Journal of Industrial and Management Optimization

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

JIMO, 22(1): 581–611.
DOI:10.3934/jimo.2026022
Received: 05 August 2025
Revised: 13 October 2025
Accepted: 05 November 2025
Published: 16 December 2025

Research article

A hybrid relaxation method for solving generalized linear fractional programs

Xia Jing[1,2], **Yuelin Gao**[1,*] **and Xiaoli Huang**[3]

[1] School of Mathematics and Information Science, North Minzu University, Yinchuan 750021, China
[2] College of Mathematics and Computer Science, Yan'an University, Yan'an 716000, China
[3] School of Mathematics and Statistics, Shaanxi Xueqian Normal University, Xian 710061, China

* **Correspondence:** Email: gaoyuelin@263.net; Tel: +18291774873.

Abstract: Performing the efficient global optimization algorithm for generalized linear fractional programs (GLFPs) is a very desirable goal in the field of optimization. As the problem's size increases, this goal is becoming increasingly difficult to achieve, although there have been some advances in recent years. In this paper, we present an efficient outcome space branch-and-bound algorithm for globally solving large-scale GLFPs. First, we convert the GLFP into its equivalent problem (EP) by introducing new variables. Second, a hybrid relaxation strategy (a convex envelope and a second-order cone) is used to derive a series of new linear relaxation problems (LRPs) that approximate the EP's optimal value. Meanwhile, a new region reduction method is presented, where the branching operation is performed in an outcome space. A novel branch-and-bound algorithm is then provided to solve GLFPs by computing a lower bound from the LRPs. Subsequently, the algorithm's convergence and worst-case iteration count are also reported. We conclude with numerical experiments illustrating the proposed algorithm's efficacy, especially when solving large-scale GLFPs.

Keywords: generalized linear fractional programs; hybrid relaxation; branch-and-bound; computational complexity
Mathematics Subject Classification: 90C26, 90C32

1. Introduction

Consider the following generalized linear fractional programming (GLFP) problem:

$$\text{GLFP}: \begin{cases} \min \ f(x) = \sum_{i=1}^{p} \delta_i \prod_{j=1}^{T_i} \left(\frac{c_{ij}^{\top} x + e_{ij}}{d_{ij}^{\top} x + f_{ij}} \right)^{\alpha_{ij}} \\ s.t. \ x \in \chi := \{x \in \mathbb{R}^n | Ax \le b\}, \end{cases}$$

where p and T_i are positive integers, $\delta_i, \alpha_{ij}, e_{ij}, f_{ij} \in \mathbb{R}$, $c_{ij}, d_{ij} \in \mathbb{R}^n$, $i \in I := \{1, \ldots, p\}$, $j \in J_i := \{1, \ldots, T_i\}$. $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, χ is a nonempty and bounded set, and both $c_{ij}^\top x + e_{ij}$ and $d_{ij}^\top x + f_{ij}$ ($i \in I$, $j \in J_i$) are positive linear functions defined on χ.

The GLFP problem, a significant category in mathematical programming, has captured the attention of numerous researchers due to its applications in scientific computing, economics, and engineering fields, including scheduling problems in transportation [1], project management [2], portfolio optimization [3–6], network flows [7,8], energy management [9], environmental science [10], and so on. It is well-known that the GLFP is a nonconvex optimization problem, characterized by many local optimal solutions that are not globally optimal, resulting in significant computational and theoretical challenges in addressing such a problem. As a result, the exploration of algorithms for tackling the GLFP challenge has become a hot topic for researchers.

Until now, many researchers have developed some deterministic algorithms for solving special cases of the GLFP problem. For example, when $p = 1$, $\alpha_{ij} = 1$, $d_{ij}^\top x + f_{ij} = 1$, $i \in I$, and $j \in J_i$, the GLFP problem becomes a linear multiplicative programming problem, for which several feasible algorithms exist, such as the monotonic optimization algorithm [11], the parametric simplex method [12], and the range division and linearization algorithm [13]. When $p \geq 2$, $T_i \geq 2$, $\alpha_{ij} = 1$, $d_{ij}^\top x + f_{ij} = 1$, $i \in I$, and $j \in J_i$, the GLFP problem is reduced to a generalized linear multiplicative programming problem, with the solution methods encompassing outer approximation [14], the cutting plane method [15,16], and the branch-and-bound algorithm [17,18]. When $p = 1$, $d_{ij}^\top x + f_{ij} = 1$, $i \in I$, $j \in J_i$, the GLFP problem is reduced to a multiplicative programming problem, which can be solved by algorithms such as the level set algorithm [19,20] and the branch-and-bound algorithm [21–23], as well as hybrid strategies combining outer approximation and branch-and-bound [24,25]. Furthermore, when $p = 1$ without additional constraints, the problem becomes a linear fractional-multiplicative program, solvable via methods such as branch-and-bound [26], among others. When $p \geq 2$, $T_i = 1$, $\alpha_{ij} = 1$, and $i \in I$, the GLFP problem simplifies to a linear sum of ratio problems, solvable with methods like the interior point method [27], the Dinkelbach-type approximation algorithm [28], the harmony search heuristic algorithm [29], and the branch-and-bound algorithm [30]. Finally, under the assumption that $d_{ij}^\top x + f_{ij} = 1$ for all $i \in I$, $j \in J_i$, the GLFP can be reformulated as a generalized polynomial programming problem by treating $c_{ij}^\top x + e_{ij}$ as a variable. Relevant solution frameworks are discussed in [31,32].

Among the methods mentioned above, the branch-and-bound algorithm and its hybrid variants have attracted considerable research interest and have been extensively investigated. For instance, Jiao et al. [22,23] introduced an equivalent reformulation of the original problem by incorporating auxiliary variables and applying a logarithmic transformation to the objective function. In [22], the authors further developed a linear relaxation technique based on the first-order mean value theorem. Subsequent work in [23] derived a new linear programming model by exploiting the convex envelope of the logarithmic function. Separately, Shen et al. [33] proposed a novel region deletion technique to remove subdomains that cannot contain the global optimum, conducting the branching process in the original n-dimensional space. More recently, Gao and Zhang [26] introduced a linear relaxation strategy that combines convex envelopes of both logarithmic and bilinear functions, augmented with an output-space reduction mechanism to enhance convergence efficiency. Despite these advances, the computational effort required by such approaches increases substantially as the problem's scale grows, posing a notable challenge for solving large-scale instances.

Very recently, some efficient branch-and-bound methods have been derived for the general case of the GLFP problem. For example, Jiao and Shang [34] developed linear relaxation methods within a branch-and-bound framework, utilizing the convex and concave envelopes of exponential and logarithmic functions for the objective function, along with branching in n-dimensional space; also refer to [35] and [36]. Moreover, Jiao and Li [37], Hou and Liu [38], and Jiao and Ma [39] developed linear relaxation techniques using a convex envelope and a concave envelope for the constraints and geometric properties of exponential and logarithmic functions for the objective function, with the branching operations occurring in an output space. Furthermore, Hou and Liu [40] proposed a branch-and-bound algorithm incorporating a second-order cone relaxation, augmented with a region reduction technique to enhance its computational performance.

