Mathematics

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

AIMS Mathematics, 10(9): 22650–22677.
DOI: 10.3934/math.20251008
Received: 02 August 2025
Revised: 09 September 2025
Accepted: 24 September 2025
Published: 29 September 2025

Research article

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

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

[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:** E-mail: zbsdx121@163.com; Tel: +8618709602837.

Abstract: This paper investigated the linear ratio sum problem, a complex non-convex optimization problem with extensive applications in finance, economics, computer vision, and other fields. We proposed a novel global optimization approach that reformulated the original problem into an equivalent one with nonlinear constraints. The approach constructed linear fractional relaxation subproblems via constraint relaxation and leveraged the structural properties of the relaxations to transform these subproblems into linear programming formulations, thereby ensuring efficient computation. Furthermore, rectangular branching rules were designed based on the relaxed nonlinear constraints. These rules, complemented by region elimination techniques, accelerated convergence by exploiting the structure of the objective function. By integrating these components into a branch-and-bound framework, a novel global optimization algorithm was devised. Theoretical analysis confirmed the convergence and computational complexity of the proposed algorithm, while numerical tests validated its effectiveness and feasibility.

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

1. Introduction

This paper examines the sum-of-linear-ratios (SLR) optimization problem:

$$\text{(SLR)}\begin{cases} \min\ \varphi(x) = \sum_{i=1}^{p} \frac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}}, \\ \text{s.t.}\ \ x \in \mathcal{X} := \{x \in \mathbb{R}^n | Ax \leq b\}, \end{cases}$$

where $p \geq 2$, $\mathbf{c}_i, \mathbf{d}_i \in \mathbb{R}^n$, $c_{0i}, d_{0i} \in \mathbb{R}$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, x is confined to a compact feasible set $\mathcal{X}$, and each denominator $\mathbf{d}_i^\top x + d_{0i}$ remains strictly positive over $\mathcal{X}$ without loss of generality (see [1]).

Many studies have focused on developing algorithms for SLR problems, driven by their extensive applications across various domains including finance [2–4], system engineering [5], network data envelopment analysis [6], and computer vision [7]. Nonetheless, addressing SLR problems is challenging due to their inherent NP-hard complexity [8] (NP: Nondeterministic Polynomial Time) and the presence of multiple local optima, which complicates achieving global optimization. This NP-hardness complicates finding global optima in polynomial time. Thus, efficient approximate or exact algorithms have important research value.

The complexity of the SLR problem increases with the parameter p. When $p = 1$, the problem can be defined as a single linear fractional program. Its solution can be obtained through the Charnes-Cooper transformation (CCT) [9] or the continuity-based iterative method [10]. The former directly converts the original problem into a linear program, while the latter requires an iterative process to reach the solution. For the case of $p = 2$, Konno and Yajima [11] reduced the number of linear fractions from two to one via the CCT and further developed a single-parameter simplex method. In contrast, by relying on the strong duality of convex optimization problems under the Slater condition, Xia et al. [12] proposed a branch-and-bound (BB) algorithm with a sawtooth waveform curve bounding strategy. In their method, the relaxed subproblems can be solved by exploiting analytical properties without invoking a solver. For $p > 2$, several methods have been suggested, including the interior point method [13], heuristic approach [14], concave minimization method [15], polynomial time approximation algorithm [16], image-space analysis method [17], monotone method [18], outer approximation algorithm [19], and various BB strategies [20–22]. Among these, the BB algorithm stands as a commonly employed method in global optimization, with broad application to solving complex optimization problems. Its core framework comprises three key operations, namely, branching, bounding, and pruning. These operations are inherently coupled throughout the iterative process and collectively determine the computational efficiency of the algorithm. This study aims to propose a BB algorithm for the SLR problem with $p > 1$, where the primary innovation resides in the bounding operation.

In the existing literature on BB algorithms for the SLR problem, the mainstream approach involves reformulating the original problem into an equivalent structure using various techniques, upon which relaxation strategies such as linear relaxation (LR), quadratic convex relaxation (QCR), or second-order cone relaxation (SOCR) are designed for bounding. Nevertheless, research on the branching and pruning steps remains relatively limited. At the same time, the spatial dimension on which branching operations act exerts a notable influence on the computational efficiency of BB algorithms, a phenomenon validated across various relevant literatures as demonstrated in References [1, 23–26]. In response to these limitations, academic research efforts on BB algorithms for SLR problems exhibit notable differences in terms of the branching space and branching objects. Early research explored branching spaces of various dimensions rather than being confined to the original decision space. Among such studies, Benson [20] proposed a simplex branching dual bounding algorithm for globally solving the SLR problem, which performs branching in a p-dimensional simplex space where the reciprocals of denominators lie. Bounding is achieved by constructing LR subproblems via Lagrange duality in this space, although the algorithm requires storing a large number of simplex vertices. Later, Benson [21] reformulated the SLR problem as an indefinite quadratic program and proposed a hybrid BB algorithm integrated with outer approximation techniques, where branching occurs over a $2p$-dimensional rectangle. While this method avoids vertex computation by solving convex programming relaxations, it increases computational complexity. Also, in the field of generalized sum-of-fractions optimization, Ashtiani and Paulo [27] proposed a cutting plane algorithm incorporating a BB subroutine with an LR strategy. This algorithm requires repeatedly invoking the BB algorithm, whose branching operations also take place in a $2p$-dimensional rectangular space, leading to substantial computational burden. During the same period, other types of research focused on the original decision space. Kuno and Masaki [7], as well as Carlsson and Shi [22], each proposed a BB algorithm utilizing the LR strategy. Specifically, the LR problem of the former requires solving p linear programs with n variables, while that of the latter requires solving a linear program with $(n + 1)p$ variables. Both algorithms take the original n-dimensional decision space of the SLR problem as the branching space and n-dimensional rectangles as branching objects. However, as the number of variables increases, branching in high-dimensional spaces leads to a considerable decline in algorithmic efficiency. Thus, these methods are more suitable for SLR problems with large p and small n.

In recent years, many researchers have further advanced the concept of the outer space branch, constructed low-dimensional branch spaces through different mappings, and taken rectangles as the branching objects. This direction has been pursued by Jiao and Liu [23], Liu and Ge [28], Liu et al. [24], and Jiao and Ma [29]. These researchers transformed the SLR problem into equivalent models with bilinear constraints or simplified fractional objectives, and developed various LR methods that improve upon bounding strategies based solely on the decision variable space. Additionally, the application of the CCT was further extended during this stage. Its underlying concept aligns with that introduced in [11], enabling a reduction in the number of linear ratio terms from p to $p - 1$. This technique has been adopted both by our research team [1, 25, 26] and by Shen et al. [30]. The resulting BB algorithms adopt a $(p - 1)$-dimensional space for branching, thereby simplifying the spatial dimension and reducing computational complexity. Among these, References [25, 26] refer to BB algorithms based on different LR strategies, while [1] and [30] both employ SOCR. The SOCR in [30] can be viewed as a special case of that in [1] when the parameter is set to 1. More recently, building upon branching in $(p - 1)$-dimensional space, researchers have optimized the performance of BB algorithms under certain conditions through adaptive branching strategies and the integration of multiple relaxation and bounding techniques. For example, Huang and Shen [31], extending the work of Shen et al. [30], proposed an enhanced QCR and a novel adaptive branching rule. Their method improves computational efficiency while maintaining the advantages of a lower-dimensional branching space. Luo et al. [32] integrated three strategies, continuous linear optimization, LR, and branch-and-cut, into a BB algorithm for SLR problems. The first two strategies significantly accelerate the tightening of upper and lower bounds, respectively.

Deng and Shen [33] proposed an adaptive BB reduction algorithm that uses LR for lower bound estimation and eliminates invalid subspaces via a region reduction strategy. This approach effectively screens subspaces after branching and indirectly reduces redundant branching operations. Following this direction, in this study, we propose a new BB algorithm. This algorithm also branches on a $(p - 1)$-dimensional rectangle and incorporates LR. Our contributions focus on the new equivalent problem and the new LR, aiming to improve the algorithm efficiency in certain SLR problem scenarios.

In this study, we present a novel BB algorithm to solve the SLR problem globally. The main idea is to transform the SLR into an equivalent problem (EP) with a nonlinear objective function and $p - 1$ nonlinear constraints by introducing intermediate variables. Then, we propose a new linear fractional relaxation method based on relaxing these nonlinear constraints. We also propose an acceleration technique for EP problems with this special structure of the objective function. As discussed above, some works in our research group [1, 25, 26] and Shen et al. [30] used the CCT to reduce the number of linear fractions in the SLR from p to $p - 1$ so that the corresponding BB algorithms branch in $p - 1$-dimensional space. In contrast to these works, we solve the problem from another perspective by using the new equivalent problem to construct a linear fractional relaxation. The linear fractional relaxation subproblem is then transformed into a linear programming formulation, which reduces the complexity. In this paper, we propose a new global optimization algorithm by using linear fractional relaxation and acceleration techniques within the BB framework. Our method is able to find a global ϵ-optimum of the EP within the given tolerance. After proving the convergence of our algorithm, we analyze its computational complexity and derive an upper bound of $\left\lfloor \prod_{i=1}^{p-1} \frac{(p-1)\left(1-\frac{L_i}{U_i}\right)(\overline{\mu}_i^0-\underline{\mu}_i^0)}{\epsilon} \right\rfloor$ for the worst-case number of iterations, which gives a quantitative characterization of the complexity. Finally, we conduct numerical experiments to verify the effectiveness of our algorithm. The compared algorithms include the linear relaxation BB algorithms proposed in [25, 29, 33], the second-order cone relaxation BB algorithm [1] and the global optimization solver BARON [34] (BARON: Branch And Reduce Optimization Navigator). The results show that for medium- and large-scale SLR instances, our algorithm can find global ϵ-optimums within acceptable computing time, and its performance is significantly better than that of both the compared BB algorithms and the BARON solver. Additionally, we utilize a cost optimization problem [35] in hospital management to validate the practical applicability of our algorithm.

The structure of the remainder of this paper is outlined below. Section 2 explores the theoretical foundations of the proposed algorithm, covering topics such as the equivalent problem, boundary operations, branching, region elimination methods, the full algorithmic process, and computational complexity analysis. Section 3 is the numerical experiment, accompanied by calculation results and a detailed analysis of the numerical results. Finally, we summarize the findings of this research and propose avenues for future investigation.

2. Theoretical framework

This section introduces a new BB algorithm designed to solve the SLR problem globally. The approach incorporates a linear fractional relaxation strategy, an adaptive branching rule, and a region elimination rule.

2.1. Equivalent problem

In this section, we introduce a novel equivalent formulation for tackling the SLR problem using the BB algorithm. Specifically, our formulation targets the SLR problem, which is

$$
\text{(EP)}\begin{cases} \min\ \phi(x,\mu) = \sum_{i=1}^{p-1}\mu_i + \dfrac{\mathbf{c}_p^\top x + c_{0p}}{\mathbf{d}_p^\top x + d_{0p}}, \\ \text{s.t.}\ \dfrac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}} = \mu_i, i \in I := \{1, 2, \cdots, p-1\}, \\ \quad x \in \mathcal{X}, \end{cases}
$$

where $\mu = (\mu_1, \mu_2, \cdots, \mu_{p-1})^\top \in \mathbb{R}^{p-1}$. As a result, the subsequent theorem is self-evident.

Theorem 2.1. *A vector $x^* \in \mathbb{R}^n$ is a global minimizer of the* SLR *problem if and only if (x^*, μ^*) solves* EP *globally, with $\mu_i^* = \frac{\mathbf{c}_i^\top x^* + c_{0i}}{\mathbf{d}_i^\top x^* + d_{0i}}$ where $i \in I$.*

Proof. Suppose x^* is a global minimizer of the SLR problem, meaning $\varphi(x^*) \le \varphi(x)$ for all $x \in \mathcal{X}$. Define $\mu_i^* = \frac{c_i^\top x^* + c_{0i}}{d_i^\top x^* + d_{0i}}$ for $i \in I$. On the one hand, since x^* is a feasible solution to the SLR problem, $x^* \in \mathcal{X}$, which satisfies the constraints on the decision variable x in the EP problem. From the definition of μ_i^*, it can be directly verified that μ_i^* satisfies the equality constraints of the EP problem. Furthermore, since $\underline{\mu}_i^0$ and $\overline{\mu}_i^0$ denote the infimum and supremum of the fractional function $\frac{c_i^\top x + c_{0i}}{d_i^\top x + d_{0i}}$ over $\mathcal{X}$, respectively, we have $\mu_i^* \in [\underline{\mu}_i^0, \overline{\mu}_i^0]$, i.e., $\mu^* \in \mathcal{H}^0$. Thus, (x^*, μ^*) is a feasible solution to the EP problem. On the other hand, take any feasible solution (x, μ) of the EP problem. By the equality constraints of the EP problem, we have $\mu_i = \frac{c_i^\top x + c_{0i}}{d_i^\top x + d_{0i}}$ for $i \in I$. Substituting this into the objective function of the EP problem gives:

$$
\phi(x,\mu) = \sum_{i=1}^{p-1}\mu_i + \frac{c_p^\top x + c_{0p}}{d_p^\top x + d_{0p}} = \sum_{i=1}^{p}\frac{c_i^\top x + c_{0i}}{d_i^\top x + d_{0i}} = \varphi(x). \tag{2.1}
$$

Similarly, it can be proven that:

$$
\phi(x^*,\mu^*) = \sum_{i=1}^{p-1}\mu_i^* + \frac{c_p^\top x^* + c_{0p}}{d_p^\top x^* + d_{0p}} = \sum_{i=1}^{p}\frac{c_i^\top x^* + c_{0i}}{d_i^\top x^* + d_{0i}} = \varphi(x^*). \tag{2.2}
$$

Since x^* is a global minimizer of the SLR problem, $\varphi(x^*) \le \varphi(x)$, which implies $\phi(x^*, \mu^*) \le \phi(x, \mu)$ for all feasible solutions (x, μ) of the EP problem. Therefore, (x^*, μ^*) is a global minimizer of the EP problem.

