\epsilon}{N}}$ *with* $\overline{\mu}_\kappa - \underline{\mu}_\kappa = \max\{\overline{\mu}_i - \underline{\mu}_i : i = 1, 2, \cdots, p-1\}$, $N = \sum_{i=1}^{p-1} \frac{1}{\alpha_i \underline{\varsigma}_i}$, *the algorithm must terminate and output a global ϵ-optimal solution to problem* SLR.

Proof. Without loss of generality, denote H_k as $H = \prod_{i=1}^{p-1}[\underline{\mu}_i, \overline{\mu}_i]$, LB^k as LB, and UB^k as UB. Since H is the selected rectangle to be divided, the following formula must be established:

$$V(\text{EP}) - V(\text{NLRP}(H)) \geq 0, \tag{2.12}$$

Now, let $(\tilde{x}, \tilde{\mu})$ be an optimal solution of NLRP(H). We have

$$\phi(\tilde{x}, \hat{\mu}) - \psi(\tilde{x}, \tilde{\mu}) = \sum_{i=1}^{p-1}(\hat{\mu}_i - \tilde{\mu}_i) \leq \sum_{i=1}^{p-1} \frac{(e_p^\top \tilde{x} + f_p)(\vartheta_i(\tilde{\mu}_i) - (\tilde{\mu}_i)^2)}{4\alpha_i(e_i^\top \tilde{x} + f_i)}, \tag{2.13}$$

where $\hat{\mu} = (\hat{\mu}_1, \hat{\mu}_2, \cdots, \hat{\mu}_{p-1})^\top$ with $\hat{\mu}_i = \frac{c_i^\top \tilde{x} + d_i}{e_i^\top \tilde{x} + f_i} + \frac{\alpha_i(e_i^\top \tilde{x} + f_i)}{e_p^\top \tilde{x} + f_p}$. Furthermore, it follows from Eqs (2.7) and (2.8) that

$$\max_{\tilde{\mu}_i \in [\underline{\mu}_i, \overline{\mu}_i]} \left(\vartheta_i(\tilde{\mu}_i) - (\tilde{\mu}_i)^2\right) = \frac{(\overline{\mu}_i - \underline{\mu}_i)^2}{4}, \; i = 1, 2, \cdots, p-1. \tag{2.14}$$

Hence, from Eqs (2.12)–(2.14), it holds that

$$0 \leq V(\text{EP}) - V(\text{NLRP}(H)) \leq \phi(\tilde{x}, \hat{\mu}) - \psi(\tilde{x}, \tilde{\mu}) \leq \sum_{i=1}^{p-1} \frac{(\overline{\mu}_i - \underline{\mu}_i)^2}{16\alpha_i \underline{\varsigma}_i} \leq \frac{N}{16}(\overline{\mu}_\kappa - \underline{\mu}_\kappa)^2, \tag{2.15}$$

where $\underline{\varsigma}_i$ is defined in Eq (2.3). When $\overline{\mu}_\kappa - \underline{\mu}_\kappa \leq 4\sqrt{\frac{\epsilon}{N}}$, it follows from Eq (2.15) that

$$0 \leq V(\text{EP}) - V(\text{NLRP}(H)) \leq \phi(\tilde{x}, \hat{\mu}) - \psi(\tilde{x}, \tilde{\mu}) \leq \frac{N}{16}(\overline{\mu}_\kappa - \underline{\mu}_\kappa)^2 \leq \epsilon. \tag{2.16}$$