Observe that the existing branch-and-bound methods are either based on convex and concave envelope relaxation or second-order cone relaxation. A meaningful question arises: Can a more efficient algorithm for solving large-scale GLFP problems be found by using the hybrid relaxation strategy of convex envelope and a quadratic cone? To answer this question, in this paper, we explore an efficient branch-and-bound algorithm for global optimization of generalized linear fractional programs via hybrid convex envelope and second-order cone relaxations. Initially, the GLFP is transformed into an equivalent problem (EP) by introducing new variables. Then, a new linear relaxation programming problem is proposed, resulting in a lower bound for the optimal value of the EP, utilizing the properties of the arithmetic mean inequality, the convex envelope, and the quadratic cone. Furthermore, a novel outer space branch-and-bound algorithm is established using the branch-and-bound framework. The convergence and complexity of the new algorithm are also analyzed. The effectiveness of the proposed algorithm is demonstrated by numerical experiments.

The highlight of this paper are illustrated as follows. (*i*) We use a hybrid relaxation strategy (convex envelope and a second-order cone) to derive a series of new linear relaxation problems (LRPs) that approximate the EP's optimal value, as opposed to the traditional single strategy of only convex relaxation or second-order cones, as described in [37–40]. Furthermore, Theorem 2.3 indicates that the algorithm constructed from this hybrid relaxation strategy may find the optimal solution of the GLFP faster. (*ii*) In contrast to [33–36], the branching operation of the proposed algorithm occurs in the outcome space, not the n-dimensional one. (*iii*) Our results show that our algorithm outperforms the algorithms in [22, 26, 37] in terms of time efficiency for large-scale problems. Specifically, the time cost of our algorithm is reduced by at least half.

The remainder of this paper is organized as follows. Section 2 formulates an EP for the GLFP and establishes a new linear relaxation programming method to compute the lower bound of the optimal value for the EP. Section 3 devises branching and region reduction techniques. Section 4 presents a branch-and-bound algorithm, along with analyses of its convergence and computational complexity. Numerical experiments in Section 5 demonstrate the validity of the proposed algorithm, and Section 6 concludes the paper.

2. Equivalent form of the GLFP and its LRP

In this section, first, the EP of the GLFP is derived by introducing some outer space variables and performing a series of equivalent transformations. Next, using hybrid relaxation strategy (a convex envelope and a second-order cone), we derive a new LRP that approximates the optimal value of the EP, and we prove the infinite approximation between the EP and the LRP.

2.1. Equivalent transformation of the GLFP

We begin by introducing auxiliary variables to reformulate the problem. For simplicity, and without affecting the generality of the argument, we assume $\alpha_{ij} > 0$ for all $i \in I$ and $j \in J_i$. For cases where $\alpha_{ij} < 0$, the term $\left(\frac{c_{ij}^\top x+e_{ij}}{d_{ij}^\top x+f_{ij}}\right)^{\alpha_{ij}}$ can be equivalently expressed as $\left(\frac{d_{ij}^\top x+f_{ij}}{c_{ij}^\top x+e_{ij}}\right)^{-\alpha_{ij}}$, thereby ensuring that all exponents remain positive. Next, for $i \in I$ and $j \in J_i$, we define $z_{ij} = \frac{c_{ij}^\top x+e_{ij}}{d_{ij}^\top x+f_{ij}}$. To facilitate the subsequent analysis, we compute the tightest possible bounds $\underline{z}_{ij}^0$ (lower bound) and $\bar{z}_{ij}^0$ (upper bound) for each z_{ij} by solving the following fractional programming problems:

$$\underline{z}_{ij}^0 = \min_{x\in\chi} \frac{c_{ij}^\top x + e_{ij}}{d_{ij}^\top x + f_{ij}}, \ \bar{z}_{ij}^0 = \max_{x\in\chi} \frac{c_{ij}^\top x + e_{ij}}{d_{ij}^\top x + f_{ij}}.$$

The lower and upper bounds $\underline{z}_{ij}^0$ and $\bar{z}_{ij}^0$ can be computed by solving a linear program via the Charnes-Cooper transformation [41]. Consequently, we obtain the inequalities $0 < \underline{z}_{ij}^0 \le z_{ij} \le \bar{z}_{ij}^0$ for $i \in I, j \in J_i$, which define an initial outer approximation in the form of a hyper-rectangle:

$$\mathcal{T}^0 = [\underline{z}^0, \bar{z}^0] = \left\{z = (z_{11}, z_{12}, \dots, z_{pT_p})^\top \in \mathbb{R}^{\hat{T}} | \underline{z}_{ij}^0 \le z_{ij} \le \bar{z}_{ij}^0, \ i \in I, \ j \in J_i\right\},$$

where $\underline{z}^0 = (\underline{z}_{11}^0, \underline{z}_{12}^0, \dots, \underline{z}_{pT_p}^0)^\top$, $\bar{z}^0 = (\bar{z}_{11}^0, \bar{z}_{12}^0, \dots, \bar{z}_{pT_p}^0)^\top$, and $\hat{T} = \sum_{i=1}^{p} T_i$ denotes the total dimension of the variable space.

Note that

$$f(x) = \sum_{i=1}^{p} \delta_i \prod_{j=1}^{T_i} \left(\frac{c_{ij}^\top x + e_{ij}}{d_{ij}^\top x + f_{ij}}\right)^{\alpha_{ij}} = \sum_{i=1}^{p} \delta_i \prod_{j=1}^{T_i} z_{ij}^{\alpha_{ij}}.$$

An equivalent reformulation (EP) of the GLFP is given by the following optimization problem:

$$\text{EP}: \begin{cases} \min \ F(z) = \sum_{i=1}^{p} \delta_i \prod_{j=1}^{T_i} z_{ij}^{\alpha_{ij}} \\ s.t. \ \frac{c_{ij}^\top x+e_{ij}}{d_{ij}^\top x+f_{ij}} = z_{ij}, \ i \in I, \ j \in J_i, \\ \quad x \in \chi, \ z \in \mathcal{T}^0. \end{cases}$$

Remark 2.1. According to the framework of the problems of GLFP and EP, it is easy to see that if (z^*, x^*) is an optimal solution to the EP, then x^* is an optimal solution to the GLFP, and the converse is also true.

2.2. Relaxation technique

In this subsection, we will relax the nonlinear terms in the objective function and the nonlinear constraints of the EP, and provide a new lower bound for the optimal value of the EP.

2.2.1. Relaxation of the objective function $F(z)$

At the k-th iteration of the proposed algorithm, the region $\mathcal{T}^k := [\underline{z}^k, \bar{z}^k]$ is defined as the hyper-rectangle in $\mathbb{R}^{\hat{T}}$ given by

$$\mathcal{T}^k = \left\{z^k = (z_{11}^k, \dots, z_{pT_p}^k)^\top \in \mathbb{R}^{\hat{T}} | \underline{z}_{ij}^0 \le \underline{z}_{ij}^k \le z_{ij}^k \le \bar{z}_{ij}^k \le \bar{z}_{ij}^0, \ i \in I, j \in J_i\right\}, \tag{2.1}$$

where the lower bound vector $\underline{z}^k$ and the upper bound vector $\bar{z}^k$ are defined component-wise as

$$\underline{z}^k = (\underline{z}_{11}^k, \underline{z}_{12}^k, \ldots, \underline{z}_{pT_p}^k)^\top,\ \bar{z}^k = (\bar{z}_{11}^k, \bar{z}_{12}^k, \ldots, \bar{z}_{pT_p}^k)^\top.$$

Obviously, $\mathcal{T}^k \subseteq \mathcal{T}^0$. For each $i \in I$, we define $\vartheta_i(z^k) := \sum_{j=1}^{T_i} \alpha_{ij} \ln z_{ij}^k$ such that $\prod_{j=1}^{T_i} (z_{ij}^k)^{\alpha_{ij}} = \exp(\vartheta_i(z^k))$, and denote

$$\vartheta_i^l(z^k) := \sum_{j=1}^{T_i} \alpha_{ij} \ln \underline{z}_{ij}^k \text{ and } \vartheta_i^u(z^k) := \sum_{j=1}^{T_i} \alpha_{ij} \ln \bar{z}_{ij}^k.$$

Given that $\alpha_{ij} > 0$ for all $i \in I$ and $j \in J_i$, it immediately follows that $\vartheta_i^l(z^k) \le \vartheta_i(z^k) \le \vartheta_i^u(z^k)$ for all $i \in I$.