Suppose (x^*, μ^*) is a global minimizer of the EP problem, meaning $\phi(x^*, \mu^*) \le \phi(x, \mu)$ for all feasible solutions (x, μ) of the EP problem. From the constraints of the EP problem, we know $x^* \in \mathcal{X}$, so x^* is a feasible solution to the SLR problem. Take any feasible solution $x \in \mathcal{X}$ of the SLR problem, and define $\mu_i = \frac{c_i^\top x + c_{0i}}{d_i^\top x + d_{0i}}$ for $i \in I$. By the definitions of $\underline{\mu}_i^0$ and $\overline{\mu}_i^0$, we have $\mu_i \in [\underline{\mu}_i^0, \overline{\mu}_i^0]$, i.e., $\mu \in \mathcal{H}^0$. Thus, (x, μ) is a feasible solution to the EP problem. By the global optimality of the EP problem, $\phi(x^*, \mu^*) \le \phi(x, \mu)$. Similar to Eqs (2.1) and (2.2), substituting μ_i and μ_i^* into their respective objective functions yields

$\phi(x,\mu) = \varphi(x)$ and $\phi(x^*,\mu^*) = \varphi(x^*)$. Therefore, $\varphi(x^*) \le \varphi(x)$ for all feasible solutions x of the SLR problem, which means x^* is a global minimizer of the SLR problem.

In conclusion, x^* is a global minimizer of the SLR problem if, and only if, (x^*,μ^*) (where $\mu_i^* = \frac{c_i^\top x^* + c_{0i}}{d_i^\top x^* + d_{0i}}$ for $i \in I$) is a global minimizer of the EP problem. This completes the proof. □

Remark 2.1. *To clarify the differences between the aforementioned equivalent transformation method and the three previously published works by the authors [1, 25, 26], a detailed comparison is provided in Table 1. As illustrated in the table, the EP construction phase in References [25, 26] relies on* CCT. *In contrast, Reference [1] and this study construct their EP without resorting to* CCT*; instead, this phase is built upon the original variable x, where fraction splitting/replacement is facilitated solely by μ_i. Furthermore, the variables of* EP *in References [25, 26] are (w,t) or (y,t), both of which are transformed via* CCT. *These variables necessitate establishing an indirect correlation with the original variable x through transformation relations, and their equivalence hinges on transformation transitivity. Conversely, the variables from the* EP *in Reference [1] and this work are derived directly from the original variable x, a design that yields a more direct equivalence to the original problem.*

Table 1. Comparison of the core structures of the four equivalent transformation methods.

Methods	Construction Logic	Objective Function	Key Constraints	Variable Dimension	Core Nonlinear Source
[25]	Two-stage + CCT: 1. $t = 1/(e_p^T x + f_p)$, $w = tx$ (convert p fractions to $p-1$ fractions + linear term); 2. Introduce $z_i = (c_i^T w + d_i t)/(e_i^T w + f_i t)$	$\sum_{i=1}^{p-1} z_i + c_p^T w + d_p t$	1. Nonlinear equality: $c_i^T w + d_i t = z_i(e_i^T w + f_i t)$; 2. Linear: $(w,t) \in W = \left\{(w,t) \in \mathbb{R}^{n+1} \middle\vert e_p^T w + f_p t = 1, Aw - bt \le 0, w \ge 0, t \ge 0\right\}$	$(w,t,z) \in \mathbb{R}^{n+p}$	Bilinear term $z_i(e_i^T w + f_i t)$
[26]	Two-stage + CCT: 1. Same CCT as [25]; 2. Introduce $s_i = c_i^T y + d_i t$, $v_i = e_i^T y + f_i t$, $z_i = 1/v_i$	$\sum_{i=1}^{p-1} s_i z_i + c_p^T y + d_p t$	1. Nonlinear equality: $v_i = 1/z_i$ (reciprocal); 2. Linear equality: $s_i = c_i^T y + d_i t$, $v_i = e_i^T y + f_i t$; 3. $(y,t) \in W$	$(s,v,z,y,t) \in \mathbb{R}^{3p+n-2}$	Reciprocal $v_i = 1/z_i$, bilinear term $s_i z_i$
[1]	Introduce μ_i and $\alpha_i > 0$ to split the original objective	$\sum_{i=1}^{p-1} \mu_i + \frac{c_p^T x + d_p - \sum_{i=1}^{p-1} \alpha_i (e_i^T x + f_i)}{e_p^T x + f_p}$	Nonlinear inequality: $\frac{c_i^T x + d_i}{e_i^T x + f_i} + \alpha_i \cdot \frac{e_i^T x + f_i}{e_p^T x + f_p} \le \mu_i$	$(x,\mu) \in \mathbb{R}^{n+p-1}$	Fraction combination inequality
ours	Introduce μ_i equal to each fraction directly	$\sum_{i=1}^{p-1} \mu_i + \frac{c_p^T x + c_{0p}}{d_p^T x + d_{0p}}$	Nonlinear equality: $\mu_i = \frac{c_i^T x + c_{0i}}{d_i^T x + d_{0i}}$	$(x,\mu) \in \mathbb{R}^{n+p-1}$	Fractional equation

Theorem 2.1 suggests that the SLR problem can be addressed by solving the corresponding EP. Furthermore, the optimal solution to the SLR problem is determined by the component x^* of the optimal pair (x^*,μ^*) for the EP. Accordingly, the subsequent analysis will primarily focus on the EP. The constraints

$$\frac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}} = \mu_i,\ i \in I \tag{2.3}$$

along with the objective function of the EP, evidently exhibit nonlinear characteristics, which contribute to the overall nonlinearity of the problem. To address this, we first investigate the relaxation of these constraints, while a thorough analysis of the objective function is deferred to Section 2.3. To facilitate this process, initial upper bounds $\overline{\mu}_i^0$ and lower bounds $\underline{\mu}_i^0$ need to be established, satisfying $\underline{\mu}_i^0 \le \mu_i^* := \frac{\mathbf{c}_i^\top x^* + c_{0i}}{\mathbf{d}_i^\top x^* + d_{0i}} \le \overline{\mu}_i^0$, and an initial rectangle $\mathcal{H}^0 := \prod_{i=1}^{p-1} \left[\underline{\mu}_i^0, \overline{\mu}_i^0\right]$ must be constructed. As a result, for each

$i \in I$, the following problems must be resolved:

$$\underline{\mu}_i^0 = \min_{x \in \mathcal{X}} \frac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}}, \quad \overline{\mu}_i^0 = \max_{x \in \mathcal{X}} \frac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}}, \tag{2.4}$$

where it is necessary to satisfy $\underline{\mu}_i^0 \leq \mu_i^* \leq \overline{\mu}_i^0$ for every $i \in I$. In fact, as demonstrated by Lemma 2.1 in [36], the problems described in Eq (2.4) can be effectively addressed by transforming them into the corresponding linear programming problems via the CCT.

Based on the foregoing analysis, it is deduced that the pair (x^*, μ^*) must satisfy both conditions $x^* \in \mathcal{X}$ and $\underline{\mu}_i^0 \leq \mu_i^* \leq \overline{\mu}_i^0$ for $i \in I$. Thus, the equivalent formulation of the EP can be articulated as follows:

$$\text{EP}(\mathcal{H}_0) \quad \begin{cases} \min \ \phi(x,\mu) = \sum_{i=1}^{p-1} \mu_i + \dfrac{\mathbf{c}_p^\top x + c_{0p}}{\mathbf{d}_p^\top x + d_{0p}} \\ \text{s.t. Eq (2.3)}, \quad x \in \mathcal{X}, \ \mu \in \mathcal{H}^0. \end{cases}$$

Let $\mathcal{H}$ denote either $\mathcal{H}^0$ or any of its sub-rectangles, where $\mathcal{H} = \prod_{i=1}^{p-1} [\underline{\mu}_i, \overline{\mu}_i]$ and $\mathcal{H} \subseteq \mathcal{H}^0$. Consequently, the subproblem of the EP defined over $\mathcal{H}$ can be expressed as follows:

$$\text{EP}(\mathcal{H}) \quad \begin{cases} \min \ \phi(x,\mu) = \sum_{i=1}^{p-1} \mu_i + \dfrac{\mathbf{c}_p^\top x + c_{0p}}{\mathbf{d}_p^\top x + d_{0p}} \\ \text{s.t. Eq (2.3)} \quad x \in \mathcal{X}, \ \mu \in \mathcal{H}. \end{cases}$$

Given the unique characteristics of the BB algorithm, we propose a linear fractional relaxation strategy for the subproblem EP($\mathcal{H}$) to facilitate the bounding operation.

2.2. *Linear fractional relaxation strategy*

In this section, each nonlinear constraint of the problem EP($\mathcal{H}$) is first reformulated into a more manageable form. Thereafter, EP($\mathcal{H}$) is relaxed into a linear fractional relaxation programming (LFRP) problem, which serves to provide a lower bound for the optimal value of EP($\mathcal{H}$).

For any $(x, \mu) \in \mathcal{X} \times \mathcal{H}$, it is observed that

$$\underline{\mu}_i \leq \frac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}} = \mu_i \leq \overline{\mu}_i, \ i \in I. \tag{2.5}$$

Besides, in the problem EP($\mathcal{H}$), Eq (2.3) together with $(x, \mu) \in \mathcal{X} \times \mathcal{H}$ indicates that

$$\underline{\mu}_i^0 \leq \underline{\mu}_i \leq \frac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}} \leq \overline{\mu}_i \leq \overline{\mu}_i^0, \ i \in I.$$

Throughout this paper, the following sets are defined:

$$\begin{aligned} I^1 &:= \{i : \underline{\mu}_i^0 > 0, i \in I\}, \quad I^2 := \{i : \overline{\mu}_i^0 < 0, i \in I\}, \\ I^3 &:= \{i : \overline{\mu}_i^0 + \underline{\mu}_i^0 \geq 0, i \in I \setminus (I^1 \cup I^2)\}, \\ I^4 &:= \{i : \overline{\mu}_i^0 + \underline{\mu}_i^0 < 0, i \in I \setminus (I^1 \cup I^2)\}. \end{aligned}$$

Next, for each $i \in I$, we define:

$$\xi_i := \begin{cases} 1, \ i \in I^1 \cup I^2, \\ \underline{\mu}_i^0 - 1, \ i \in I^3, \\ 1 + \overline{\mu}_i^0, \ i \in I^4. \end{cases}$$

Thus, for every $i \in I$, and for any $x \in \mathcal{X}_{\mathcal{H}^i} := \{x \in \mathcal{X} | \underline{\mu}_i \leq \frac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}} \leq \overline{\mu}_i\}$, we have that

$$\begin{cases} \overline{\mu}_i - \xi_i \geq \dfrac{\mathbf{c}_i^\top x + c_{0i} - \xi_i(\mathbf{d}_i^\top x + d_{0i})}{\mathbf{d}_i^\top x + d_{0i}} \geq \underline{\mu}_i - \xi_i \geq \underline{\mu}_i^0 - \xi_i \geq 1 > 0, \ i \in I^1 \cup I^3, \\ \underline{\mu}_i - \xi_i \leq \dfrac{\mathbf{c}_i^\top x + c_{0i} - \xi_i(\mathbf{d}_i^\top x + d_{0i})}{\mathbf{d}_i^\top x + d_{0i}} \leq \overline{\mu}_i - \xi_i \leq \overline{\mu}_i^0 - \xi_i \leq -1 < 0, \ i \in I^2 \cup I^4. \end{cases}$$

Given that $\min\limits_{x \in \mathcal{X}} \mathbf{d}_i^\top x + d_{0i} > 0$ for $i \in I$, it holds that

$$\begin{cases} \min\limits_{x \in \mathcal{X}_{\mathcal{H}^i}} [\mathbf{c}_i^\top x + c_{0i} - \xi_i(\mathbf{d}_i^\top x + d_{0i})] > 0, \ i \in I^1 \cup I^3, \\ \max\limits_{x \in \mathcal{X}_{\mathcal{H}^i}} [\mathbf{c}_i^\top x + c_{0i} - \xi_i(\mathbf{d}_i^\top x + d_{0i})] < 0, \ i \in I^2 \cup I^4. \end{cases}$$