Since $LB = LB(H)$ is the smallest lower bound at the current iteration, it holds that

$$\psi(\tilde{x}, \tilde{\mu}) = LB \leq V(\text{EP}) = V(\text{SLR}) \leq UB = \varphi(x^v) = \phi(x^v, \mu^v) \leq \phi(\tilde{x}, \hat{\mu}), \tag{2.17}$$

where $\mu^v = (\mu_1^v, \mu_2^v, \cdots, \mu_{p-1}^v)^\top$ with $\mu_i^v = \frac{c_i^\top x^v + d_i}{e_i^\top x^v + f_i} + \frac{\alpha_i(e_i^\top x^v + f_i)}{e_p^\top x^v + f_p}$, $i = 1, 2, \cdots, p-1$. When $\overline{\mu}_\kappa - \underline{\mu}_\kappa \leq 4\sqrt{\frac{\epsilon}{N}}$, it follows from Eqs (2.16) and (2.17) that

$$\phi(x^v, \mu^v) - V(\text{EP}) = \varphi(x^v) - V(\text{SLR}) \leq UB - LB \leq \phi(\tilde{x}, \hat{\mu}) - \psi(\tilde{x}, \tilde{\mu}) \leq \epsilon, \tag{2.18}$$

which clarifies that x^v and (x^v, μ^v) are global ϵ-optimal solutions to problems SLR and EP, respectively. □

Theorem 3 shows that the optimal value of EP over each selected sub-rectangle and that of its relaxation problem NLRP(H) are gradually approaching in the limiting sense. This implies that the bounding and branching operations are consistent, so the B&B algorithm is theoretically globally convergent.

Now, let us analyze the complexity of SOCPRBBRA based on Theorem 3 and the iterative mechanism of this algorithm.

Theorem 4. *Given a tolerance $\epsilon > 0$, the maximum number of iterations required by* SOCPRBBRA *to obtain a global ϵ-optimal solution for* SLR *is*

$$\left\lfloor \prod_{i=1}^{p-1}(\overline{\mu}_i^0 - \underline{\mu}_i^0)\left(\frac{N}{16\epsilon}\right)^{(p-1)/2} \right\rfloor,$$

where $N = \sum_{i=1}^{p-1} \frac{1}{\alpha_i \underline{\varsigma}_i}$.

Proof. When SOCPRBBRA terminates, either $k = 0$ or $k \geq 1$. If $k = 0$, the algorithm does not enter the iteration loop. Thus, let us talk about the case where the algorithm terminates after many iterations.

At the case of $k \geq 1$, it follows from Theorem 3 that $\overline{\mu}_\kappa - \underline{\mu}_\kappa \leq 4\sqrt{\frac{\epsilon}{N}}$ is essentially a sufficient condition for the termination criterion of the algorithm, $UB^k - LB^k \leq \epsilon$, to hold. Further, according to the rectangular branching rule in Step 2, a total of $k+1$ sub-rectangles are generated for the initial rectangle H_0. For convenience, we denote these subrectangles as $H_1, H_2, \cdots, H_{k+1}$, respectively. Obviously, $H_0 = \bigcup_{\iota=1}^{k+1} H_\iota$. In the worst case, suppose that the longest edge of each subrectangle $H_\iota := \prod_{i=1}^{p-1}[\underline{\mu}_i, \overline{\mu}_i]$ satisfies $\overline{\mu}_\kappa - \underline{\mu}_\kappa \leq 4\sqrt{\frac{\epsilon}{N}}$, where $\kappa \in \arg\max\{\overline{\mu}_i - \underline{\mu}_i : i = 1, 2, \cdots, p-1\}$. At this point, every edge $[\underline{\mu}_i, \overline{\mu}_i]$ of H_ι satisfies

$$\overline{\mu}_i - \underline{\mu}_i \leq 4\sqrt{\frac{\epsilon}{N}},\ i = 1, \cdots, p-1, \tag{2.19}$$

which implies that the volume $Vol(H_\iota)$ of H_ι does not exceed the volume $Vol(\bar{H}_\iota)$ of a rectangle $\bar{H}_\iota$ with a unique edge length $4\sqrt{\frac{\epsilon}{N}}$. Thus, we have

$$Vol(H_0) = \prod_{i=1}^{p-1}(\overline{\mu}_i^0 - \underline{\mu}_i^0) = \sum_{\iota=1}^{k+1} Vol(H_\iota) \leq (k+1)Vol(\bar{H}_\iota) = (k+1)\left(4\sqrt{\frac{\epsilon}{N}}\right)^{p-1}. \tag{2.20}$$