Since $\exp(\vartheta_i(z^k))$ is monotonically increasing over the interval $[\vartheta_i^l(z^k), \vartheta_i^u(z^k)]$, its concave envelope is given by the affine function:

$$A_i(\vartheta_i(z^k) - \vartheta_i^l(z^k)) + \exp(\vartheta_i^l(z^k)),$$

where $A_i = \frac{\exp(\vartheta_i^u(z^k)) - \exp(\vartheta_i^l(z^k))}{\vartheta_i^u(z^k) - \vartheta_i^l(z^k)}$ for $i \in I$. Notably, the tangent to $\exp(\vartheta_i(z^k))$ at $\vartheta_i(z^k) = \ln A_i$ shares the same slope A_i as the concave envelope. Consequently, using the geometric properties of the exponential function and [34], we derive the following tight bounding inequalities for each $i \in I$:

$$A_i(\vartheta_i(z^k) + 1 - \ln A_i) \le \exp(\vartheta_i(z^k)) \le A_i(\vartheta_i(z^k) - \vartheta_i^l(z^k)) + \exp(\vartheta_i^l(z^k)), \tag{2.2}$$

which implies that

$$\begin{aligned} &\sum_{i=1,\delta_i>0}^{p} \delta_i A_i \left(\vartheta_i(z^k) + 1 - \ln A_i\right) + \sum_{i=1,\delta_i<0}^{p} \delta_i \left(A_i(\vartheta_i(z^k) - \vartheta_i^l(z^k)) + \exp(\vartheta_i^l(z^k))\right) \\ &\le \sum_{i=1,\delta_i>0}^{p} \delta_i \exp(\vartheta_i(z^k)) + \sum_{i=1,\delta_i<0}^{p} \delta_i \exp(\vartheta_i(z^k)) = F(z^k). \end{aligned} \tag{2.3}$$

Let $B_{ij} := \frac{\ln \bar{z}_{ij}^k - \ln \underline{z}_{ij}^k}{\bar{z}_{ij}^k - \underline{z}_{ij}^k}$ for $i \in I$, $j \in J_i$. Using the geometric properties of $\ln z_{ij}^k$ on the interval $[\underline{z}_{ij}^k, \bar{z}_{ij}^k]$ and [34], we have

$$B_{ij}(z_{ij}^k - \underline{z}_{ij}^k) + \ln \underline{z}_{ij}^k \le \ln z_{ij}^k \le B_{ij} z_{ij}^k - 1 - \ln B_{ij},$$

implying that

$$\sum_{j=1}^{T_i} \alpha_{ij} \left(B_{ij}(z_{ij}^k - \underline{z}_{ij}^k) + \ln \underline{z}_{ij}^k\right) \le \vartheta_i(z^k) \le \sum_{j=1}^{T_i} \alpha_{ij} \left(B_{ij} z_{ij}^k - 1 - \ln B_{ij}\right).$$

By (2.2) and (2.3), it then follows that

$$h(z^k) := \sum_{i=1,\delta_i>0}^{p} \delta_i h_i^l(z^k) + \sum_{i=1,\delta_i<0}^{p} \delta_i h_i^u(z^k) \le F(z^k), \tag{2.4}$$

where

$$h_i^l(z^k) := A_i\left(\sum_{j=1}^{T_i}\alpha_{ij}\left(B_{ij}(z_{ij}^k - \underline{z}_{ij}^k) + \ln \underline{z}_{ij}^k\right) + 1 - \ln A_i\right),$$

$$h_i^u(z^k) := A_i\left(\sum_{j=1}^{T_i}\alpha_{ij}(B_{ij}z_{ij}^k - 1 - \ln B_{ij}) - \vartheta_i^l(z^k)\right) + \exp(\vartheta_i^l(z^k)).$$

It therefore follows from (2.4) that $h(z^k)$ provide a lower estimate of $F(z^k)$.

2.2.2. Relaxation of the nonlinear constraint $c_{ij}^\top x + e_{ij} = z_{ij}^k(d_{ij}^\top x + f_{ij})$

We now describe a procedure for approximating the nonlinear constraint

$$c_{ij}^\top x + e_{ij} = z_{ij}^k(d_{ij}^\top x + f_{ij})$$

by a system of linear inequalities.

Step 1. Compute the upper and lower bounds for $d_{ij}^\top x + f_{ij}$ over the feasible set χ. These bounds, denoted $\underline{d}_{ij}$ and $\bar{d}_{ij}$, are obtained by solving the linear programs:

$$\underline{d}_{ij} := \min_{x\in\chi} d_{ij}^\top x + f_{ij} \text{ and } \bar{d}_{ij} := \max_{x\in\chi} d_{ij}^\top x + f_{ij}.$$

By construction, for all $x \in \chi$, we have

$$0 < \underline{d}_{ij} \le d_{ij}^\top x + f_{ij} \le \bar{d}_{ij}.$$

Step 2. For each $i \in I$, $j \in J_i$, and $x \in \chi$, and the given bounds

$$0 < \underline{z}_{ij}^k \le z_{ij}^k \le \bar{z}_{ij}^k, 0 < \underline{d}_{ij} \le d_{ij}^\top x + f_{ij} \le \bar{d}_{ij},$$

we approximate the nonlinear relation $z_{ij}^k = \frac{c_{ij}^\top x + e_{ij}}{d_{ij}^\top x + f_{ij}}$ using the auxiliary functions:

$$\varrho_{ij}^k(x) := \max\left\{\theta_{ij}^k(x), \zeta_{ij}^k(x)\right\},\ \eta_{ij}^k(x) := \max\left\{\varrho_{ij}^k(x), \varphi_{ij}^k(x)\right\}, \tag{2.5}$$

where

$$\begin{aligned}
\theta_{ij}^k(x) &:= \frac{c_{ij}^\top x + e_{ij}}{\underline{d}_{ij}} + \bar{z}_{ij}^k - \frac{\bar{z}_{ij}^k}{\underline{d}_{ij}}(d_{ij}^\top x + f_{ij}),\\
\zeta_{ij}^k(x) &:= \frac{c_{ij}^\top x + e_{ij}}{\bar{d}_{ij}} + \underline{z}_{ij}^k - \frac{\underline{z}_{ij}^k}{\bar{d}_{ij}}(d_{ij}^\top x + f_{ij}),\\
\varphi_{ij}^k(x) &:= \frac{4\left(c_{ij}^\top x + e_{ij}\right)}{\underline{d}_{ij} + \underline{z}_{ij}^k + \bar{d}_{ij} + \bar{z}_{ij}^k} + \frac{\left(d_{ij}^\top x + f_{ij} - \frac{c_{ij}^\top x + e_{ij}}{d_{ij}^\top x + f_{ij}}\right)^2 + (\underline{d}_{ij} + \underline{z}_{ij}^k)(\bar{d}_{ij} + \bar{z}_{ij}^k)}{\underline{d}_{ij} + \underline{z}_{ij}^k + \bar{d}_{ij} + \bar{z}_{ij}^k}\\
&\quad - (d_{ij}^\top x + f_{ij}).
\end{aligned} \tag{2.6}$$

As $k \to \infty$, one may verify that $\eta_{ij}^k(x)$ converges to z_{ij}^k for all $i \in I, j \in J_i$.