Currently, solve the following problems for each $i \in I$:

$$l_i := \min_{x \in \mathcal{X}} \frac{\mathbf{c}_i^\top x + c_{0i} - \xi_i(\mathbf{d}_i^\top x + d_{0i})}{\mathbf{d}_p^\top x + d_{0p}},$$
$$u_i := \max_{x \in \mathcal{X}} \frac{\mathbf{c}_i^\top x + c_{0i} - \xi_i(\mathbf{d}_i^\top x + d_{0i})}{\mathbf{d}_p^\top x + d_{0p}}.$$

It satisfies that

$$\begin{cases} u_i(\mathbf{d}_p^\top x + d_{0p}) \geq \mathbf{c}_i^\top x + c_{0i} - \xi_i(\mathbf{d}_i^\top x + d_{0i}) \geq l_i(\mathbf{d}_p^\top x + d_{0p}) > 0, \ i \in I^1 \cup I^3, \\ l_i(\mathbf{d}_p^\top x + d_{0p}) \leq \mathbf{c}_i^\top x + c_{0i} - \xi_i(\mathbf{d}_i^\top x + d_{0i}) \leq u_i(\mathbf{d}_p^\top x + d_{0p}) < 0, \ i \in I^2 \cup I^4. \end{cases}$$

Thus, for any $x \in \mathcal{X}_{\mathcal{H}^i}$, we have

$$\begin{cases} \dfrac{u_i(\mathbf{d}_p^\top x + d_{0p})}{\mathbf{d}_i^\top x + d_{0i}} \geq \dfrac{\mathbf{c}_i^\top x + c_{0i} - \xi_i(\mathbf{d}_i^\top x + d_{0i})}{\mathbf{d}_i^\top x + d_{0i}} \geq \underline{\mu}_i - \xi_i > 0, \ i \in I^1 \cup I^3, \\ \overline{\mu}_i - \xi_i \geq \dfrac{\mathbf{c}_i^\top x + c_{0i} - \xi_i(\mathbf{d}_i^\top x + d_{0i})}{\mathbf{d}_i^\top x + d_{0i}} \geq \dfrac{l_i(\mathbf{d}_p^\top x + d_{0p})}{\mathbf{d}_i^\top x + d_{0i}} > 0, \ i \in I^1 \cup I^3, \\ \underline{\mu}_i - \xi_i \leq \dfrac{\mathbf{c}_i^\top x + c_{0i} - \xi_i(\mathbf{d}_i^\top x + d_{0i})}{\mathbf{d}_i^\top x + d_{0i}} \leq \dfrac{u_i(\mathbf{d}_p^\top x + d_{0p})}{\mathbf{d}_i^\top x + d_{0i}} < 0, \ i \in I^2 \cup I^4, \\ \dfrac{l_i(\mathbf{d}_p^\top x + d_{0p})}{\mathbf{d}_i^\top x + d_{0i}} \leq \dfrac{\mathbf{c}_i^\top x + c_{0i} - \xi_i(\mathbf{d}_i^\top x + d_{0i})}{\mathbf{d}_i^\top x + d_{0i}} \leq \overline{\mu}_i - \xi_i < 0, \ i \in I^2 \cup I^4. \end{cases} \tag{2.6}$$

Eq (2.6) indicates that any element x in $\mathcal{X}_{\mathcal{H}^i}$ satisfies the following condition:

$$\begin{cases} 0 < \dfrac{l_i}{\overline{\mu}_i - \xi_i} \leq \dfrac{\mathbf{d}_i^\top x + d_{0i}}{\mathbf{d}_p^\top x + d_{0p}} \leq \dfrac{u_i}{\underline{\mu}_i - \xi_i}, \ i \in I^1 \cup I^3, \\ 0 < \dfrac{u_i}{\underline{\mu}_i - \xi_i} \leq \dfrac{\mathbf{d}_i^\top x + d_{0i}}{\mathbf{d}_p^\top x + d_{0p}} \leq \dfrac{l_i}{\overline{\mu}_i - \xi_i}, \ i \in I^2 \cup I^4. \end{cases} \tag{2.7}$$

Additionally, we solve the following linear fractional programming problems:

$$0 < \underline{g}_i = \min_{x \in \mathcal{X}} \frac{\mathbf{d}_i^\top x + d_{0i}}{\mathbf{d}_p^\top x + d_{0p}}, \quad \overline{g}_i = \max_{x \in \mathcal{X}} \frac{\mathbf{d}_i^\top x + d_{0i}}{\mathbf{d}_p^\top x + d_{0p}}, \tag{2.8}$$

By combining Eq (2.7) with Eq (2.8), we define:

$$L_i = \begin{cases} \max\left\{\underline{g}_i, \dfrac{l_i}{\overline{\mu}_i - \xi_i}\right\}, \ i \in I^1 \cup I^3, \\ \max\left\{\underline{g}_i, \dfrac{u_i}{\underline{\mu}_i - \xi_i}\right\}, \ i \in I^2 \cup I^4, \end{cases} \quad U_i = \begin{cases} \min\left\{\overline{g}_i, \dfrac{u_i}{\underline{\mu}_i - \xi_i}\right\}, \ i \in I^1 \cup I^3, \\ \min\left\{\overline{g}_i, \dfrac{l_i}{\overline{\mu}_i - \xi_i}\right\}, \ i \in I^2 \cup I^4. \end{cases} \tag{2.9}$$

For each $i \in I$, Eqs (2.7) to (2.9) imply:

$$0 < L_i \le \frac{\mathbf{d}_i^\top x + d_{0i}}{\mathbf{d}_p^\top x + d_{0p}} \le U_i, \ x \in \mathcal{X}_{\mathcal{H}^i}.$$

Equivalently, for any $x \in \mathcal{X}_{\mathcal{H}^i}$, the following inequalities hold:

$$\begin{cases} \left(\dfrac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}} - \overline{\mu}_i\right)\left(\dfrac{\mathbf{d}_i^\top x + d_{0i}}{\mathbf{d}_p^\top x + d_{0p}} - L_i\right) \le 0, \ i \in I, \\ \left(\dfrac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}} - \underline{\mu}_i\right)\left(\dfrac{\mathbf{d}_i^\top x + d_{0i}}{\mathbf{d}_p^\top x + d_{0p}} - U_i\right) \le 0, \ i \in I. \end{cases} \tag{2.10}$$

Then, for each $i \in I$ and every element $x \in \mathcal{X}_{\mathcal{H}^i}$, by rearranging the terms in Eq (2.10) and transforming the resulting inequality, we obtain:

$$\begin{cases} \dfrac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}} \ge \dfrac{1}{L_i} \dfrac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_p^\top x + d_{0p}} - \dfrac{\overline{\mu}_i}{L_i} \dfrac{\mathbf{d}_i^\top x + d_{0i}}{\mathbf{d}_p^\top x + d_{0p}} + \overline{\mu}_i =: \theta^1_{\mathcal{H}^i}(x), \\ \dfrac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}} \ge \dfrac{1}{U_i} \dfrac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_p^\top x + d_{0p}} - \dfrac{\underline{\mu}_i}{U_i} \dfrac{\mathbf{d}_i^\top x + d_{0i}}{\mathbf{d}_p^\top x + d_{0p}} + \underline{\mu}_i =: \theta^2_{\mathcal{H}^i}(x). \end{cases} \tag{2.11}$$

From Eq (2.11), each non-convex constraint $\frac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}} = \mu_i$ is relaxed to:

$$\mu_i \ge \theta^1_{\mathcal{H}^i}(x), \ \mu_i \ge \theta^2_{\mathcal{H}^i}(x), \ i \in I. \tag{2.12}$$

When the equality in Eq (2.5) is broken, a crude relaxation of the non-convex constraint $\frac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}} = \mu_i$ is achieved, i.e.,

$$\mu_i \le \overline{\mu}_i, \ \frac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}} \ge \underline{\mu}_i, \ i \in I. \tag{2.13}$$

Given $\min_{x \in \mathcal{X}} \mathbf{d}_i^\top x + d_{0i} > 0$, the inequality $\frac{\mathbf{c}_i^\top x + c_{0i}}{\mathbf{d}_i^\top x + d_{0i}} \ge \underline{\mu}_i$ can be reformulated as:

$$\mathbf{c}_i^\top x + c_{0i} \ge \underline{\mu}_i(\mathbf{d}_i^\top x + d_{0i}), \ i \in I. \tag{2.14}$$

By combining Eqs (2.12), (2.13), and (2.14), EP($\mathcal{H}$) can be relaxed to:

$$\text{LFRP}(\mathcal{H}):\begin{cases}\min\ \underline{\phi}(x,\mu)=\sum_{i=1}^{p-1}\mu_i+\dfrac{\mathbf{c}_p^\top x+c_{0p}}{\mathbf{d}_p^\top x+d_{0p}}\\ \text{s.t. } \theta^1_{\mathcal{H}^i}(x)\le\mu_i,\ \theta^2_{\mathcal{H}^i}(x)\le\mu_i,\ i\in I,\\ \quad\ \mathbf{c}_i^\top x+c_{0i}\ge\underline{\mu}_i(\mathbf{d}_i^\top x+d_{0i}),\ i\in I\\ \quad\ \mu_i\le\overline{\mu}_i,\ i\in I,\\ \quad\ x\in\mathcal{X}.\end{cases}$$

Remark 2.2. *It should be noted that from $\mu_i\ge\theta^1_{\mathcal{H}^i}(x)$ and $\mu_i\le\overline{\mu}_i$, one may deduce $\mathbf{c}_i^\top x+c_{0i}\le\overline{\mu}_i(\mathbf{d}_i^\top x+d_{0i})$. Additionally, from $\mu_i\ge\theta^2_{\mathcal{H}^i}(x)$ and $\mathbf{c}_i^\top x+c_{0i}\ge\underline{\mu}_i(\mathbf{d}_i^\top x+d_{0i})$, it can be inferred that $\mu_i\ge\underline{\mu}_i$.*

Remark 2.3. *For each $i\in I$, the optimal solution (x^*,μ^*) for* LFRP*($\mathcal{H}$) must satisfy $\mu_i^*=\max\{\theta^1_{\mathcal{H}^i}(x^*),\theta^2_{\mathcal{H}^i}(x^*)\}$. If not, a contradiction arises readily through a proof by contradiction.*

From Eqs (2.12) and (2.13), one can deduce that the feasible region of EP($\mathcal{H}$) exists as a subset within that of LFRP($\mathcal{H}$). Consequently, it is impossible for the optimal value of LFRP($\mathcal{H}$) to surpass that of EP($\mathcal{H}$). Thus, solving LFRP($\mathcal{H}$) yields an effective lower bound for the optimal value of EP($\mathcal{H}$).

Theorem 2.2. *If the problems* LFRP*($\mathcal{H}$) and* EP*($\mathcal{H}$) share the optimum (x^*,μ^*), then for each $i\in I$, at least one of the following four equations must hold:*

$$\frac{\mathbf{c}_i^\top x^*+c_{0i}}{\mathbf{d}_i^\top x^*+d_{0i}}=\overline{\mu}_i,\ \frac{\mathbf{d}_i^\top x^*+d_{0i}}{\mathbf{d}_p^\top x^*+d_{0p}}=L_i,\ \frac{\mathbf{c}_i^\top x^*+c_{0i}}{\mathbf{d}_i^\top x^*+d_{0i}}=\underline{\mu}_i,\ \frac{\mathbf{d}_i^\top x^*+d_{0i}}{\mathbf{d}_p^\top x^*+d_{0p}}=U_i. \tag{2.15}$$

Proof. Suppose there exists an index $i\in I$ such that $\frac{\mathbf{c}_i^\top x^*+c_{0i}}{\mathbf{d}_i^\top x^*+d_{0i}}\notin\{\underline{\mu}_i,\overline{\mu}_i\}$ and $\frac{\mathbf{d}_i^\top x^*+d_{0i}}{\mathbf{d}_p^\top x^*+d_{0p}}\notin\{L_i,U_i\}$. By the optimality condition of EP($\mathcal{H}$), it follows that $\frac{\mathbf{c}_i^\top x^*+c_{0i}}{\mathbf{d}_i^\top x^*+d_{0i}}=\mu_i^*$, which implies $\mu_i^*\in(\underline{\mu}_i,\overline{\mu}_i)$ and $\frac{\mathbf{d}_i^\top x^*+d_{0i}}{\mathbf{d}_p^\top x^*+d_{0p}}\in(L_i,U_i)$. Let $\tilde{\mu}_i\triangleq\max\left\{\vartheta^1_{\mathcal{H}}(x^*),\vartheta^2_{\mathcal{H}}(x^*)\right\}$. Through application of the definitions of $\vartheta^1_{\mathcal{H}}(x^*)$ and $\vartheta^2_{\mathcal{H}}(x^*)$ in LFRP($\mathcal{H}$), it can be derived that $\mu_i^*>\tilde{\mu}_i$. Furthermore, from the definitions and given conditions, $\tilde{\mu}_i\ge\vartheta^2_{\mathcal{H}}(x^*)\ge\underline{\mu}_i$ is obtained. Thus, $\tilde{\mu}_i<\mu_i^*<\overline{\mu}_i$, and $(x^*,\tilde{\mu})$ is a feasible solution of LFRP($\mathcal{H}$). However, $\underline{\phi}(x^*,\tilde{\mu})<\underline{\phi}(x^*,\mu^*)$ is observed, which contradicts the optimality of (x^*,μ^*) for LFRP($\mathcal{H}$).