Next, by combining Eq (2.20), we have

$$k \geq \frac{Vol(H_0)}{Vol(\bar{H}_\iota)} - 1 = \prod_{i=1}^{p-1}(\overline{\mu}_i^0 - \underline{\mu}_i^0)\left(\frac{N}{16\epsilon}\right)^{(p-1)/2} - 1.$$

However, when Eq (2.19) holds and $k = \left\lfloor \prod_{i=1}^{p-1}(\overline{\mu}_i^0 - \underline{\mu}_i^0)\left(\frac{N}{16\epsilon}\right)^{(p-1)/2} \right\rfloor$, these $k+1$ sub-rectangles must be deleted in Step 4. Thus, the number of iterations at which SOCPRBBRA terminates is at most $\left\lfloor \prod_{i=1}^{p-1}(\overline{\mu}_i^0 - \underline{\mu}_i^0)\left(\frac{N}{16\epsilon}\right)^{(p-1)/2} \right\rfloor$. This completes the proof. □

Remark 3. *Theorem 4 reveals that when* SOCPRBBRA *finds a global ϵ-optimal solution for* SLR*, the computational time required is at most*

$$2T\left\lfloor \prod_{i=1}^{p-1}(\overline{\mu}_i^0 - \underline{\mu}_i^0)\left(\frac{N}{16\epsilon}\right)^{(p-1)/2} \right\rfloor$$

seconds, where T denotes the upper bounds of the time required to solve a SOCP problem SOCP*(H) (see Section 2.3).*

Remark 4. *Theorem 4 sufficiently guarantees that* SOCPRBBRA *completes termination in a finite number of iterations because of the existence of this most extreme number of iterations.*

3. Numerical experiments

To verify the effectiveness and feasibility of SOCPRBBRA, we compared it with the algorithms in [1, 2, 31, 33] and the commercial solver BARON [37]. The corresponding codes were compiled and run on Matlab(2023a) and a series of numerical experiments were performed. All calculations were carried out on a desktop computer with Win7 operating system and an Intel(R) Core(TM) i5-8500 3.00 GHz power processor and 8 GB of memory. Besides, all linear programming and second-order cone programming problems are addressed by linprog and coneprog solvers in Matlab.

In all experiments of solving the random instances generated by Problem 1, the same tolerance ϵ adopted by all algorithms is 10^{-6}. For each set of parameters (p, m, n), ten random instances of the same size are generated and solved by related algorithms, and the average numerical results are recorded in Tables 1 and 2. The symbols in the header line of these tables are interpreted as: CPU: The average CPU running time after solving ten test instances by an algorithm; Iter: the average number of iterations after solving ten test instances by an algorithm; Opt.val: the average optimal value after solving ten test instances by an algorithm.

Problem 1.

$$\begin{cases} \min \ \displaystyle\sum_{i=1}^{p} \frac{\sum_{j=1}^{n} d_{ij}x_j + g_i}{\sum_{j=1}^{n} c_{ij}x_j + h_i} \\ \text{s.t.} \ \displaystyle\sum_{j=1}^{n} a_{kj}x_j \le b_k, \ k = 1, 2, \cdots, m \\ \qquad x_j \ge 0.0, \quad j = 1, 2, \cdots, n. \end{cases}$$

where all d_{ij}, c_{ij}, b_k and a_{kj} are randomly generated in [0,10]; all g_i and h_i are randomly generated in [0,1].

From the numerical results of Tables 1 and 2, all optimal values of SOCPRBBRA are not much different from other algorithms, especially compared with the commercial software package BARON. Moreover, SOCPRBBRA is optimal in both the number of iterations and CPU time, which reflects the excellent computational ability of our algorithm in solving the SLR problem.