Theorem 2.1. Let

$$\mathcal{T}^k = \left\{z^k = (z_{11}^k, \ldots, z_{pT_p}^k) \in \mathbb{R}^{\hat{T}} | \underline{z}_{ij}^0 \le \underline{z}_{ij}^k \le z_{ij}^k \le \bar{z}_{ij}^k \le \bar{z}_{ij}^0,\ i \in I, j \in J_i\right\}.$$

For each $i \in I$, $j \in J_i$, and $x \in \chi$, let $\varrho_{ij}^k(x)$ and $\eta_{ij}^k(x)$ be defined as in (2.5). Then, for any $z_{ij}^k \in [\underline{z}_{ij}^k, \bar{z}_{ij}^k]$, $\eta_{ij}^k(x) \le z_{ij}^k$. Moreover, as $k \to \infty$, $\varrho_{ij}^k(x) \to z_{ij}^k$ and $\eta_{ij}^k(x) \to z_{ij}^k$.

Proof. Observe that $c_{ij}^\top x + e_{ij} = z_{ij}^k(d_{ij}^\top x + f_{ij})$. It then follows from (2.6) that

$$\begin{aligned}
z_{ij}^k - \theta_{ij}^k(x) &= z_{ij}^k - \frac{c_{ij}^\top x + e_{ij}}{\underline{d}_{ij}} - \bar{z}_{ij}^k + \frac{\bar{z}_{ij}^k}{\underline{d}_{ij}}\left(d_{ij}^\top x + f_{ij}\right) \\
&= \left(z_{ij}^k - \bar{z}_{ij}^k\right) + \frac{\bar{z}_{ij}^k - z_{ij}^k}{\underline{d}_{ij}}\left(d_{ij}^\top x + f_{ij}\right) \\
&= \frac{\left(\bar{z}_{ij}^k - z_{ij}^k\right)\left(d_{ij}^\top x + f_{ij} - \underline{d}_{ij}\right)}{\underline{d}_{ij}} \\
&\le \frac{\left(\bar{z}_{ij}^k - \underline{z}_{ij}^k\right)\left(d_{ij}^\top x + f_{ij} - \underline{d}_{ij}\right)}{\underline{d}_{ij}}.
\end{aligned} \tag{2.7}$$

Similarly,

$$z_{ij}^k - \zeta_{ij}^k(x) = \frac{\left(z_{ij}^k - \underline{z}_{ij}^k\right)\left(\bar{d}_{ij} - d_{ij}^\top x - f_{ij}\right)}{\bar{d}_{ij}} \le \frac{\left(\bar{z}_{ij}^k - \underline{z}_{ij}^k\right)\left(\bar{d}_{ij} - d_{ij}^\top x - f_{ij}\right)}{\bar{d}_{ij}}. \tag{2.8}$$

By (2.7), (2.8), $\underline{z}_{ij}^k \le z_{ij}^k \le \bar{z}_{ij}^k$ and $\underline{d}_{ij} \le d_{ij}^\top x + f_{ij} \le \bar{d}_{ij}$, we know that $\theta_{ij}^k(z) \le z_{ij}^k$ and $\zeta_{ij}^k(x) \le z_{ij}^k$. Therefore, $\varrho_{ij}^k(x) \le z_{ij}^k$.

We next prove that $\varphi_{ij}^k(x) \le z_{ij}^k$. By the characteristic of the arithmetic mean-geometric inequality, it follows that

$$z_{ij}^k\left(d_{ij}^\top x + f_{ij}\right) = \left(\frac{d_{ij}^\top x + f_{ij} + z_{ij}^k}{2}\right)^2 - \left(\frac{d_{ij}^\top x + f_{ij} - z_{ij}^k}{2}\right)^2. \tag{2.9}$$

Applying a concave envelope of $(\frac{d_{ij}^\top x + f_{ij} + z_{ij}^k}{2})^2$ with respect to

$$\frac{d_{ij}^\top x + f_{ij} + z_{ij}^k}{2} \in [\frac{\underline{d}_{ij} + \underline{z}_{ij}^k}{2}, \frac{\bar{d}_{ij} + \bar{z}_{ij}^k}{2}],$$

we have

$$\left(\frac{d_{ij}^\top x + f_{ij} + z_{ij}^k}{2}\right)^2 \le \frac{\underline{d}_{ij} + \underline{z}_{ij}^k + \bar{d}_{ij} + \bar{z}_{ij}^k}{4}\left(d_{ij}^\top x + f_{ij} + z_{ij}^k\right) - \frac{\left(\underline{d}_{ij} + \underline{z}_{ij}^k\right)\left(\bar{d}_{ij} + \bar{z}_{ij}^k\right)}{4},$$

which, together with (2.9), yields that

$$c_{ij}^\top x + e_{ij} + \left(\frac{d_{ij}^\top x + f_{ij} - z_{ij}^k}{2}\right)^2$$

$$\le \frac{\underline{z}_{ij}^k + \underline{d}_{ij} + \bar{d}_{ij} + \bar{z}_{ij}^k}{4}\left(d_{ij}^\top x + f_{ij} + z_{ij}^k\right) - \frac{\left(\underline{d}_{ij} + \underline{z}_{ij}^k\right)\left(\bar{d}_{ij} + \bar{z}_{ij}^k\right)}{4}.$$

This means that

$$\frac{4\left(c_{ij}^\top x + e_{ij}\right) + \left(d_{ij}^\top x + f_{ij} - \frac{c_{ij}^\top x + e_{ij}}{d_{ij}^\top x + f_{ij}}\right)^2 + \left(\underline{d}_{ij} + \underline{z}_{ij}^k\right)\left(\bar{d}_{ij} + \bar{z}_{ij}^k\right)}{\underline{d}_{ij} + \underline{z}_{ij}^k + \bar{d}_{ij} + \bar{z}_{ij}^k} - \left(d_{ij}^\top x + f_{ij}\right) \le z_{ij}^k,$$

i.e., $\varphi_{ij}^k(x) \le z_{ij}^k$. Hence, $\eta_{ij}^k(x) \le z_{ij}^k$.