Consequently, it is concluded that at least one of the four equations in Eq (2.15) holds for all $i\in I$. This completes the proof. □

Remark 2.4. *Theorem 2.2 states that if the optimum (x^*,μ^*) of* LFRP($\mathcal{H}$) *satisfies none of the four equations in* Eq (2.15) *for at least one $i\in I$, then (x^*,μ^*) is not optimal for* EP($\mathcal{H}$).

The following theorem gives the approximate error between the optimal values of LFRP($\mathcal{H}$) and EP($\mathcal{H}$), and shows that the approximation error in the worst case is related to the side length of the rectangle $\mathcal{H}$.

Theorem 2.3. *Let (x^*,μ^*) be the optimum to problem* LFRP*($\mathcal{H}$) with $\mathcal{H}=\prod_{i=1}^{p-1}[\underline{\mu}_i,\overline{\mu}_i]$, and let $\tau\in\arg\max_{i\in I}\left\{\left(\overline{\mu}_i-\underline{\mu}_i\right)\left(1-\frac{L_i}{U_i}\right)\right\}$, where it holds that*

$$\lim_{\overline{\mu}_\tau-\underline{\mu}_\tau\to0}(V_{\text{EP}}(\mathcal{H})-V_{\text{LFRP}}(\mathcal{H}))=0.$$

Proof. From Eqs (2.7), (2.10), and (2.11), it follows that

$$\begin{aligned}
0 &\le V_{\mathrm{EP}}(\mathcal{H}) - V_{\mathrm{LFRP}}(\mathcal{H}) \\
&\le \phi(x^*, \tilde{\mu}^*) - \underline{\phi}(x^*, \mu^*) \\
&\le \sum_{i=1}^{p-1} \left(\frac{\mathbf{c}_i^\top x^* + c_{0i}}{\mathbf{d}_i^\top x^* + d_{0\tau}} - \mu_i^* \right) \\
&= \sum_{i=1}^{p-1} \left(\frac{\mathbf{c}_i^\top x^* + c_{0i}}{\mathbf{d}_i^\top x^* + d_{0i}} - \max\{\theta^1_{\mathcal{H}^i}(x^*), \theta^2_{\mathcal{H}^i}(x^*)\} \right) \\
&= \sum_{i=1}^{p-1} \min \left\{ \frac{\mathbf{c}_i^\top x^* + c_{0i}}{\mathbf{d}_i^\top x^* + d_{0i}} - \theta^1_{\mathcal{H}_i}(x^*), \frac{\mathbf{c}_i^\top x^* + c_{0i}}{\mathbf{d}_i^\top x^* + d_{0i}} - \theta^2_{\mathcal{H}_i}(x^*) \right\} \\
&= \sum_{i=1}^{p-1} \min \left\{ \frac{1}{L_i} \left(\overline{\mu}_i - \frac{\mathbf{c}_i^\top x^* + c_{0i}}{\mathbf{d}_i^\top x^* + d_{0i}} \right) \left(\frac{\mathbf{d}_i^\top x^* + d_{0i}}{\mathbf{d}_p^\top x^* + d_{0p}} - L_i \right), \frac{1}{U_i} \left(\frac{\mathbf{c}_i^\top x^* + c_{0i}}{\mathbf{d}_i^\top x^* + d_{0i}} - \underline{\mu}_i \right) \left(U_i - \frac{\mathbf{d}_i^\top x^* + d_{0i}}{\mathbf{d}_p^\top x^* + d_{0p}} \right) \right\} \\
&\le \sum_{i=1}^{p-1} \min \left\{ \frac{1}{L_i} \left(\overline{\mu}_i - \underline{\mu}_i \right) (U_i - L_i), \frac{1}{U_i} \left(\overline{\mu}_i - \underline{\mu}_i \right) (U_i - L_i) \right\} \\
&= \sum_{i=1}^{p-1} \frac{1}{U_i} \left(\overline{\mu}_i - \underline{\mu}_i \right) (U_i - L_i) \\
&= (p-1) \left(\overline{\mu}_\tau - \underline{\mu}_\tau \right) \left(1 - \frac{L_\tau}{U_\tau} \right).
\end{aligned} \tag{2.16}$$

Therefore, as $\overline{\mu}_\tau - \underline{\mu}_\tau \to 0$, Eq (2.16) implies that $V_{\mathrm{EP}}(\mathcal{H}) - V_{\mathrm{LFRP}}(\mathcal{H}) \to 0$. □

As inferred from Eq (2.16), the rectangular regions $\mathcal{H} = \prod_{i=1}^{p-1} [\underline{\mu}_i, \overline{\mu}_i]$ associated with the variables μ_τ are directly subjected to branching and thinning operations. As a result, the optimal values of EP($\mathcal{H}$) and LFRP($\mathcal{H}$) converge asymptotically, approaching each other in the limit. This behavior ensures the coherence between bounding and branching mechanisms in the BB algorithm.

Theorem 2.4. *Given $\epsilon > 0$, let $\tau \in \arg\max_{i \in I} \left\{ \left(\overline{\mu}_i - \underline{\mu}_i \right) \left(1 - \frac{L_i}{U_i} \right) \right\}$. If*

$$\Upsilon := (p-1) \left(\overline{\mu}_\tau - \underline{\mu}_\tau \right) \left(1 - \frac{L_\tau}{U_\tau} \right) \le \epsilon, \tag{2.17}$$

the optimal solution (x^*, μ^*) of the problem LFRP($\mathcal{H}$) *yields a global ϵ-optimal solution $(x^*, \tilde{\mu}^*)$ for* EP($\mathcal{H}$)*, where $\tilde{\mu}^* = \left(\frac{\mathbf{c}_1^\top x^* + c_{01}}{\mathbf{d}_1^\top x^* + d_{01}}, \cdots, \frac{\mathbf{c}_{p-1}^\top x^* + c_{0p-1}}{\mathbf{d}_{p-1}^\top x^* + d_{0p-1}} \right)^\top$.*

Proof. From Remark 2.2, it can be inferred that $(x^*, \tilde{\mu}^*)$ is feasible for EP($\mathcal{H}$), giving rise to the following chain of inequalities:

$$0 \le \phi(x^*, \tilde{\mu}^*) - V_{\mathrm{EP}}(\mathcal{H}) \le \phi(x^*, \tilde{\mu}^*) - V_{\mathrm{LFRP}}(\mathcal{H}) = \phi(x^*, \tilde{\mu}^*) - \underline{\phi}(x^*, \mu^*) \le \Upsilon \le \epsilon.$$

This implies that $\phi(x^*, \tilde{\mu}^*) \le V_{\mathrm{EP}}(\mathcal{H}) + \epsilon$, thereby validating the conclusion. □

2.3. *Solving problem* LFRP($\mathcal{H}$)

For a given rectangle $\mathcal{H} \subseteq \mathcal{H}^0$, the problem LFRP($\mathcal{H}$) is solvable, though it cannot be directly resolved using standard convex optimization solvers. Notably, LFRP($\mathcal{H}$) exhibits implicit convexity due to its linear fractional components sharing a common denominator, $\mathbf{d}_p^\top x + d_{0p}$. By implementing the CCT defined as $t = \frac{1}{\mathbf{d}_p^\top x + d_{0p}}$ and $y = tx$, the linear fractional relaxation problem is reformulated into a linear program:

$$\mathrm{LP}(\mathcal{H}) \quad \begin{cases} \min \omega(y,t,w) = \sum_{i=1}^{p-1} \mu_i + \mathbf{c}_p^\top y + c_{0p}t \\ \text{s.t.} \ \ \frac{1}{L_i}(\mathbf{c}_i^\top y + c_{0i}t) - \frac{\overline{\mu}_i}{L_i}(\mathbf{d}_i^\top y + d_{0i}t) + \overline{\mu}_i \le \mu_i, \ i \in I, \\ \quad \frac{1}{U_i}(\mathbf{c}_i^\top y + c_{0i}t) - \frac{\underline{\mu}_i}{U_i}(\mathbf{d}_i^\top y + d_{0i}t) + \underline{\mu}_i \le \mu_i, \ i \in I, \\ \quad \mu_i \le \overline{\mu}_i, \ \mathbf{c}_i^\top y + c_{0i}t \ge \underline{\mu}_i(\mathbf{d}_i^\top y + d_{0i}t), \ i \in I, \\ \quad \mathbf{d}_p^\top y + d_{0p}t = 1, \ Ay - bt \le 0, \ t > 0. \end{cases}$$

The problem LP($\mathcal{H}$) is well-defined, as demonstrated in [9, 25]. Thus, the conclusion given in the following Remark 2.5 follows directly.

Remark 2.5. *Define the set* $\mathcal{W} = \{(y,t) \mid \mathbf{d}_p^\top y + d_{0p}t = 1, \ Ay - bt \le 0, \ t > 0\}$. *For any* $(y,t) \in \mathcal{W}$, *we have* $t > 0$ *and* $y/t \in \mathcal{X}$. *Conversely, for any* $x \in \mathcal{X}$, *setting* $t = \frac{1}{\mathbf{d}_p^\top x + d_{0p}}$ *and* $y = tx$ *yields* $(y,t) \in \mathcal{W}$ *and* $\omega(y,t,\mu) = \underline{\phi}(x,\mu)$. *Thus, the optimum* $\left(\frac{\hat{y}}{\hat{t}}, \hat{\mu}\right)$ *of* LFRP($\mathcal{H}$) *is derived from the optimum* $(\hat{y}, \hat{t}, \hat{\mu})$ *of* LP($\mathcal{H}$).

2.4. *Branching rule*

The branching rule represents a central component of the BB algorithm. Consider a sub-rectangle $\mathcal{H} = \prod_{i=1}^{p-1}[\underline{\mu}_i, \overline{\mu}_i] \subseteq \mathcal{H}^0$ chosen at a specific iteration of the BB algorithm. Let (x^*, μ^*) be the optimal solution for the LFRP problem over $\mathcal{H}$, where its optimal value represents the minimum lower bound. When $\left(\overline{\mu}_\tau - \underline{\mu}_\tau\right)\left(1 - \frac{L_\tau}{U_\tau}\right) \le \frac{\epsilon}{p-1}$, Theorem 2.4 indicates that $(x^*, \tilde{\mu}^*)$ with $\tilde{\mu}^* = \left(\frac{\mathbf{c}_1^\top x^* + c_{01}}{\mathbf{d}_1^\top x^* + d_{01}}, \cdots, \frac{\mathbf{c}_{p-1}^\top x^* + c_{0p-1}}{\mathbf{d}_{p-1}^\top x^* + d_{0p-1}}\right)^\top$ is a global ϵ-optimum of the problem EP($\mathcal{H}$). It will also be proven later that $(x^*, \tilde{\mu}^*)$ is simultaneously a global ϵ-optimum of EP. Hence, based on Theorems 2.3 and 2.4, τ is as follows:

$$\tau \in \arg\max \left\{ \left(\overline{\mu}_i - \underline{\mu}_i\right)\left(1 - \frac{L_i}{U_i}\right) : i \in I \right\}. \tag{2.18}$$

Next, consider the rectangle $\mathcal{H} \subseteq \mathcal{H}^0$ chosen in a specific iteration. The branching rule for this rectangle is outlined below:

(1) Let $\tau \in \arg\max \left\{ \left(\overline{\mu}_i - \underline{\mu}_i\right)\left(1 - \frac{L_i}{U_i}\right) : i \in I \right\}$, $\nu_\tau = \frac{\underline{\mu}_\tau + \overline{\mu}_\tau}{2}$;

(2) By utilizing ν_τ, the interval $[\underline{\mu}_\tau, \overline{\mu}_\tau]$ is divided into two subintervals: $[\underline{\mu}_\tau, \nu_\tau]$ and $[\nu_\tau, \overline{\mu}_\tau]$. This division refines $\mathcal{H}$ into sub-rectangles:

$$\mathcal{H}^1 = \prod_{i=1}^{\tau-1}[\underline{\mu}_i, \overline{\mu}_i] \times [\underline{\mu}_\tau, \nu_\tau] \times \prod_{i=\tau+1}^{p-1}[\underline{\mu}_i, \overline{\mu}_i] \text{ and } \mathcal{H}^2 = \prod_{i=1}^{\tau-1}[\underline{\mu}_i, \overline{\mu}_i] \times [\nu_\tau, \overline{\mu}_\tau] \times \prod_{i=\tau+1}^{p-1}[\underline{\mu}_i, \overline{\mu}_i].$$

It holds that $\mathcal{H}^1 \cap \mathcal{H}^2 = \{\mu \in \mathbb{R}^{p-1} \mid \mu_\tau = \nu_\tau\}$ and $\mathcal{H}^1 \cup \mathcal{H}^2 = \mathcal{H}$.