A fact that can be observed from Table 1 is that BARON seems to be only suitable for solving some small-scale problems, and can clearly know that problems with $(p, m, n) = (2, 5, 2000), (2, 5, 3000), (2, 5, 5000), (3, 5, 2000)$ cannot be handled by the package within 3600s. Nevertheless, it is not difficult to find from Tables 1 and 2 that the number p of fractions is a major factor affecting the computational performance of our algorithm. However, compared with the algorithms in [1,2,31,33], this effect is relatively small. Particularly, the numerical results in Table 2 reveal that the algorithm in [33] cannot handle problems with $(p, m, n) = (4, 100, 500), (4, 140, 700)$ and $(5, 100, 500)$ in 3600s. Thus, SOCPRBBRA is expected to become a potential solver for solving some specific SLR problems.

In summary, although the number of fractional terms affects the computational performance of SOCPRBBRA, the computational ability of the algorithm is stronger than that of BARON and the algorithms in [1, 2, 31, 33] when solving specific SLR problems. Moreover, our algorithm may be more suitable for solving SLR problems with fewer linear fractional terms, especially for some large-scale problems.

Table 1. Computational comparisons among SOCPRBBRA, BARON and the algorithms in [2,31] for Problem 1 with $m = 5$.

(p, m, n)	SOCPRBBRA			Ref. [2]			Ref. [31]			BARON		
	Iter	CPU	Opt.val	Iter	CPU	Opt.val	Iter	CPU	Opt.val	Iter	CPU	Opt.val
(2,25)	3.7	0.109	10.2331	32.2	0.579	10.2331	209.2	3.672	10.2331	342.2	1.110	10.2331
(2,50)	3.8	0.038	14.6885	22.4	0.446	14.6885	133.3	2.399	14.6885	287.6	1.113	14.6885
(2,100)	9.6	0.098	10.4083	30.2	0.644	10.4083	241.8	4.883	10.4083	325.4	1.938	10.4083
(2,300)	6.6	0.138	71.5596	133.8	3.743	71.5596	241.1	8.079	71.5596	306.6	19.588	71.5596
(2,500)	7.8	0.231	28.5955	135.0	5.533	28.5954	386.0	19.916	28.5955	234.2	32.980	28.5954
(2,1000)	9.2	0.639	16.4638	85.4	7.991	16.4638	921.4	129.494	16.4639	7513.2	221.180	16.4638
(2,2000)	9.8	1.224	27.9538	296.6	88.716	27.9538	406.0	195.906	27.9539	–	–	–
(2,3000)	10.8	2.300	30.7365	261.8	182.034	30.7365	347.2	388.677	30.7365	–	–	–
(2,5000)	10.5	6.967	20.3183	259.7	563.471	20.3183	316.7	1042.680	20.3183	–	–	–
(3,25)	20.4	0.171	12.0348	80.2	1.249	12.0348	232.2	3.491	12.0348	448.2	1.738	12.0348
(3,50)	26.0	0.246	9.2501	203.4	3.213	9.2501	320.0	5.155	9.2501	655.4	2.870	9.2501
(3,100)	30.8	0.344	37.5424	286.0	5.037	37.5425	437.2	8.344	37.5425	9481.8	28.846	37.5424
(3,300)	27.2	0.618	19.6112	306.5	8.720	19.6112	1709.2	58.863	19.6112	6871.4	132.248	19.6112
(3,500)	44.4	5.189	19.2796	274.0	16.396	19.2797	1416.0	95.242	19.2797	7650.4	171.470	19.2796
(3,1000)	20.7	1.561	25.9518	152.6	22.036	25.9518	968.0	167.915	25.9519	9847.2	320.890	25.9518
(3,2000)	29.6	4.159	41.5536	1827.4	711.645	41.5534	2441.2	1425.414	41.5531	–	–	–
(4,25)	46.6	0.423	17.0477	300.0	4.693	17.0478	664.4	10.417	17.0477	1245.0	4.772	17.0477
(4,50)	49.6	0.589	20.7630	388.2	6.426	20.7630	892.8	14.924	20.7631	16828.4	107.098	20.7629
(4,100)	67.8	0.861	22.9271	6443.8	118.457	22.9273	503.2	10.365	22.9272	29978.4	104.886	22.9271
(4,300)	138.2	3.938	44.8316	1310.0	49.521	44.8318	1785.8	72.131	44.8313	9476.2	115.400	44.8311
(4,500)	111.4	7.506	20.4091	7008.4	392.185	20.4091	2202.0	154.648	20.4092	12791.6	381.632	20.4051
(4,1000)	122.4	10.826	36.0531	2995.6	493.966	36.0574	1662.0	348.335	36.0574	1795.8	357.048	36.0572
(5,25)	88.2	0.916	18.0635	6467.0	110.256	18.0635	1470.0	23.007	18.0636	3047.4	13.856	18.0635
(5,50)	121.8	1.430	112.1293	820.8	13.552	112.1294	2895.0	48.051	112.1318	22175.0	182.518	112.1293
(5,75)	589.2	8.338	25.6361	667.7	12.919	25.6362	557.6	11.029	25.6362	2816.2	21.212	25.6361
(5,100)	447.6	7.258	18.4318	1138.6	22.054	18.4318	1764.8	35.983	18.4318	4263.8	38.404	18.4317
(5,300)	586.8	20.066	51.9267	5177.4	176.973	51.9297	984.6	43.367	51.9291	1777.4	49.352	51.9287
(6,25)	273.2	3.733	18.5081	1291.5	20.626	18.5081	658.6	10.266	18.5083	4481.4	28.206	18.5081
(6,50)	99.0	1.497	38.6333	66966.4	1258.696	38.6347	2914.6	50.151	38.6364	13926.0	137.688	38.6347
(6,100)	200.8	3.900	35.0379	1790.8	35.522	35.0386	2417.8	52.226	35.0386	15529.4	189.494	35.0380
(7,25)	577.4	8.135	65.8240	5949.4	98.132	65.8241	860.2	13.738	65.8315	4233.8	30.704	65.8240