When $k \to \infty$, we have $|\bar{z}_{ij}^k - \underline{z}_{ij}^k| \to 0$, and it follows from (2.7) and (2.8) that $\theta_{ij}^k(x) \to z_{ij}^k$ and $\zeta_{ij}^k(x) \to z_{ij}^k$, thus $\varrho_{ij}^k(x) \to z_{ij}^k$, which, together with $\varrho_{ij}^k(x) \le \eta_{ij}^k(x) \le z_{ij}^k$, implies that $\eta_{ij}^k(x) \to z_{ij}^k$. This completes the proof. □

By Theorem 2.1, the function $\eta_{ij}^k(x)$ provides a lower estimate for z_{ij}^k, for all $i \in I$, $j \in J_i$. Accordingly, using (2.4) and Theorem 2.1, the original problem (EP) can be relaxed over the domain $\mathcal{T}^k$ to the following relaxation problem (RP):

$$\text{RP}: \begin{cases} \min & h(z^k) = \sum_{i=1,\delta_i>0}^{p} \delta_i h_i^l(z^k) + \sum_{i=1,\delta_i<0}^{p} \delta_i h_i^u(z^k) \\ s.t. & \eta_{ij}^k(x) \le z_{ij}^k,\ i \in I,\ j \in J_i, \\ & x \in \chi,\ z^k \in \mathcal{T}^k. \end{cases}$$

Note that the expression $\varphi_{ij}^k(x)$ in $\eta_{ij}^k(x)$ is originally nonlinear. To enable efficient computation, we reformulate it into a linear form by introducing an auxiliary variable S_{ij}, defined as $S_{ij} := (\frac{d_{ij}^\top x + f_{ij} - z_{ij}^k}{2})^2$, with $z_{ij}^k = \frac{c_{ij}^\top x + e_{ij}}{d_{ij}^\top x + f_{ij}}$ for $i \in I$, $j \in J_i$. Then, $\varphi_{ij}^k(x)$ becomes

$$\varphi_{ij}^k(x) = \frac{4(c_{ij}^\top x + e_{ij})}{\underline{d}_{ij} + \underline{z}_{ij}^k + \bar{d}_{ij} + \bar{z}_{ij}^k} + \frac{4S_{ij} + (\underline{d}_{ij} + \underline{z}_{ij}^k)(\bar{d}_{ij} + \bar{z}_{ij}^k)}{\underline{d}_{ij} + \underline{z}_{ij}^k + \bar{d}_{ij} + \bar{z}_{ij}^k} - (d_{ij}^\top x + f_{ij}),$$

which is clearly linear in x and S_{ij}. Furthermore, from

$$\left(\frac{d_{ij}^\top x + f_{ij} - z_{ij}^k}{2} - \frac{\underline{d}_{ij} - \bar{z}_{ij}^k + \bar{d}_{ij} - \underline{z}_{ij}^k}{4}\right)^2 \ge 0,$$

and

$$\frac{\underline{d}_{ij} - \bar{z}_{ij}^k}{2} \le \frac{d_{ij}^\top x + f_{ij} - z_{ij}^k}{2} \le \frac{\bar{d}_{ij} - \underline{z}_{ij}^k}{2},$$

it follows that

$$\frac{\underline{d}_{ij} - \bar{z}_{ij}^k + \bar{d}_{ij} - \underline{z}_{ij}^k}{4}(d_{ij}^\top x + f_{ij} - z_{ij}^k) - \frac{(\underline{d}_{ij} - \bar{z}_{ij}^k + \bar{d}_{ij} - \underline{z}_{ij}^k)^2}{16} \le S_{ij}.$$

Using this linear lower bound on S_{ij}, we construct the following LRP as an approximation to the EP over $\mathcal{T}^k$:

$$\text{LRP}:\begin{cases}\min\ h(z^k)=\sum\limits_{i=1,\delta_i>0}^{p}\delta_i h_i^l(z^k)+\sum\limits_{i=1,\delta_i<0}^{p}\delta_i h_i^u(z^k)\\ s.t.\ \theta_{ij}^k(x)\le z_{ij}^k,\ \zeta_{ij}^k(x)\le z_{ij}^k,\ \varphi_{ij}^k(x)\le z_{ij}^k,\\ \quad\frac{\underline{d}_{ij}-\bar{z}_{ij}^k+\bar{d}_{ij}-\underline{z}_{ij}^k}{4}(d_{ij}^\top x+f_{ij}-z_{ij}^k)-\frac{(\underline{d}_{ij}-\bar{z}_{ij}^k+\bar{d}_{ij}-\underline{z}_{ij}^k)^2}{16}\le S_{ij},\\ \quad x\in\chi,\ z^k\in\mathcal{T}^k,\ 0\le S_{ij}\le\frac{\left(\bar{d}_{ij}-\underline{z}_{ij}^k\right)^2}{4},\ i\in I,\ j\in J_i.\end{cases}$$

The following theorem establishes that under asymptotic refinement of the partition (as $k\to\infty$), the optimal value of the LRP converges to the optimal value of the EP.

Theorem 2.2. Let $\mathcal{T}^k$ be the hyper-rectangle defined in (2.1), and let (z^*,x^*,S^*) be an optimal solution of the LRP, where $z^*=(z_{11}^*,z_{12}^*,\ldots,z_{pT_p}^*)^\top\in\mathbb{R}^{\hat{T}}$ with $z_{ij}^*=\frac{c_{ij}^\top x^*+e_{ij}}{d_{ij}^\top x^*+f_{ij}}\in[\underline{z}_{ij}^k,\bar{z}_{ij}^k]$ for $i\in I, j\in J_i$, and $S^*=(\frac{d_{ij}^\top x^*+f_{ij}-z_{ij}^*}{2})^2$. When $k\to\infty$, $\vartheta_i^u(z^k)-\vartheta_i^l(z^k)\to 0$, $h_i^l(z^*)=h_i^u(z^*)=\exp(\vartheta_i(z^*))$, where $\vartheta_i(z^*)=\sum_{j=1}^{T_i}\alpha_{ij}\ln z_{ij}^*$, $\vartheta_i^l(z^k)=\sum_{j=1}^{T_i}\alpha_{ij}\ln\underline{z}_{ij}^k$ and $\vartheta_i^u(z^k)=\sum_{j=1}^{T_i}\alpha_{ij}\ln\bar{z}_{ij}^k$. Consequently, (z^*,x^*) is an optimal solution to the EP.

Proof. By assumption, we have $\vartheta_i(z^*)\in[\vartheta_i^l(z^k),\vartheta_i^u(z^k)]$. For $i\in I$ and $j\in J_i$, as $k\to\infty$, the interval lengths tend to zero ($|\bar{z}_{ij}^k-\underline{z}_{ij}^k|\to 0$), leading to

$$\vartheta_i^u(z^k)-\vartheta_i^l(z^k)=\sum_{j=1}^{T_i}\alpha_{ij}\ln\bar{z}_{ij}^k-\sum_{j=1}^{T_i}\alpha_{ij}\ln\underline{z}_{ij}^k=\sum_{j=1}^{T_i}\alpha_{ij}B_{ij}(\bar{z}_{ij}^k-\underline{z}_{ij}^k)\to 0, \tag{2.10}$$

where $B_{ij}=\frac{\ln\bar{z}_{ij}^k-\ln\underline{z}_{ij}^k}{\bar{z}_{ij}^k-\underline{z}_{ij}^k}$. From the definitions of $h_i^l(z^k)$ and $h_i^u(z^k)$ in (2.4), it follows that

$$\exp(\vartheta_i(z^k))-h_i^l(z^k)=\Omega_i^{1,k}+\Omega_i^{2,k}\text{ and }h_i^u(z)-\exp(\vartheta_i(z))=\Omega_i^{3,k}+\Omega_i^{4,k}, \tag{2.11}$$

where

$$\begin{aligned}\Omega_i^{1,k}&:=\exp(\vartheta_i(z^k))-A_i\left(1+\vartheta_i(z^k)-\ln A_i\right),\\ \Omega_i^{2,k}&:=A_i\sum_{j=1}^{T_i}\alpha_{ij}\left(\ln z_{ij}^k-(B_{ij}z_{ij}^k-B_{ij}\underline{z}_{ij}^k)-\ln\underline{z}_{ij}^k\right),\\ \Omega_i^{3,k}&:=A_i\sum_{j=1}^{T_i}\alpha_{ij}\left(B_{ij}z_{ij}^k-1-\ln B_{ij}-\ln z_{ij}^k\right),\\ \Omega_i^{4,k}&:=A_i\left(\vartheta_i(z^k)-\vartheta_i^l(z^k)\right)+\exp(\vartheta_i^l(z^k))-\exp(\vartheta_i(z^k)),\end{aligned}$$

with $A_i=\frac{\exp(\vartheta_i^l(z^k))(\exp(\vartheta_i^u(z^k)-\vartheta_i^l(z^k))-1)}{\vartheta_i^u(z^k)-\vartheta_i^l(z^k)}$.