2.5. Region elimination

This subsection introduces a technique for eliminating nonoptimal regions within a sub-rectangle $\mathcal{H}$, termed region elimination. This method reduces the number of explored child nodes and compresses the search domain, thereby speeding up the convergence of the BB algorithm. Let $\mathcal{H} = \prod_{i=1}^{p-1}[\underline{\mu}_i, \overline{\mu}_i] \subseteq \mathcal{H}^0$ be defined without loss of generality, while $\mathcal{UB}$ is designated as the current best-known objective function value for EP.

For any feasible solution (x, μ) worth considering to problem EP($\mathcal{H}$), it must necessarily satisfy:

$$\phi(x, \mu) = \sum_{i=1}^{p-1} \mu_i + \frac{\mathbf{c}_p^\top x + c_{0p}}{\mathbf{d}_p^\top x + d_{0p}} \leq \mathcal{UB}.$$

Thus, for each $\iota \in I$, we define:

$$\eta = \hat{\mu} + \sum_{i=1}^{p-1} \underline{\mu}_i \text{ and } \delta_\iota = \mathcal{UB} - \eta + \underline{\mu}_\iota$$

with $\hat{\mu} = \min\limits_{x \in \mathcal{X}} \frac{\mathbf{c}_p^\top x + c_{0p}}{\mathbf{d}_p^\top x + d_{0p}}$. If the global optimum $(\tilde{x}, \tilde{\mu})$ of EP is derivable from $\mathcal{H}$, a necessary condition is that

$$\phi(\tilde{x}, \tilde{\mu}) \leq \phi(x, \mu) \leq \mathcal{UB} \; for \; some \; (x, \mu) \in \mathcal{X} \times \mathcal{H}. \tag{2.19}$$

This condition serves as foundational basis for the subsequent rectangular reduction theorem.

Theorem 2.5. *Let $\mathcal{H} = \prod_{i=1}^{p-1}[\underline{\mu}_i, \overline{\mu}_i] \subseteq \mathcal{H}^0$ be a hyperrectangle. For problem* EP*:*

i). If $\delta_\iota < \underline{\mu}_\iota$ for some $\iota \in I$, then EP *admits no global optimum over $\mathcal{H}$.*

ii). If $\underline{\mu}_\iota \leq \delta_\iota < \overline{\mu}_\iota$ for some $\iota \in I$, then no global optimum exists in the sub-rectangle

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

Proof. i): Suppose $\delta_\iota < \underline{\mu}_\iota$ for $\iota \in I$. For any $(x, \mu) \in \mathcal{X} \times \mathcal{H}$, the upper bound $\mathcal{UB}$ satisfies:

$$\phi(\tilde{x}, \tilde{\mu}) \leq \mathcal{UB} = \delta_\iota + \eta - \underline{\mu}_\iota < \eta = \hat{\mu} + \sum_{i=1}^{p-1} \underline{\mu}_i \leq \phi(x, \mu),$$

which contradicts the optimality condition in Eq (2.19). Hence, no global optimum exists over $\mathcal{H}$.

ii): If $\underline{\mu}_\iota \leq \delta_\iota < \overline{\mu}_\iota$ and $\mu \in \overline{\mathcal{H}}_\iota$, then $\delta_\iota < \mu_\iota \leq \overline{\mu}_\iota$. For $(x, \mu) \in \mathcal{X} \times \overline{\mathcal{H}}_\iota$,

$$\phi(\tilde{x}, \tilde{\mu}) \leq \mathcal{UB} = \delta_\iota + \eta - \underline{\mu}_\iota < w_\iota + \eta - \underline{\mu}_\iota \leq w_\iota + \hat{\mu} + \sum_{\substack{i=1 \\ i \neq \iota}}^{p-1} \mu_i \leq \phi(x, \mu).$$

This implies $\phi(\tilde{x}, \tilde{\mu}) < \phi(x, \mu)$ for all $w \in \overline{\mathcal{H}}_\iota$, so $(\tilde{x}, \tilde{\mu})$ cannot lie in $\mathcal{X} \times \overline{\mathcal{H}}_\iota$. Thus, $\overline{\mathcal{H}}_\iota$ contains no optimal components. Hence, the proof of the theorem is finished. □

2.6. Linear fractional relaxation based branch-and-bound reduction algorithm

To identify the globally optimum in the lifted problem EP, a linear fractional relaxation-based BB reduction (LFRBBR) algorithm is developed. The algorithm introduces novel relaxation subproblems, rectangular branching operations, and domain reduction techniques. The detailed implementation steps are given as follows.

Algorithm 1 Linear Fractional Relaxation-based BB Reduction Algorithm

Require: An instance of Problem SLR and a tolerance $\epsilon > 0$.

Construct the initial rectangle $\mathcal{H}^0 = \prod_{i=1}^{p-1}[\underline{\mu}_i^0, \overline{\mu}_i^0]$ using Eq (2.4).
Solve the relaxation problem LFRP($\mathcal{H}^0$) to obtain its optimal value $\mathcal{LB}(\mathcal{H}^0)$ and optimal solution (x^0, μ^0).
Set $\mathcal{LB}^0 = \mathcal{LB}(\mathcal{H}^0)$, $\mathcal{UB}^0 = \varphi(x^0)$, $x^v = x^0$, $\Xi = \{[\mathcal{H}^0, \mathcal{LB}(\mathcal{H}^0)]\}$ and $k = 0$.
while $\mathcal{UB}^k - \mathcal{LB}^k > \epsilon$ **do**
    Use bisection (Section 2.4) to split $\mathcal{H}^k = \prod_{i=1}^{p-1}[\underline{\mu}_i^k, \overline{\mu}_i^k]$ into $\mathcal{H}^{k1}$ and $\mathcal{H}^{k2}$.
    Remove $[\mathcal{H}^k, \mathcal{LB}(\mathcal{H}^k)]$ from Ξ.
    for $\varsigma = 1, 2$ **do**
        Apply the region elimination method proposed in Section 2.5 to the sub-rectangle $\mathcal{H}^{k\varsigma}$.
        if $\mathcal{H}^{k\varsigma}$ does not satisfy Conclusion i) of Theorem 2.5 **then**
            Put the compressed rectangle (still denoted as $\mathcal{H}^{k\varsigma}$) into the set Q.
        end if
    end for
    if $|Q| \neq 0$ **then**
        for $\mathcal{H} \in Q$ **do**
            Solve the relaxation problem LFRP($\mathcal{H}$) using the method in Section 2.3.
            if LFRP($\mathcal{H}$) is solvable **then**
                Output the optimal value $\mathcal{LB}(\mathcal{H})$ and the optimal solution $(\hat{x}, \hat{\mu})$ of LFRP($\mathcal{H}$).
                if $\mathcal{UB}^k - \mathcal{LB}(\mathcal{H}) > \epsilon$ **then** Add $[\mathcal{H}, \mathcal{LB}(\mathcal{H})]$ to Ξ.
                    if $\varphi(\hat{x}) < \mathcal{UB}^k$ **then** Set $\mathcal{UB}^k = \varphi(\hat{x})$ and $x^v = \hat{x}$.
                    end if
                end if
            end if
        end for
    end if
    if $\Xi \neq \emptyset$ **then**
        Set $\mathcal{LB}^k = \min\{\mathcal{LB}(\mathcal{H}) | [\cdot, \mathcal{LB}(\mathcal{H})] \in \Xi\}$.
        Select $\{[\mathcal{H}^k, \cdot]\} \in \Xi$ where $\mathcal{LB}(\mathcal{H}^k) = \mathcal{LB}^k$. Set $k = k + 1$.
    end if
end while
return x^v, $\mathcal{UB}^k$.

Remark 2.6. *In the algorithm, the optimal solution* $(\hat{x}, \hat{\mu})$ *of* LFRP($\mathcal{H}$) *is generally infeasible for* EP. *Define* $\check{\mu} = (\check{\mu}_1, \cdots, \check{\mu}_{p-1})^\top$ *with* $\check{\mu}_i = \frac{\mathbf{d}_i^\top \hat{x} + d_{0i}}{\mathbf{c}_i^\top \hat{x} + c_{0i}}$, *which renders* $(\hat{x}, \check{\mu})$ *feasible for* EP *and ensures*

$\phi(\hat{x}, \check{\mu}) = \varphi(\hat{x})$. *In such cases,* $\varphi(\hat{x})$ *may replace* $\phi(\hat{x}, \check{\mu})$ *to update the upper bound.*

In the proposed algorithm, let k be the iteration index. Each step replaces a subproblem with at most two new ones, which have better optimal values to update upper and lower bounds. As k increases, the bound gap shrinks. The LFRBBR algorithm finds a global ϵ-optimum ($\epsilon > 0$) for EP: a solution $(x^v, \mu^v) \in \mathcal{F}$ satisfying $\phi(x^v, \mu^v) \leq V(\text{EP}) + \epsilon$, where $V(\text{EP})$ is an EP optimal value and $\mathcal{F}$ is its feasible region.

Theorem 2.6. *Given a tolerance* $\epsilon > 0$*, at the kth iteration of the* LFRBBR *algorithm, when reaching Step 1, if the subproblem* $\{[\mathcal{H}^k, \mathcal{LB}(\mathcal{H}^k)]\}$ *satisfies* $\overline{\mu}_\tau^k - \underline{\mu}_\tau^k \leq \frac{\epsilon}{(p-1)\left(1-\frac{L_\tau}{U_\tau}\right)}$ *for* $\tau \in \arg\max\left\{\left(\overline{\mu}_i^k - \underline{\mu}_i^k\right)\left(1 - \frac{L_i}{U_i}\right) : i \in I\right\}$*, then the algorithm terminates and yields a global* ϵ*-optimum for the* SLR *problem.*

Proof. Let (x^*, μ^*) be an optimum of LFRP($\mathcal{H}^k$). From the proof of Theorem 2.3, it holds that

$$\phi(x^*, \tilde{\mu}^*) - \underline{\phi}(x^*, \mu^*) \leq (p-1)(\overline{\mu}_\tau^k - \underline{\mu}_\tau^k)\left(1 - \frac{L_\tau}{U_\tau}\right), \tag{2.20}$$

where $\tilde{\mu}^* = (\tilde{\mu}_1^*, \tilde{\mu}_2^*, \cdots, \tilde{\mu}_{p-1}^*)^\top$ and $\tilde{\mu}_i^* = \frac{\mathbf{c}_i^\top x^* + c_{0i}}{\mathbf{d}_i^\top x^* + d_{0i}}$.

Since $\mathcal{LB}^k = \mathcal{LB}(\mathcal{H}^k)$ is the minimum lower bound at the current iteration, we get

$$\underline{\phi}(x^*, \mu^*) = \mathcal{LB}^k \leq V(\text{EP}) \leq \mathcal{UB}^k = \phi(x^v, w^v) \leq \phi(x^*, \tilde{\mu}^*), \tag{2.21}$$

where $\mu^v = (\mu_1^v, \mu_2^v, \cdots, \mu_{p-1}^v)^\top$ and $\mu_i^v = \frac{\mathbf{c}_i^\top x^v + c_{0i}}{\mathbf{d}_i^\top x^v + d_{0i}}$ for $i \in I$. When $\overline{\mu}_\tau^k - \underline{\mu}_\tau^k \leq \frac{\epsilon}{(p-1)\left(1-\frac{L_\tau}{U_\tau}\right)}$, from Eqs (2.20) and (2.21), we can infer that

$$\phi(x^v, \mu^v) - V(\text{EP}) \leq \mathcal{UB}^k - \mathcal{LB}^k \leq \phi(x^*, \tilde{\mu}^*) - \underline{\phi}(x^*, \mu^*) \leq \epsilon, \tag{2.22}$$

which shows that (x^v, w^v) is a global ϵ-optimum of EP. □

Remark 2.7. *It can be verified from the definition of* μ^v *presented in the proof of Theorem 2.6 that*

$$\varphi(x^v) = \phi(x^v, \mu^v). \tag{2.23}$$

Furthermore, the conclusion of Theorem 2.1 establishes that

$$V(\text{EP}) = V(\text{SLR}). \tag{2.24}$$

When Eq *(2.22) is combined with Eqs (2.23) and (2.24), it can be deduced that*

$$\phi(x^v, \mu^v) - V(\text{EP}) = \varphi(x^v) - V(\text{SLR}) \leq \epsilon.$$

As demonstrated in Theorem 2.6, the optimal value of EP for any selected sub-rectangle converges to the value of the relaxed problem LFRP($\mathcal{H}$) as the limit is approached. This convergence ensures the uniform application of bounding and branching steps, which confirms the theoretical convergence of the BB algorithm. Subsequently, we will analyze the complexity of LFRBBR by leveraging Theorem 2.6 and Remark 2.7.