Table 2. Computational comparisons among SOCPRBBRA and the algorithms in Refs [1, 33] for Problem 1 .

(p,m,n)	SOCPRBBRA			Ref. [1]			Ref. [33]		
	Iter	CPU	Opt.val	Iter	CPU	Opt.val	Iter	CPU	Opt.val
(2,5,25)	4.1	0.0511	10.6717	10.0	0.3393	10.6717	486.4	14.9469	10.6717
(2,10,50)	6.6	0.0928	27.0132	14.6	0.3042	27.0132	538.2	5.1878	27.0132
(2,20,100)	5.5	0.1246	5.7241	20.1	0.3643	5.7241	873.5	11.7983	5.7241
(2,60,300)	7.3	1.4472	11.7670	31.9	4.1865	11.7670	398.0	33.5582	11.7670
(2,100,500)	6.0	5.7734	14.1794	21.0	14.2905	14.1794	259.7	99.0577	14.1794
(2,140,700)	5.7	11.5332	4.9275	17.3	7.5279	4.9275	77.2	67.8522	4.9275
(2,200,1000)	6.3	35.0878	6.2781	20.2	90.5737	6.2781	213.4	535.7422	6.2781
(3,5,25)	15.6	0.3541	10.9691	29.1	1.0736	10.9691	1091.9	8.9137	10.9691
(3,10,50)	26.4	0.4569	27.3195	461.6	5.9316	27.3195	7546.8	79.8372	27.3195
(3,20,100)	21.5	0.4950	10.3249	69.9	1.4715	10.3249	9261.5	137.6806	10.3249
(3,60,300)	24.4	5.1643	10.0868	90.1	12.2862	10.0868	30913.6	2656.9443	10.0868
(3,100,500)	19.9	5.9031	10.9452	98.8	137.0207	10.9452	2364.8	1144.3356	10.9452
(3,140,700)	20.8	46.5581	15.6849	169.7	292.8312	15.6849	2243.5	1809.0920	15.6849
(3,200,1000)	21.9	127.9725	6.6388	119.5	573.9409	6.6388	932.1	2393.1107	6.6388
(4,5,25)	42.7	0.5303	20.7392	2047.7	38.6920	20.7392	12789.1	116.9910	20.7392
(4,10,50)	72.5	1.2027	23.6421	134.1	1.9314	23.6421	53546.3	715.2585	23.6421
(4,20,100)	59.3	1.7275	11.6142	763.3	24.7755	11.6142	31351.7	667.6312	11.6142
(4,60,300)	58.7	21.7087	19.3050	320.9	68.0013	19.3050	9373.6	1074.9317	19.3050
(4,100,500)	49.4	47.8845	11.9962	322.0	193.6454	11.9962	–	–	–
(4,140,700)	41.2	153.1522	6.6882	630.7	1425.6532	6.6882	–	–	–
(5,5,25)	119.8	1.6097	27.3708	17176.8	543.1353	39.9423	12604.2	151.9283	33.6565
(5,10,50)	103.0	1.9965	14.4297	12746.7	164.5762	14.4297	61295.9	764.7003	14.4297
(5,20,100)	78.4	2.4796	21.3268	397.5	9.1383	21.3268	47865.2	836.0249	21.3268
(5,60,300)	84.7	45.3486	21.3502	369.0	101.5605	21.3502	20321.8	3491.9032	21.3502
(5,100,500)	116.8	113.4898	45.4468	482.4	319.6263	45.4471	–	–	–