We now prove that $\Omega_i^{1,k},\Omega_i^{2,k},\Omega_i^{3,k},\Omega_i^{4,k}\to 0$ as $k\to\infty$, which directly implies $h_i^l(z^k)\to\exp(\vartheta_i(z^k))$ and $h_i^u(z^k)\to\exp(\vartheta_i(z^k))$.

(i) Since $\Omega_i^{1,k}$ is convex in $\vartheta_i(z^k)$, its maximum over $[\vartheta_i^l(z^k), \vartheta_i^u(z^k)]$ is reached at an endpoint. Without losing generality, assume that it occurs at $\vartheta_i^l(z^k)$:

$$\begin{aligned}\Omega_i^{1_{\max},k} &= \exp(\vartheta_i^l(z^k)) - A_i(1 + \vartheta_i^l(z^k) - \ln A_i)\\ &= \exp(\vartheta_i^l(z^k)) - \exp(\vartheta_i^l(z^k))\rho_i\left(1 + \vartheta_i^l(z^k) - \ln\exp(\vartheta_i^l(z^k))\rho_i\right)\\ &= \exp(\vartheta_i^l(z^k))(1 - \rho_i(1 - \ln\rho_i)),\end{aligned}$$

where $\rho_i = \frac{\exp(\vartheta_i^u(z^k)-\vartheta_i^l(z^k))-1}{\vartheta_i^u(z^k)-\vartheta_i^l(z^k)}$. By (2.10), $\vartheta_i^u(z^k) - \vartheta_i^l(z^k) \to 0$ implies $\rho_i \to 1$, so $\Omega_i^{1_{\max},k} \to 0$, which together with $0 \le \Omega_i^{1,k} \le \Omega_i^{1_{\max},k}$ implies that $\Omega_i^{1,k} \to 0$.

(ii) Observing that

$$0 \le \Omega_i^{2,k} \le A_i \sum_{j=1}^{T_i} \alpha_{ij}(\ln \bar{z}_{ij}^k - \ln \underline{z}_{ij}^k - B_{ij}z_{ij}^k + B_{ij}\underline{z}_{ij}^k) \le A_i \sum_{j=1}^{T_i} \alpha_{ij}B_{ij}(\bar{z}_{ij}^k - \underline{z}_{ij}^k),$$

the right-hand side vanishes as $k \to \infty$, proving $\Omega_i^{2,k} \to 0$.

(iii) Since $\Omega_i^{3,k}$ is convex, it reaches its maximum $\Omega_i^{3_{\max},k}$ at $z_{ij}^k = \underline{z}_{ij}^k$ or $z_{ij}^k = \bar{z}_{ij}^k$. Therefore,

$$\begin{aligned}0 \le \Omega_i^{3,k} \le \Omega_i^{3_{\max},k} &= A_i \sum_{j=1}^{T_i} \alpha_{ij}\left(B_{ij}\underline{z}_{ij}^k - 1 - \ln B_{ij} - \ln \underline{z}_{ij}^k\right)\\ &= A_i \sum_{j=1}^{T_i} \alpha_{ij}\left(\frac{\ln \frac{\bar{z}_{ij}^k}{\underline{z}_{ij}^k}}{\frac{\bar{z}_{ij}^k}{\underline{z}_{ij}^k} - 1} - 1 - \ln \frac{\ln \frac{\bar{z}_{ij}^k}{\underline{z}_{ij}^k}}{\frac{\bar{z}_{ij}^k}{\underline{z}_{ij}^k} - 1}\right).\end{aligned}$$

As $k \to \infty$, it follows that $\Omega_i^{3_{\max},k} \to 0$, and hence $\Omega_i^{3,k} \to 0$.

(iv) Since $\Omega_i^{4,k}$ is concave and $\vartheta_i(z^k) = \ln A_i$ satisfies $\nabla\Omega_i^4 = 0$, the function reaches its global maximum at this point. Thus,

$$\begin{aligned}0 \le \Omega_i^{4,k} \le \Omega_i^{4_{\max},k} &= A_i(\ln A_i - \vartheta_i^l(z^k)) + \exp(\vartheta_i^l(z^k)) - \exp(\ln A_i)\\ &= \exp(\vartheta_i^l(z^k)) - A_i\left(\vartheta_i^l(z^k) + 1 - \ln A_i\right).\end{aligned}$$

Substituting the expressions for ρ_i and A_i yields

$$\begin{aligned}0 \le \Omega_i^{4,k} \le \Omega_i^{4_{\max},k} &= \exp(\vartheta_i^l(z^k)) - \exp(\vartheta_i^l(z^k))\rho_i\left(\vartheta_i^l(z^k) + 1 - \vartheta_i^l(z^k) - \ln\rho_i\right)\\ &= \exp(\vartheta_i^l(z^k))\left(1 - \rho_i(1 - \ln\rho_i)\right).\end{aligned}$$

When $k \to \infty$, we have $\vartheta_i^u(z^k) - \vartheta_i^l(z^k) \to 0$, leading to $\rho_i \to 1$, $\Omega_i^{4_{\max},k} \to 0$ and, consequently, $\Omega_i^{4,k} \to 0$.

Since (z^*, x^*, S^*) is an optimal solution to the LRP, (z^*, x^*) is feasible for the EP. From the above analysis, we have

$$\begin{aligned}&0 \le \exp(\vartheta_i(z^*)) - h_i^l(z^*) \le \Omega_i^{1_{\max},k} + \Omega_i^{2,k},\\ &0 \le h_i^u(z^*) - \exp(\vartheta_i(z^*)) \le \Omega_i^{3_{\max},k} + \Omega_i^{4_{\max},k}.\end{aligned}$$

As $k \to \infty$, it follows that $h_i^l(z^*) = \exp(\vartheta_i(z^*))$ and $h_i^u(z^*) = \exp(\vartheta_i(z^*))$. Therefore, (z^*, x^*) is an optimal solution to the EP. The proof is completed. □

2.3. Comparison with existing relaxation methods

Theorem 2.1 establishes the inequality $\varrho_{ij}^k(x) \le \eta_{ij}^k(x)$ for all $i \in I,\ j \in J_i$, which implies that the constraint relaxations proposed in [10, 13, 22, 24] are no tighter than the hybrid strategy introduced in this paper. Moreover, in certain cases, the strict inequality $\varrho_{ij}^k(x) < \eta_{ij}^k(x)$ may hold for some pairs (i, j). This indicates that $\eta_{ij}^k(x)$ can provide a tighter approximation of z_{ij}^k than $\varrho_{ij}^k(x)$, suggesting that algorithms based on $\eta_{ij}^k(x)$ may converge faster to the optimal value of the GLFP. We now present sufficient conditions for this strict improvement.