Theorem 2.7. *For a tolerance $\epsilon > 0$, the maximum number of iterations required for the* LFRBBR *algorithm to find a global ϵ-optimal solution for the* SLR *problem is* $\left\lceil \prod_{i=1}^{p-1} \frac{(p-1)\left(1-\frac{L_i}{U_i}\right)(\overline{\mu}_i^0-\underline{\mu}_i^0)}{\epsilon} \right\rceil$.

Proof. For each $i \in I$, define the initial interval length as $\Delta_i^0 = \overline{\mu}_i^0 - \underline{\mu}_i^0$. In the worst-case scenario (where the algorithm requires the most iterations to achieve ϵ-optimality), assume every subinterval length Δ_i in all sub-rectangles satisfies:

$$\Delta_i \leq \frac{\epsilon}{(p-1)\left(1-\frac{L_i}{U_i}\right)},\ i \in I. \tag{2.25}$$

Let N_i denote the number of subintervals in the i-th dimension. The subinterval length in dimension i is $\Delta_i = \Delta_i^0 / N_i$. Substituting into Eq (2.25) gives $\frac{\Delta_i^0}{N_i} \leq \frac{\epsilon}{(p-1)\left(1-\frac{L_i}{U_i}\right)}$. Rearranging (since all terms are positive) yields a lower bound on N_i:

$$N_i \geq \frac{(p-1)\left(1-\frac{L_i}{U_i}\right)\Delta_i^0}{\epsilon}.$$

Given that the dimensional partitions are independent of each other, the total number of all sub-rectangles satisfies:

$$N_{\text{total}} = \prod_{i=1}^{p-1} N_i \geq \prod_{i=1}^{p-1} \frac{(p-1)\left(1-\frac{L_i}{U_i}\right)\Delta_i^0}{\epsilon}.$$

Let k denote the iteration index. Since $k+1 = N_{\text{total}}$ (as k starts from 0), rearranging gives:

$$k \geq \prod_{i=1}^{p-1} \frac{(p-1)\left(1-\frac{L_i}{U_i}\right)\Delta_i^0}{\epsilon} - 1.$$

By the pigeonhole principle, if $k < \left\lceil \prod_{i=1}^{p-1} \frac{(p-1)\left(1-\frac{L_i}{U_i}\right)\Delta_i^0}{\epsilon} \right\rceil$, then $N_{\text{total}} < \prod_{i=1}^{p-1} \frac{(p-1)\left(1-\frac{L_i}{U_i}\right)\Delta_i^0}{\epsilon}$. This implies some $N_i < \frac{(p-1)\left(1-\frac{L_i}{U_i}\right)\Delta_i^0}{\epsilon}$, yielding a subinterval length exceeding $\frac{\epsilon}{(p-1)\left(1-\frac{L_i}{U_i}\right)}$, violating Eq (2.25). When $k = \left\lceil \prod_{i=1}^{p-1} \frac{(p-1)\left(1-\frac{L_i}{U_i}\right)\Delta_i^0}{\epsilon} \right\rceil$, every subregion $\mathcal{H} = \prod_{i=1}^{p-1}[\underline{\mu}_i, \overline{\mu}_i]$ satisfies Eq (2.25), and

$$\mathcal{UB}^k - \mathcal{LB}(\mathcal{H}) \leq (p-1)(\overline{\mu}_\tau - \underline{\mu}_\tau)\left(1-\frac{L_\tau}{U_\tau}\right) \leq \epsilon,$$

where $\tau \in \arg\max\left\{(\overline{\mu}_i - \underline{\mu}_i)\left(1-\frac{L_i}{U_i}\right) : i \in I\right\}$. This empties Ξ via the pruning rule, halting iterations.

Thus, $k = \left\lceil \prod_{i=1}^{p-1} \frac{(p-1)\left(1-\frac{L_i}{U_i}\right)\Delta_i^0}{\epsilon} \right\rceil$, giving the maximum iterations as stated. □

Remark 2.8. *Theorem 2.7 bounds the computational time of* LFRBBR *when it finds a global ϵ-optimum for* SLR *as*

$$2\mathcal{T}\left\lceil \prod_{i=1}^{p-1} \frac{(p-1)\left(1-\frac{L_i}{U_i}\right)(\overline{\mu}_i^0 - \underline{\mu}_i^0)}{\epsilon} \right\rceil s,$$

where $\mathcal{T}$ is the upper bound of the time to solve an LP($\mathcal{H}$) *problem (defined in Section 2.3).*

Remark 2.9. *Theorem 2.7 confirms finite termination of the* LFRBBR *algorithm via a well-defined iteration upper bound.*

2.7. Differences among various BB algorithms

Table 2. Differences in the core characteristics of the BB algorithm among various references.

References	Equivalent Problem Characteristics	Relaxation Type	Var. Dimension	Properties of constraints
[20]	With p bilinear equality constraints and the objective being the sum of p bilinear functions	LR: Via Lagrangian weak duality	$p+m+1$	$[(p+1)(n+1)+m]$ linear constraints
[21]	With $2p$ convex constraints and the objective being the sum of $2p$ bilinear functions	Convex relaxation: Using convex envelopes of bilinear terms	$n+4p$	$(m+9p+n)$ linear constraints and p convex constraints
[7]	With pn bilinear constraints and a linear objective	LR: Relaxing all bilinear constraints via variable bounds	$p(n+1)$	$p(m+3n+2)$ linear constraints
[22]	With p^2 bilinear constraints and a linear objective	LR: Relaxing all bilinear constraints via variable bounds	$p(n+1)$	$p(m+2n+3)$ linear constraints
[23]	With p bilinear inequality constraints and the objective being the sum of p bilinear functions	LR: Relax objective via convex envelope of bilinear terms, and relax constraints using bounds of introduced variables	$n+3p$	$(m+8p+n)$ linear constraints
[24]	Similar to [20]	LR: Relax objective and constraints using variable signs and variable bounds	$n+p$	$(m+4p+n)$ linear constraints
[28]	Similar to [20]	LR: Relax objective via convex envelope and constraints via both concave and convex envelopes of bilinear function	$n+3p$	$(m+11p+n)$ linear constraints
[29]	Only objective contains sum of p simple linear fractions	LR: Two-stage relaxation of each simple linear fraction	$n+2p$	$(m+8p+n)$ linear constraints
[25]	With $(p-1)$ bilinear equality constraints and a linear objective	LR: Use concave envelope for bilinear constraints	$n+p$	$(m+n+3p-1)$ linear constraints
[26]	With $(p-1)$ equality constraints $v_i = 1/z_i$ and objective being sum of $(p-1)$ bilinear terms and a linear function	LR: Use convex envelope for objective and concave envelope for $1/z_i$ with tangent cuts	$n+2p-1$	$(m+n+5p-3)$ linear constraints
[1]	With linear fractional objective and $(p-1)$ inequality constraints with sum of two linear fractions	SOCR: Use concave envelope of $(\mu_i)^2$ to relax each constraint, then apply CCT to transform problem	$n+2p-1$	$(m+n+3p-1)$ linear constraints and $(p-1)$ second-order cone constraints
[30]	After process similar to [25], form has $(p-1)$ D.C. constraints and a linear objective	SOCR: Use concave envelope of $(s_i)^2$ to relax each D.C. constraint and introduce new variables for SOCR	$n+5p-4$	$(m+n+6p-4)$ linear constraints and $(p-1)$ second-order cone constraints
[31]	Similar to [30]	QCR: Use concave envelope of $(w_i)^2$ to relax all D.C. constraints	$n+3p-2$	$(m+n+4p-2)$ linear constraints and $(p-1)$ quadratic convex constraints
[32]	Objective consists of sum of $(p-1)$ simple linear fractions and a linear function	LR: Use concave envelope to linearize objective on constructed trapezoid and provide tightening inequalities	$n+2p-1$	$(m+n+4p-2)$ or $(m+n+4p-1)$ linear constraints
[33]	With $(p-1)$ inequality constraints $1/t_i \le v_i$ and objective being sum of $(p-1)$ bilinear terms and a linear function	LR: Use convex envelope for objective then use tangents of $1/t_i$ at endpoints to relax $1/t_i$ and add valid inequalities	$n+3p-2$	$(m+n+7p-5)$ or $(m+n+7p-5+\lvert\bar{I}\rvert)$ linear constraints, $\bar{I} \subseteq I$
Ours	With linear fractional objective and $(p-1)$ linear fractional equality constraints	LFR: Use bounds of each fraction, its numerator and denominator to relax fractional equality constraints	$n+p-1$	$(m+n+2p-2)$ linear constraints $(2p-2)$ linear fractional inequality constraints

To gain a clearer grasp of the discrepancies between our BB algorithm and other extant algorithms, Table 2 lays out the key features of these methods, including the core attributes of equivalent problems, the characteristics of relaxation problems, the variable dimensions of relaxation problems, as well as the properties of their constraints.

From Table 2, it can be observed that most existing algorithms primarily rely on bilinear/quadratic constraints or reciprocal terms for problem formulation, and most involve generally higher variable dimensions, with relaxation methods such as LR, SOCR, and QCR utilized. Furthermore, other characteristics of the BB algorithms in the relevant literature listed in the table can be found in Section 1. Our LFR-based BB algorithm first constructs an equivalent problem with a linear fractional objective and $(p-1)$ linear fractional equality constraints; subsequently, after LFR relaxation, it employs CCT to transform the problem into a linear program with $n+p$ variables and $m+n+4p-2$ linear constraints.

In summary, the information in the table indicates that the main innovations of most existing studies differ in terms of equivalent problems and relaxation problems.

3. Numerical experiments

In order to assess the efficacy and dependability of the LFRBBR, we conducted computational experiments. These experiments juxtaposed the LFRBBR with cutting-edge BB algorithms from recent academic contributions [1, 25, 29, 33] and the commercially available solver, BARON [34]. Furthermore, with the goal of demonstrating the potential of the proposed algorithm in real-world scenarios, we tackled the cost optimization problem in hospital management to validate its practical utility. The algorithms were coded in Matlab (2023a) and tested in numerical experiments. All computations were performed on a desktop running Windows 7, equipped with an Intel(R) Core(TM) i5-8500 3.00 GHz CPU and 8 GB of RAM. The Cplex solver embedded in Matlab was used to solve all linear programming and SOCP problems. For these experiments, a uniform tolerance of 10^{-4} was applied across all six algorithms.

3.1. Comparison of algorithms

To assess the performance of these methods, we proceed with randomized experiments on the following Problems 1 and 2.

Problem 1

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

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

Problem 2

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

where all e_{ij}, f_i, b_k, and a_{kj} are randomly generated in [0,1]; and c_{ij} and d_i are randomly generated in [-1,1].

Table 3. Computational comparisons among LFRBBR and BARON on Problem 1.

(p, m, n)	LFRBBR			BARON		
	Iter	CPU	Opt.val	Iter	CPU	Opt.val
(2,20,100)	10.8	0.054	1.3576	15.2	0.365	1.3576
(2,60,300)	8.9	0.077	2.0316	22.2	6.549	2.0316
(2,100,500)	9.9	0.266	3.4444	20.3	27.530	3.4444
(2,140,700)	8.2	0.571	3.2151	49.1	39.951	3.2151
(2,200,1000)	14.8	2.420	2.1690	61.8	109.339	2.1690
(2,400,2000)	5.9	3.818	2.9435	10.9	492.200	2.9435
(2,600,3000)	13.2	20.172	1.5038	–	–	–
(3,20,100)	37.5	0.271	3.4245	12.9	0.406	3.4245
(3,60,300)	51.7	0.618	3.4093	78.4	13.953	3.4093
(3,100,500)	31.1	1.282	4.3145	69.8	44.166	4.3145
(3,140,700)	38.2	2.553	3.3055	79.9	78.857	3.3055
(3,200,1000)	35.5	5.158	6.8248	178.3	197.488	6.8248
(3,400,2000)	27.4	14.253	5.5535	–	–	–
(3,500,2500)	72.8	39.495	6.2716	–	–	–
(4,20,100)	122.2	0.458	4.9181	13.4	0.549	4.9181
(4,60,300)	153.1	1.835	4.9374	36.8	16.155	4.9374
(4,100,500)	120.9	4.399	5.6523	25.1	46.920	5.6523
(4,140,700)	151.8	10.711	9.9620	123.3	125.978	9.9620
(4,200,1000)	207.1	30.432	9.8768	259.0	220.862	9.8768
(5,20,100)	340.9	1.535	5.9230	154.5	2.469	5.9230
(5,60,300)	359.4	4.183	7.2545	608.4	31.272	7.2545
(5,100,500)	236.1	9.287	6.2716	47.8	43.271	6.2716
(5,140,700)	366.5	26.613	10.4443	248.3	162.253	10.4443
(5,200,1000)	278.8	46.827	10.6524	177.9	322.479	10.6524

For each parameter set (p, m, n), we generated 10 random instances of identical size, which were subsequently solved using the six algorithms under study. The mean numerical outcomes are detailed in Tables 3–5. The headers for these tables are defined as follows:

- CPU: the average Central Processing Unit time necessitated by an algorithm to resolve ten instances,

- Iter: the average number of iterations required to resolve ten instances,
- Opt.val: the average optimal value attained across ten resolved instances.
- −: A certain algorithm cannot solve these ten instances within 3600 seconds.