4. Conclusions

In this paper, we study the SLR problem. By employing a new equivalent transformation technique, SLR is transformed into the problem EP. Then we reconstruct and relax these non-convex constraints, so that the non-convex relaxation subproblem of EP is obtained. Furthermore, a new global optimization algorithm SOCPRBBRA is constructed by combining non-convex relaxations with B&B technique. The branching operations of the algorithm takes place in the space $\mathbb{R}^{p-1}$, rather than space $\mathbb{R}^p$, $\mathbb{R}^{2p}$ or $\mathbb{R}^n$, which greatly saves the computational workload of the algorithm. A large number of numerical results show that SOCPRBBRA not only has stronger computing power than the commercial software package BARON, but also has higher computational efficiency than the four existing B&B algorithms (i.e., algorithms in [1, 2, 31, 33]) when solving some specific SLR problems. The future work is to extend our algorithm to generalized nonlinear fractional programming problems.

Author contributions

Bo Zhang: Formal analysis, investigation, resources, methodology, writing-original draft, validation, data curation, and funding acquisition; Yuelin Gao: Formal analysis, invesigation, writing-review & editing, software, data curation; Ying Qiao: Conceptualization, supervision, project administration; Ying Sun: Project administration, methodology, validation, and formal funding acquisition. All authors have read and agreed to the published version of the manuscript.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research is supported by the National Natural Science Foundation of China under Grant [grant number 12301401] and the Basic discipline research projects supported by Nanjing Securities [grant number NJZQJCXK202201].

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. B. Zhang, Y. Gao, X. Liu, X. Huang, A new deterministic global computing algorithm for solving a kind of linear fractional programming, *Optimization,* **72** (2023), 1485–1531. http://dx.doi.org/10.1080/02331934.2022.2027940
2. B. Zhang, Y. Gao, An output-space based branch-and-bound algorithm for sum-of-linear-ratios problem, *Asia Pac. J. Oper. Res.,* **42** (2023), 2250010. http://dx.doi.org/10.1142/S0217595922500105
3. S. Schaible, *Fractional programming*, In: Horst, R., Pardalos, P.M. (eds) Handbook of Global Optimization. Nonconvex Optimization and Its Applications, Boston: Springer Press, 1995. http://dx.doi.org/10.1007/978-1-4615-2025-2_10
4. H. Konno, H. Watanabe, Bond portfolio optimization problems and their applications to index tracking: A partial optimization approach, *J. Oper. Res. Soc. Jpn.,* **39** (2017), 295–306. http://dx.doi.org/10.2307/3010378
5. H. Konno, M. Inori, Bond portfolio optimization by bilinear fractional programming, *J. Oper. Res. Soc. Jpn.,* **32** (1989), 143–158. http://dx.doi.org/10.2307/2583552
6. C. Colantoni, R. Manes, A. Whinston, Programming, profit rates and pricing decisions, *Account. Rev.,* **44** (1969), 467–481. http://dx.doi.org/10.2307/244427