Theorem 2.3. Suppose that for some $i \in I,\ j \in J_i$, the following conditions hold:

$$\bar{z}_{ij}^k = \underline{d}_{ij},\ \frac{\underline{z}_{ij}^k + \bar{d}_{ij}}{2} < \underline{d}_{ij},\ \frac{(\underline{d}_{ij})^2}{\bar{d}_{ij}} < \underline{z}_{ij}^k.$$

In this case, for $x \in \chi$, such as $z_{ij}^k = \frac{c_{ij}^\top x + e_{ij}}{d_{ij}^\top x + f_{ij}} \in (\underline{z}_{ij}^k, \bar{z}_{ij}^k]$, we have $\varrho_{ij}^k(x) < \eta_{ij}^k(x)$.

Proof. Note that

$$\begin{aligned}\varphi_{ij}^k(x) - \theta_{ij}^k(x) = &\left(\frac{4}{\underline{d}_{ij} + \underline{z}_{ij}^k + \bar{d}_{ij} + \bar{z}_{ij}^k} - \frac{1}{\underline{d}_{ij}}\right)(c_{ij}^\top x + e_{ij}) + \left(\frac{\bar{z}_{ij}^k}{\underline{d}_{ij}} - 1\right)(d_{ij}^\top x + f_{ij})\\ &+ \frac{\left(d_{ij}^\top x + f_{ij} - \frac{c_{ij}^\top x + e_{ij}}{d_{ij}^\top x + f_{ij}}\right)^2 + \left(\underline{d}_{ij} + \underline{z}_{ij}^k\right)(\bar{d}_{ij} + \bar{z}_{ij}^k)}{\underline{d}_{ij} + \underline{z}_{ij}^k + \bar{d}_{ij} + \bar{z}_{ij}^k} - \bar{z}_{ij}^k.\end{aligned}$$

From $\bar{z}_{ij}^k = \underline{d}_{ij}$, $\frac{\underline{z}_{ij}^k + \bar{d}_{ij}}{2} < \underline{d}_{ij}$, it follows that

$$0 < \frac{\underline{d}_{ij} + \underline{z}_{ij}^k + \bar{d}_{ij} + \bar{z}_{ij}^k}{4} < \underline{d}_{ij},\ \frac{\bar{z}_{ij}^k}{\underline{d}_{ij}} = 1,$$

which implies

$$\left(\frac{4}{\underline{d}_{ij} + \underline{z}_{ij}^k + \bar{d}_{ij} + \bar{z}_{ij}^k} - \frac{1}{\bar{d}_{ij}}\right)(c_{ij}^\top x + e_{ij}) > 0,$$

and

$$\left(\frac{\bar{z}_{ij}^k}{\underline{d}_{ij}} - 1\right)\left(d_{ij}^\top x + f_{ij}\right) = 0.$$

From $\left(d_{ij}^\top x + f_{ij} - \frac{c_{ij}^\top x + e_{ij}}{d_{ij}^\top x + f_{ij}}\right)^2 \ge 0$, it follows that

$$\begin{aligned}\varphi_{ij}^k(x) - \theta_{ij}^k(x) &> \frac{(\underline{d}_{ij} + \underline{z}_{ij}^k)(\bar{d}_{ij} + \bar{z}_{ij}^k)}{\underline{d}_{ij} + \underline{z}_{ij}^k + \bar{d}_{ij} + \bar{z}_{ij}^k} - \bar{z}_{ij}^k\\ &= \frac{\bar{d}_{ij}\underline{z}_{ij}^k - (\underline{d}_{ij})^2}{\underline{d}_{ij} + \underline{z}_{ij}^k + \bar{d}_{ij} + \bar{z}_{ij}^k}.\end{aligned}$$

From $\frac{(\underline{d}_{ij})^2}{\bar{d}_{ij}} < \underline{z}_{ij}^k < z_{ij}^k$, we conclude that $\varphi_{ij}^k(x) > \theta_{ij}^k(x)$. A similar argument shows that $\varphi_{ij}^k(x) > \zeta_{ij}^k(x)$. Hence, $\eta_{ij}^k(x) > \varrho_{ij}^k(x)$. □

Theorem 2.3 confirms that the proposed relaxation yields strictly tighter bounds than those in [26, 37–39] under certain conditions. We now compare the optimal value of our RP with the second-order cone relaxation in [40].

For clarity, we first restate the relaxation from [14]. For a given $\mathcal{T}^k \subseteq \mathcal{T}^0$, the relaxed problem in [14] is formulated as:

$$\text{SRP}: \begin{cases} \min & h(z^k) \\ s.t. & z^k_{ij} \geq \varphi^k_{ij}(x),\ z^k_{ij} \geq \phi^k_{ij}(x), i \in I, j \in J_i, \\ & x \in \chi,\ z^k \in \mathcal{T}^k. \end{cases}$$

where $\phi^k_{ij}(x) := \frac{\left(d_{ij}^\top x + f_{ij} + \frac{c_{ij}^\top x + e_{ij}}{d_{ij}^\top x + f_{ij}}\right)^2 - 4\left(c_{ij}^\top x + e_{ij}\right) + (\underline{z}^k_{ij} - \bar{d}_{ij})(\bar{z}^k_{ij} - \underline{d}_{ij})}{\underline{z}^k_{ij} - \bar{d}_{ij} + \bar{z}^k_{ij} - \underline{d}_{ij}} + (d_{ij}^\top x + f_{ij})$, for $i \in I, j \in J_i$.

The following theorem shows that the proposed relaxation (RP) is tighter than (Second-order cone Relaxation Programming, SRP) under certain conditions.

Theorem 2.4. For $\mathcal{T}^k \subseteq \mathcal{T}^0$, under the conditions:

$$\bar{z}^k_{ij} = \underline{d}_{ij},\quad 0 < \underline{d}_{ij} \leq \bar{d}_{ij},$$
$$0 < \underline{z}^k_{ij} \leq z^k_{ij} \leq \min\left\{ \frac{\bar{d}_{ij}\underline{d}_{ij} + \bar{d}_{ij}\underline{z}^k_{ij} + (\underline{d}_{ij})^2 - 2(\bar{d}_{ij})^2}{\bar{d}_{ij}}, \frac{(\underline{d}_{ij})^2 + \bar{d}_{ij}\underline{z}^k_{ij} - (\bar{d}_{ij})^2}{\bar{d}_{ij}} \right\} \leq \bar{z}^k_{ij}.$$

hold for some $i \in I, j \in J_i$. In this case, $\theta^k_{ij}(x) > \phi^k_{ij}(x)$, $\zeta^k_{ij}(x) > \phi^k_{ij}(x)$.