The numerical comparison results between the LFRBBR algorithm and other algorithms (including the commercial solver BARON) are presented in Tables 3 to 5, which verify the feasibility and effectiveness of our proposed algorithm in handling SLR instances.

Table 4. Computational comparisons among LFRBBR and the algorithms in References [29, 33] on Problem 1.

(p, m, n)	LFRBBR			Reference [29]			Reference [33]		
	Iter	CPU	Opt.val	Iter	CPU	Opt.val	Iter	CPU	Opt.val
(2,20,100)	3.5	0.0051	1.6465	21.1	0.0367	1.6465	6.2	0.0121	1.6465
(2,60,300)	5.4	0.0266	2.1387	23.6	0.0994	2.1387	6.4	0.0408	2.1387
(2,100,500)	6.5	0.1087	2.8209	43.1	0.3663	2.8200	10.4	0.2046	2.8209
(2,140,700)	2.4	0.0778	2.7451	13.8	0.2306	2.7451	4.2	0.1562	2.7451
(2,200,1000)	2.7	0.1866	4.3968	12.1	0.4546	4.3968	3.0	0.2472	4.3968
(2,400,2000)	3.0	0.7826	2.1727	4.5	0.9018	2.1727	1.8	0.7410	2.1727
(2,500,2500)	2.5	2.3608	7.0467	7.6	2.3901	7.0467	2.4	2.4312	7.0467
(2,600,3000)	2.7	3.7112	4.2048	5.6	4.1068	4.2048	4.0	5.1432	4.2048
(3,20,100)	15.2	0.0687	2.6878	66.8	0.1378	2.6878	21.0	0.0703	2.6878
(3,60,300)	17.6	0.0883	2.4837	100.2	0.4787	2.4837	34.8	0.2429	2.4837
(3,100,500)	32.2	0.5628	3.1112	262.9	4.2353	3.1112	83.0	1.5695	3.1112
(3,140,700)	13.4	0.4437	3.3607	44.2	0.7591	3.3607	34.8	1.2310	3.3607
(3,200,1000)	11.6	0.7818	7.2415	38.4	1.5902	7.2415	12.4	1.2195	7.2415
(3,400,2000)	18.1	4.8664	4.9612	49.2	10.0245	4.9612	24.0	8.1677	4.9612
(4,20,100)	389.3	3.6587	4.8283	749.5	4.2889	4.8283	810.5	2.5897	4.8283
(4,60,300)	494.8	4.5736	6.7220	333.4	4.2631	6.7220	314.9	8.9588	6.6940
(4,100,500)	223.4	5.1272	5.7219	497.0	5.7221	5.7220	461.2	11.2709	5.7221
(4,140,700)	128.9	5.4505	6.2284	310.6	10.6189	6.2284	210.4	9.0479	6.2284
(4,200,1000)	138.4	10.5127	3.4521	297.2	14.4592	3.4521	133	10.1089	3.4521
(5,20,100)	610.2	1.6857	5.5604	1063.5	2.7348	5.5604	621.9	2.6962	5.5604
(5,60,300)	280.0	2.7107	6.5604	575.1	3.6904	6.5604	418.0	3.8034	6.5604
(5,100,500)	241.2	4.8058	8.8081	1333.4	16.0834	8.8083	420.8	9.7153	8.7702
(5,140,700)	204.6	8.3264	5.9375	3433.4	71.5348	5.9375	739.0	30.5808	5.9375
(5,200,1000)	250.0	19.5086	6.5334	591.9	28.7758	6.5334	232.4	19.2090	6.5334

Table 5. Computational comparisons among LFRBBR and the algorithms in References [1, 25] on Problem 2.

(p, m, n)	LFRBBR			Reference [25]			Reference [1]		
	Iter	CPU	Opt.val	Iter	CPU	Opt.val	Iter	CPU	Opt.val
(2,5,25)	19.9	0.0126	-8.7486	20.8	0.0267	-8.7486	5.7	0.7373	-8.7486
(2,10,50)	10.8	0.0102	-0.5697	10.8	0.0167	-0.5697	5.3	0.6812	-0.5697
(2,20,100)	11.4	0.0125	-1.2524	12.0	0.0238	-1.2524	5.9	0.8831	-1.2524
(2,60,300)	3.9	0.0178	-0.2832	4.4	0.0344	-0.2832	5.8	1.1442	-0.2832
(2,100,500)	4.3	0.0621	-0.4972	2.8	0.0754	-0.4972	4.7	1.3745	-0.4972
(2,140,700)	5.8	0.1584	-4.8840	3.9	0.1891	-4.8840	4.6	2.0455	-4.8840
(2,200,1000)	3.8	0.2189	-0.4521	2.2	0.2760	-0.4521	3.5	2.9405	-0.4521
(2,400,2000)	2.4	0.5915	-0.4672	1.5	1.0119	-0.4672	2.7	10.4942	-0.4672
(3,5,25)	36.0	0.0257	-1.5554	41.3	0.0514	-1.5554	16.0	1.8743	-1.5554
(3,10,50)	63.7	0.0515	-2.4821	69.8	0.1004	-2.4843	14.3	1.6889	-2.4843
(3,20,100)	27.5	0.0343	-2.9380	36.6	0.0696	-2.9380	14.6	2.0868	-2.9380
(3,60,300)	21.2	0.0908	-2.5622	23.0	0.1412	-2.5622	13.5	3.0267	-2.5622
(3,100,500)	66.0	3.0768	-2.4879	105.8	5.2090	-2.4879	14.8	20.5486	-2.4879
(3,140,700)	16.4	1.7714	1.4472	23.9	3.5092	1.4472	11.3	49.0087	1.4472
(3,200,1000)	24.6	3.4955	-1.8196	26.4	4.7259	-1.8196	8.8	81.4375	-1.8196
(4,5,25)	290.1	1.0059	-2.1218	764.9	3.2590	-2.1218	32.3	14.8638	-2.1218
(4,10,50)	395.1	1.4701	-33.9716	1017.2	4.3842	-33.9716	30.2	14.1333	-33.9716
(4,20,100)	246.6	1.4054	-11.4552	685.2	4.6018	-11.4548	32.3	17.4182	-11.4548
(4,60,300)	150.3	2.9411	-6.3919	314.2	6.8332	-6.3919	28.6	25.1296	-6.3919
(4,100,500)	180.5	6.3247	-30.4056	503.5	11.0786	-30.4056	18.6	31.9987	-30.4056
(4,140,700)	122.6	8.9266	-3.9189	132.4	14.2129	-3.9190	19.4	81.6202	-3.9189
(5,5,25)	249.7	0.6726	-3.0549	477.7	2.3312	-3.0549	52.9	21.4983	-3.0549
(5,10,50)	950.5	3.7329	-8.0687	2761.2	15.7465	-8.0687	61.7	28.2353	-8.0687
(5,20,100)	200.4	0.7315	-3.3460	676.2	5.1062	-3.3460	46.6	23.0552	-3.3460
(5,60,300)	308.7	5.3293	3.2308	954.0	20.8327	3.2308	71.6	70.9919	3.2308
(5,100,500)	276.3	15.5954	-4.9926	907.5	55.6446	-4.9926	53.0	99.3233	-4.9926

Tables 3 and 4 present the numerical comparison results for solving Problem 1, with Table 3 clearly illustrating the performance differences between LFRBBR and BARON. As indicated in Table 3, in terms of the number of iterations, LFRBBR requires fewer iterations for most large-scale problems (e.g., $n \geq 500$). For instance, in the case of $(p, m, n) = (2, 600, 3000)$, LFRBBR converges in only 13.2 iterations, while BARON times out. BARON exhibits an advantage in iteration count solely for small-scale problems with $n = 100$ and $p \geq 3$; for example, in the scenario of $(p, m, n) = (3, 20, 100)$, BARON iterates 12.9 times compared to 37.5 iterations for LFRBBR. In terms of CPU time, LFRBBR consumes significantly less time than BARON across all solvable cases. Taking the $(2, 100, 500)$ test case as an example, LFRBBR takes merely 0.266 seconds, whereas BARON requires 27.530 seconds. Furthermore, the growth of CPU time with problem size is much more gradual for LFRBBR: when n increases from 100 to 1000, the CPU time of LFRBBR increases by approximately 67-fold, while that of BARON surges by over 487-fold. In addition, the optimal values obtained by both algorithms are completely identical across all test cases, indicating that they converge to solutions of comparable quality. For complex problems with $p \geq 3$ and $n \geq 2000$, BARON repeatedly exceeded the 1-hour time limit, whereas LFRBBR successfully solved them within the time frame. This demonstrates the stronger scalability and robustness of LFRBBR, particularly when handling high-dimensional ($n \geq 1000$) or high-p ($p = 4, 5$) problems, owing to its superior iterative strategy and efficient resource utilization. Although BARON shows advantages in iteration efficiency for small-scale ($n = 100$) and high-p scenarios, its time complexity increases sharply with problem size, rendering it incapable of solving large-scale problems. It is thus evident that LFRBBR is more suitable for large-scale optimization scenarios in SLRP problems, while BARON, constrained by its computational complexity, is only applicable to lightweight tasks.

Beyond the comparison with BARON, the numerical results in Table 4 further exhibit the performance differences between LFRBBR and the algorithms reported in [29] and [33] in solving Problem P1. For most problem instances, especially those with larger scales and higher p values, LFRBBR consistently demands fewer iterations. In terms of CPU time, LFRBBR maintains high efficiency in most cases, particularly when n is large, with only occasional instances where alternative methods perform slightly better. Regarding solution quality, the three algorithms yield nearly identical optimal values across most test cases, with only minor deviations observed in specific instances. From the perspective of parameter trends, as n increases from 100 to 3000 or p rises from 2 to 5, LFRBBR demonstrates more stable performance in reducing the number of iterations and saving CPU time compared to the other two methods. For example, in the scenario of $(p, m, n) = (5, 100, 500)$, LFRBBR requires 241.2 iterations and 4.8058 seconds of CPU time, while the algorithm in [29] necessitates 1333.4 iterations and 16.0834 seconds, and the algorithm in [33] utilizes 420.8 iterations and 9.7153 seconds. This further confirms the stronger scalability of LFRBBR for large-scale problems.

Turning to the solution of Problem 2, the numerical comparisons in Table 5 further validate the comprehensive performance of LFRBBR. The optimal values obtained by the three algorithms exhibit a high degree of consistency, with only negligible minor deviations in very few test cases, fully confirming their excellent ability to approximate the optimal solution of the problem. In terms of the number of iterations, the iteration count of LFRBBR is generally smaller than that of the algorithm in [25], with only a slight excess in individual test cases such as $(p, m, n) = (2, 100, 500)$. However, the algorithm in [1] demonstrates significantly fewer iterations than the previous two across all test cases, exhibiting a more streamlined iterative path. In terms of CPU time consumption, the running time of LFRBBR is generally lower than that of the algorithm in [25], and this advantage becomes increasingly prominent as the problem scale increases. Compared with the algorithm in [1], the time efficiency advantage of LFRBBR is particularly remarkable in large-scale problems. For instance, in the test case $(p, m, n) = (3, 200, 1000)$, the CPU time of LFRBBR is only 3.4955, which is much lower than the 81.4375 required by the algorithm in [1]. It is thus evident that while ensuring excellent solution accuracy, LFRBBR also demonstrates outstanding performance in terms of iteration efficiency and computational time consumption. Its advantages are more significant, especially when handling large-scale problems, highlighting its superior performance in complex optimization scenarios.

3.2. Applications in hospital management

Consider the charge optimization problem in hospital management [35], which originates from the distinct reimbursement mechanisms for Medicare/Medicaid patients and non-Medicare/Medicaid patients. A uniform across-the-board price increase strategy for all departments may fail to achieve optimal revenue.

Specifically, hospital services are divided into several departments, each administering a number of procedures. The hospital sets a specific charge for each procedure, and the revenue return depends on whether a patient is a Medicare/Medicaid beneficiary. When a patient is not a Medicare/Medicaid beneficiary, the hospital can receive all charges for that patient, with bad debts being temporarily ignored. For Medicare/Medicaid patients, the reimbursement for inpatient services is a fixed amount, which is independent of the hospital's charges and thus ignored in the discussion. Meanwhile, the reimbursement for outpatient services is determined through a cost accounting procedure. First, the fixed Medicare/Medicaid cost for each department is determined in accordance with government guidelines. Subsequently, the Medicare/Medicaid outpatient reimbursement for each department is calculated as this fixed cost multiplied by the proportion of Medicare/Medicaid outpatient charges to total charges for that particular department.