7. B. Sawik, *Downside risk approach for multi-objective portfolio optimization*, In: Klatte, D., Luthi, HJ., Schmedders, K. (eds) Operations Research Proceedings 2011, Berlin: Springer Press, 2012. http://dx.doi.org/10.1007/978-3-642-29210-1_31
8. A. Billionnet, Mathematical optimization ideas for biodiversity conservation, *Eur. J. Oper. Res.,* **231** (2013), 514–534. http://dx.doi.org/10.1016/j.ejor.2013.03.025
9. C. Kao, Network data envelopment analysis: A review, *Eur. J. Oper. Res.,* **239** (2014), 1–16. http://dx.doi.org/10.1016/j.ejor.2014.02.039
10. T. Kuno, T. Masaki, A practical but rigorous approach to sum-of-ratios optimization in geometric applications, *Comput. Optim. Appl.,* **54** (2013), 93–109. http://dx.doi.org/10.1007/s10589-012-9488-5
11. A. Fakhri, M. Ghatee, Minimizing the sum of a linear and a linear fractional function applying conic quadratic representation: Continuous and discrete problems, *Optimization,* **65** (2016), 1023–1038. http://dx.doi.org/10.1080/02331934.2015.1113532
12. T. Matsui, NP-hardness of linear multiplicative programming and related problems, *J. Global Optim.,* **9** (1996), 113–119. http://dx.doi.org/10.1007/BF00121658
13. B. Ozkok, An iterative algorithm to solve a linear fractional programming problem, *Comput. Ind. Eng.,* **140** (2020), 106234. http://dx.doi.org/10.1016/j.cie.2019.106234
14. A. Charnes, W. Cooper, Programming with linear fractional functionals, *Nav. Res. Log.,* **9** (1962), 181–186. http://dx.doi.org/10.1002/nav.3800150308
15. H. Konno, Y. Yajima, T. Matsui, Parametric simplex algorithms for solving a special class of nonconvex minimization problems, *J. Global Optim.,* **1** (1991), 65–81. http://dx.doi.org/10.1007/BF00120666
16. Y. Xia, L. Wang, X. Wang, Globally minimizing the sum of a convex-concave fraction and a convex function based on wave-curve bounds, *J. Global Optim.,* **77** (2020), 301–318. http://dx.doi.org/10.1007/s10898-019-00870-2
17. Y. Nesterov, A. Nemirovskii, An interior-point method for generalized linear-fractional programming, *Math. Program.,* **69** (2003), 177–204. http://dx.doi.org/10.1007/BF01585557
18. H. Konno, N. Abe, Minimization of the sum of three linear fractional functions, *J. Global Optim.,* **15** (1999), 419–432. http://dx.doi.org/10.1023/A:1008376731013
19. H. Benson, On the global optimization of sums of linear fractional func- tions over a convex set, *J. Optim. Theory Appl.,* **121** (2004), 19–39. http://dx.doi.org/10.1023/B:JOTA.0000026129.07165.5a
20. P. Shen, B. Huang, L. Wang, Range division and linearization algorithm for a class of linear ratios optimization problems, *J. Comput. Appl. Math.,* **350** (2019), 324–342. http://dx.doi.org/10.1016/j.cam.2018.10.038
21. J. Falk, S. Palocsay, Image space analysis of generalized fractional programs, *J. Global Optim.,* **4** (1994), 63–88. http://dx.doi.org/10.1007/BF01096535
22. N. Phuong, H. Tuy, A unified monotonic approach to generalized linear fractional programming, *J. Global Optim.,* **26** (2003), 229–259. http://dx.doi.org/10.1007/BF01096535
23. H. Konno, H. Yamashita, Minimizing sums and products of linear fractional functions over a polytope, *Nav. Res. Log.,* **46** (1999), 583–596. http://dx.doi.org/10.1002/(SICI)1520-6750(199908)46:5¡583::AID-NAV8¿3.0.CO;2-5