To construct an optimization model, let d denote the number of departments in the hospital. For each $i = 1, 2, \ldots, d$, let p_i be the number of procedures in department i. For each procedure j in department i, let c_{ij}, m_{ij}, and o_{ij} represent the total charges, Medicare/Medicaid outpatient charges, and non-Medicare/Medicaid charges during the past year, respectively. It is assumed that $c_{ij} > 0$, while $m_{ij} \geq 0$ and $o_{ij} \geq 0$ hold for all i and j. Additionally, let C_i denote the fixed Medicare/Medicaid outpatient cost for department i. Then, the hospital's revenue in the past year can be expressed as

$$\sum_{i=1}^{d} \left[\sum_{j=1}^{p_i} o_{ij} + C_i \frac{\sum_{j=1}^{p_i} m_{ij}}{\sum_{j=1}^{p_i} c_{ij}} \right].$$

Now, introduce r_{ij} as the fraction of increase in the charge for procedure j in department i. Based on the distribution of patients and services from the previous year, the expected revenue function for the next year is given by

$$F(\mathbf{r}) = \sum_{i=1}^{d} \left[\sum_{j=1}^{p_i} o_{ij}(1 + r_{ij}) + C_i \frac{\sum_{j=1}^{p_i} m_{ij}(1 + r_{ij})}{\sum_{j=1}^{p_i} c_{ij}(1 + r_{ij})} \right],$$

where $\mathbf{r}$ is a vector containing all components r_{ij}. Owing to the nonlinear dependence of $F(\mathbf{r})$ on $\mathbf{r}$, a uniform proportional price increase (i.e., all r_{ij} taking a constant value) may not optimize $F(\mathbf{r})$. In fact, it may even be necessary to reduce charges for some procedures to optimize the Medicare/Medicaid charge ratio for a specific department.

If the hospital aims to achieve an overall charge increase of $q \times 100\%$ and sets upper and lower bounds l_{ij} and u_{ij} for each r_{ij} (with $l_{ij} \leq q \leq u_{ij}$ satisfied), then one needs to solve the following nonlinear programming problem:

$$\begin{aligned} &\max\ F(\mathbf{r}) \\ &\text{s.t.} \quad \sum_{i=1}^{d} \sum_{j=1}^{p_i} c_{ij} r_{ij} = q \sum_{i=1}^{d} \sum_{j=1}^{p_i} c_{ij} \\ &\qquad\ l_{ij} \leq r_{ij} \leq u_{ij},\ i = 1, 2, \cdots, d,\ j = 1, 2, \cdots, p_i, \end{aligned} \tag{3.1}$$

where $l_{ij} < u_{ij}$ holds for all i and j.

Since "$\max F(\mathbf{r}) = -\min -F(\mathbf{r})$", Problem (3.1) has the same global optimal solution as the following problem:

$$\begin{aligned} \min \quad & \sum_{i=1}^{d}\left[-\sum_{j=1}^{p_i} o_{ij}(1+r_{ij}) - C_i \frac{\sum_{j=1}^{p_i} m_{ij}(1+r_{ij})}{\sum_{j=1}^{p_i} c_{ij}(1+r_{ij})}\right] \\ \text{s.t.} \quad & \sum_{i=1}^{d}\sum_{j=1}^{p_i} c_{ij} r_{ij} = q \sum_{i=1}^{d}\sum_{j=1}^{p_i} c_{ij} \\ & l_{ij} \le r_{ij} \le u_{ij},\ i = 1,2,\cdots,d,\ j = 1,2,\cdots,p_i. \end{aligned} \tag{3.2}$$

This problem exhibits the typical characteristics of an SLR problem, namely, that the objective function takes the form of a sum of linear fractional terms, and thus it can serve as a practical instance of the SLR problem.

Consistent with the example given in [35], suppose a hospital has four departments, each containing five procedures, i.e., $d = 4$, $p_i = 5$ for $i = 1,2,3,4$; and for all i and j, let $q = 0.10$, $u_{ij} = 0.15$, and $l_{ij} = -0.05$. The values of c_{ij}, m_{ij}, o_{ij}, and C_i are presented in Table 6.

Table 6. Simulated hospital charges and costs.

Departments	Procedures	o_{ij}	m_{ij}	c_{ij}	C_i
1	1	16.35	48.54	64.89	185.68
1	2	74.12	10.11	84.23	185.68
1	3	39.26	2.34	41.60	185.68
1	4	48.20	5.69	53.89	185.68
1	5	12.63	7.20	19.83	185.68
2	1	13.22	48.45	61.66	118.90
2	2	12.67	20.94	33.61	118.90
2	3	22.71	64.95	87.66	118.90
2	4	5.44	22.88	28.32	118.90
2	5	7.24	17.49	24.73	118.90
3	1	12.66	13.60	26.26	134.21
3	2	6.84	6.97	13.81	134.21
3	3	3.72	22.39	26.11	134.21
3	4	6.66	55.18	61.84	134.21
3	5	15.89	63.44	79.33	134.21
4	1	33.03	14.04	47.07	166.56
4	2	0.18	7.07	7.25	166.56
4	3	5.23	1.33	6.56	166.56
4	4	7.60	41.53	49.13	166.56
4	5	67.54	23.74	91.28	166.56

Subsequently, we utilize the proposed LFRBBR algorithm to solve the above problem (3.2). The algorithm completes 40 iterations, ultimately yielding an optimal value of −780.3268 and an optimal solution of $\mathbf{r}^* = \begin{bmatrix} 0.15 & 0.15 & 0.15 & 0.15 & 0.15 \\ 0.0854 & 0.15 & 0.15 & -0.05 & 0.15 \\ 0.0607 & 0.15 & -0.05 & -0.05 & -0.05 \\ 0.15 & 0.15 & 0.15 & 0.15 & 0.15 \end{bmatrix}$. This result is in close proximity to that from the commercial optimization software BARON, further validating the practicality of our algorithm.

In summary, LFRBBR demonstrates significant advantages in terms of the number of iterations, CPU time efficiency, and scalability, particularly excelling in large-scale, high-dimensional, or high-p problems. Meanwhile, it can ensure solution quality comparable to or even consistent with other algorithms, making it a highly applicable algorithm in complex optimization scenarios. Finally, the successful application of our algorithm to solving the cost optimization problem in hospital management further demonstrates its practical applicability.

4. Conclusions

In this study, a new approach was proposed to solve the SLR problem by transforming it into EP and then developing a linear fractional relaxation strategy within the BB framework. The key innovations encompass a linear fractional relaxation sub-problem that leverages structural properties for efficient linear programming reformulation, an adaptive branching rule in the $\mathbb{R}^{p-1}$ space to minimize computational complexity, and a region elimination technique to bolster convergence. The resultant linear fractional relaxation-based algorithm, LFRBBR, offers theoretical assurances of convergence and complexity. Numerous numerical experiments revealed that, compared to the BB algorithms in recent literature [1, 25, 29, 33] and the commercial solver BARON, LFRBBR exhibited superior performance in solving SLR problems where p is significantly smaller than n. Notably, LFRBBR demonstrated remarkable efficiency in high-dimensional scenarios by branching in the reduced-dimensional output space of the intermediate variables and integrating adaptive acceleration techniques, thereby circumventing the curse of dimensionality inherent in traditional decision space branching. Moreover, we utilized the cost optimization problem in hospital management to demonstrate the practical applicability of our algorithm. Future research will extend the LFRBBR algorithm to SLR problems with quadratic functions and linear fractional multi-product optimization problems, and will explore its integration with machine learning-driven heuristic algorithms to further improve scalability and robustness.

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 receives support from the Youth Talent Cultivation Program of North Minzu University [grant number 2024QNPY16], the National Natural Science Foundation of China under Grant [grant number 12301401], the Construction Project of First-Class Subjects in Ningxia Higher Education [grant number NXYLXK2017B09], 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, Y. Qiao, Y. Sun, A nonlinear relaxation-strategy-based algorithm for solving sum-of-linear-ratios problems, *AIMS Math.*, **9** (2024), 25396–25412. http://dx.doi.org/10.3934/math.20241240
2. S. Schaible, Fractional programming, In: *Handbook of global optimization*, Boston: Springer, 1995. http://dx.doi.org/10.1007/978-1-4615-2025-2_10
3. H. Konno, H. Watanabe, Bond portfolio optimization problems and their applications to index tracking: A partial optimization approach, *J. Oper. Res. Soc. Jpn.*, **39** (1996), 295–306. http://dx.doi.org/10.15807/jorsj.39.295
4. I. Stancu-Minasian, A ninth bibliography of fractional programming, *Optimization*, **68** (2019), 2125–2169. http://dx.doi.org/10.1080/02331934.2019.1632250
5. B. Sawik, Downside risk approach for multi-objective portfolio optimization, In: *Operations Research Proceedings 2011*, Berlin: Springer, 2012. http://dx.doi.org/10.1007/978-3-642-29210-1_31
6. 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
7. 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
8. T. Matsui, NP-hardness of linear multiplicative programming and related problems, *J. Glob. Optim.*, **9** (1996), 113–119. http://dx.doi.org/10.1007/BF00121658
9. A. Charnes, W. Cooper, Programming with linear fractional functionals, *Nav. Res. Log.*, **9** (1962), 181–186. http://dx.doi.org/10.1002/nav.3800090303
10. 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
11. H. Konno, Y. Yajima, T. Matsui, Parametric simplex algorithms for solving a special class of nonconvex minimization problems, *J. Glob. Optim.*, **1** (1991), 65–81. http://dx.doi.org/10.1007/BF00120666

12. 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. Glob. Optim.*, **77** (2020), 301–318. http://dx.doi.org/10.1007/s10898-019-00870-2

13. Y. Nesterov, A. Nemirovskii, An interior-point method for generalized linear-fractional programming, *Math. Program.*, **69** (1995), 177–204. http://dx.doi.org/10.1007/BF01585557

14. H. Konno, N. Abe, Minimization of the sum of three linear fractional functions, *J. Glob. Optim.*, **15** (1999), 419–432. http://dx.doi.org/10.1023/A:1008376731013

15. H. Benson, On the global optimization of sums of linear fractional functions over a convex set, *J. Optim. Theory Appl.*, **121** (2004), 19–39. http://dx.doi.org/10.1023/B:JOTA.0000026129.07165.5a

16. 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

17. J. Falk, S. Palocsay, Image space analysis of generalized fractional programs, *J. Glob. Optim.*, **4** (1994), 63–88. http://dx.doi.org/10.1007/BF01096535

18. N. Phuong, H. Tuy, A unified monotonic approach to generalized linear fractional programming, *J. Glob. Optim.*, **26** (2003), 229–259. http://dx.doi.org/10.1023/A:1023274721632

19. 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

20. 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

21. 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

22. J. Carlsson, J. Shi, A linear relaxation algorithm for solving the sum-of-linear-ratios problem with lower dimension, *Oper. Res. Lett.*, **41** (2013), 381–389. http://dx.doi.org/10.1016/j.orl.2013.04.005

23. 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

24. 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

25. 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

26. B. Zhang, Y. Gao, An output-space based branch-and-bound algorithm for sum-of-linear-ratios problem, *Asia Pac. J. Oper. Res.*, **40** (2023), 2250010. http://dx.doi.org/10.1142/S0217595922500105

27. 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

28. 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

29. 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
30. P. Shen, Y. Wang, D. Wu, A spatial branch and bound algorithm for solving the sum of linear ratios optimization problem, *Numer. Algor.*, **93** (2023), 1373–1400. http://dx.doi.org/10.1007/s11075-022-01471-z
31. B. Huang, P. Shen, An efficient global optimization algorithm for the sum of linear ratios problems based on a novel adjustable branching rule, *Comput. Optim. Appl.*, **91** (2025), 1339–1371. http://dx.doi.org/10.1007/s10589-025-00679-8
32. H. Luo, Y. Xu, H. Wu, G. Wang, A new branch-and-cut algorithm for linear sum-of-ratios problem based on slo method and lo relaxation, *Comput. Optim. Appl.*, **90** (2025), 257–301. http://dx.doi.org/10.1007/s10589-024-00622-3
33. Y. Deng, P. Shen, An adaptive branch-and-bound reduction algorithm for minimizing sum of linear ratios programs, *Comput. Appl. Math.*, **44** (2025), 107. http://dx.doi.org/10.1007/s40314-024-03063-0
34. A. Khajavirad, N. Sahinidis, A hybrid LP/NLP paradigm for global optimization relaxations, *Math. Prog. Comp.*, **10** (2018), 383–421. http://dx.doi.org/10.1007/s12532-018-0138-5
35. F. H. Mathis, L. J. Mathis, A nonlinear programming algorithm for hospital management, *SIAM Rev.*, **37** (1995), 230–234. https://doi.org/10.1137/1037046
36. P. Shen, Y. Deng, Y. Wang, A one-dimensional branching rule based branch-and-bound algorithm for minimax linear fractional programming, *J. Comput. Appl. Math.*, **448** (2024), 115900. http://dx.doi.org/10.1016/j.cam.2024.115900

AIMS Press