24. H. Benson, A simplicial branch and bound duality-bounds algorithm for the linear sum-of-ratios problem, *Eur. J. Oper. Res.,* **182** (2007), 597–611. http://dx.doi.org/10.1016/j.ejor.2006.08.036
25. H. Benson, Branch-and-bound outer approximation algorithm for sum-of-ratios fractional programs, *J. Optim. Theory Appl.,* **146** (2010), 1–18. http://dx.doi.org/10.1007/s10957-010-9647-8
26. J. Carlsson, J. Shi, A linear relaxation algorithm for solving the sum-of-linear-ratios problem with lower dimension, *Ope. Res. Lett.,* **41** (2013), 381–389. http://dx.doi.org/10.1016/j.orl.2013.04.005
27. H. Jiao, B. Li, Y. Shang, An outer space approach to tackle generalized affine fractional program problems, *J. Optim. Theory Appl.,* **201** (2024), 1–35. http://dx.doi.org/10.1007/s10957-023-02368-0
28. H. Jiao, B. Li, W. Yang, A criterion-space branch-reduction-bound algorithm for solving generalized multiplicative problems, *J. Global Optim.,* **89** (2024), 597–632. http://dx.doi.org/10.1007/s10898-023-01358-w
29. A. Ashtiani, A. Paulo, A branch-and-cut algorithm for a class of sum-of-ratios problems, *Appl. Math. Comput.,* **268** (2015), 596–608. http://dx.doi.org/10.1016/j.amc.2015.06.089
30. H. Jiao, S. Liu, A practicable branch and bound algorithm for sum of linear ratios problem, *Eur. J. Oper. Res.,* **243** (2015), 723–730. http://dx.doi.org/10.1016/j.ejor.2015.01.039
31. S. Liu, L. Ge, An outcome space algorithm for minimizing a class of linear ratio optimization problems, *Comput. Appl. Math.,* **40** (2021), 225. http://dx.doi.org/10.1007/s40314-021-01614-3
32. X. Liu, Y. Gao, B. Zhang, A new global optimization algorithm for a class of linear fractional programming, *Mathematics,* **7** (2019), 867. http://dx.doi.org/10.3390/math7090867
33. H. Jiao, J. Ma, An efficient algorithm and complexity result for solving the sum of general affine ratios problem, *Chaos Soliton. Fract.,* **164** (2022), 112701. http://dx.doi.org/10.1016/j.chaos.2022.112701
34. P. Shen, Y. Wang, D. Wu, A spatial branch and bound algorithm for solving the sum of linear ratios optimization problem, *Numer. Algorithms,* **93** (2023), 1373–1400. http://dx.doi.org/10.1007/s11075-022-01471-z
35. I. Stancu-Minasian, A ninth bibliography of fractional programming, *Optimization,* **68** (2019), 2125–2169. http://dx.doi.org/10.1080/02331934.2019.1632250
36. P. Shen, T. Lu, Regional division and reduction algorithm for minimizing the sum of linear fractional functions, *J. Inequal. Appl.,* **2018** (2018), 63. http://dx.doi.org/10.1186/s13660-018-1651-9
37. A. Khajavirad, N. Sahinidis, A hybrid LP/NLP paradigm for global optimization relaxations, *Math. Prog. Comput.,* **10** (2018), 383–421. http://dx.doi.org/10.1007/s12532-018-0138-5