Proof. Under the stated conditions, we have

$$\theta^k_{ij}(x) - \phi^k_{ij}(x) = \frac{c_{ij}^\top x + e_{ij}}{\underline{d}_{ij}} + \bar{z}^k_{ij} - \left(\frac{\bar{z}^k_{ij}}{\underline{d}_{ij}} + 1\right)(d_{ij}^\top x + f_{ij})$$
$$- \frac{\left(d_{ij}^\top x + f_{ij} + \frac{c_{ij}^\top x + e_{ij}}{d_{ij}^\top x + f_{ij}}\right)^2 - 4\left(c_{ij}^\top x + e_{ij}\right) + (\underline{z}^k_{ij} - \bar{d}_{ij})(\bar{z}^k_{ij} - \underline{d}_{ij})}{\underline{z}^k_{ij} - \bar{d}_{ij} + \bar{z}^k_{ij} - \underline{d}_{ij}}.$$

To simplify the description, we denote $d_{ij} := d_{ij}^\top x + f_{ij}$, $c_{ij}^\top x + e_{ij} = d_{ij} z^k_{ij}$. Then according to $\bar{z}^k_{ij} = \underline{d}_{ij}$, we obtain

$$\begin{aligned}
\theta^k_{ij}(x) - \phi^k_{ij}(x) &= \frac{d_{ij}(z^k_{ij} - \bar{z}^k_{ij})}{\underline{d}_{ij}} + (\bar{z}^k_{ij} - d_{ij}) + \frac{(d_{ij} - z^k_{ij})^2}{\bar{d}_{ij} - \underline{z}^k_{ij}} \\
&= \frac{d_{ij}(z^k_{ij} - \bar{z}^k_{ij})(\bar{d}_{ij} - \underline{z}^k_{ij}) + \underline{d}_{ij}(\bar{d}_{ij} - \underline{z}^k_{ij})(\bar{z}^k_{ij} - d_{ij}) + \underline{d}_{ij}(d_{ij} - z^k_{ij})^2}{\underline{d}_{ij}(\bar{d}_{ij} - \underline{z}^k_{ij})} \\
&\geq \frac{(z^k_{ij} - d_{ij})(\bar{d}_{ij} - \underline{z}^k_{ij}) + (\bar{d}_{ij} - \underline{z}^k_{ij})(\bar{z}^k_{ij} - d_{ij}) + (d_{ij} - z^k_{ij})^2}{\bar{d}_{ij} - \underline{z}^k_{ij}} \\
&\geq \frac{z^k_{ij}(\bar{d}_{ij} - \underline{z}^k_{ij}) - \bar{d}_{ij}(\bar{d}_{ij} - \underline{z}^k_{ij}) + \bar{d}_{ij}(\bar{z}^k_{ij} - \bar{d}_{ij}) - \underline{z}^k_{ij}(\bar{z}^k_{ij} - d_{ij})}{\bar{d}_{ij} - \underline{z}^k_{ij}} \\
&\quad + \frac{+d_{ij}(d_{ij} - 2z^k_{ij}) + (z^k_{ij})^2}{\bar{d}_{ij} - \underline{z}^k_{ij}}
\end{aligned}$$

$$\geq \frac{\bar{d}_{ij}\underline{z}_{ij}^k + \bar{d}_{ij}\underline{d}_{ij} + \underline{d}_{ij}^2 - 2(\bar{d}_{ij})^2 - \bar{d}_{ij}z_{ij}^k}{\bar{d}_{ij} - \underline{z}_{ij}^k}.$$

From $\underline{z}_{ij}^k \leq z_{ij}^k \leq \frac{\bar{d}_{ij}\underline{d}_{ij}+\bar{d}_{ij}\underline{z}_{ij}^k+(\underline{d}_{ij})^2-2(\bar{d}_{ij})^2}{\bar{d}_{ij}} \leq \bar{z}_{ij}^k$, it follows that $\theta_{ij}^k(x) \geq \phi_{ij}^k(x)$.

Similarly, from $\bar{z}_{ij}^k = \underline{d}_{ij}$, $0 < \underline{z}_{ij}^k \leq z_{ij}^k \leq \frac{(\underline{d}_{ij})^2+\bar{d}_{ij}\underline{z}_{ij}^k-(\bar{d}_{ij})^2}{\bar{d}_{ij}} \leq \bar{z}_{ij}^k$, we have $\zeta_{ij}^k(x) > \phi_{ij}^k(x)$. Therefore, by

$$0 < \underline{z}_{ij}^k \leq z_{ij}^k \leq \min\left\{\frac{\bar{d}_{ij}\underline{d}_{ij} + \bar{d}_{ij}\underline{z}_{ij}^k + (\underline{d}_{ij})^2 - 2(\bar{d}_{ij})^2}{\bar{d}_{ij}}, \frac{(\underline{d}_{ij})^2 + \bar{d}_{ij}\underline{z}_{ij}^k - (\bar{d}_{ij})^2}{\bar{d}_{ij}}\right\} \leq \bar{z}_{ij}^k,$$

we have $\theta_{ij}^k \geq \phi_{ij}^k(x)$, and $\zeta_{ij}^k(x) \geq \phi_{ij}^k(x)$. It follows that the RP might provides a tighter lower bound than the SRP under the given conditions. □

3. Region reduction techniques

In this section, we present a new region reduction method within the outcome space $\mathbb{R}^{\hat{T}}$. This method is designed to eliminate regions that do not contain the global optimal solution for the EP.

For simplicity, we restrict our reduction operation to the rectangle $\mathcal{T}^k = [\underline{z}^k, \bar{z}^k] \subseteq \mathcal{T}^0$. Given any $z^k \in [\underline{z}^k, \bar{z}^k]$, the inequality (2.4) ensures that $h(z^k) \leq F(z^k)$, which implies

$$\min_{z^k \in [\underline{z}^k, \bar{z}^k]} h(z^k) \leq \min_{z^k \in [\underline{z}^k, \bar{z}^k]} F(z^k) =: F(z^*).$$

Furthermore, from the expressions of $h_i^l(z^k)$ and $h_i^u(z^k)$ in (2.4), we have

$$A_i\left(\sum_{j=1}^{T_i} \alpha_{ij} \ln \underline{z}_{ij}^k + 1 - \ln A_i\right) \leq h_i^l(z^k),$$
$$h_i^u(z) \leq A_i\left(\sum_{j=1}^{T_i} \alpha_{ij}(B_{ij}\bar{z}_{ij} - 1 - \ln B_{ij}) - \vartheta_i^l(z^k)\right) + \exp(\vartheta_i^l(z^k)).$$

Let $\overline{UB}^k := F(z^k)$ denote the upper bound, and define the lower bound $\underline{LB}^k$ as

$$\underline{LB}^k := \sum_{i=1,\delta_i>0}^{p} \delta_i A_i\left(\sum_{j=1}^{T_i} \alpha_{ij} \ln \underline{z}_{ij}^k + 1 - \ln A_i\right)$$
$$+ \sum_{i=1,\delta_i<0}^{p} \delta_i \left(A_i\left(\sum_{j=1}^{T_i} \alpha_{ij}(B_{ij}\bar{z}_{ij}^k - 1 - \ln B_{ij}) - \vartheta_i^l(z^k)\right) + \exp(\vartheta_i^l(z^k))\right).$$

It follows immediately that

$$\underline{LB}^k \leq \min_{z^k \in [\underline{z}^k, \bar{z}^k]} h(z^k) \leq F(z^*) \leq \overline{UB}^k,$$

confirming that $\overline{UB}^k$ and $\underline{LB}^k$ provide rigorous upper and lower bounds, respectively, for the optimal value of the EP over $\mathcal{T}^k$.

Now, we utilize $\overline{UB}^k$ and $\underline{LB}^k$ to eliminate subregions that cannot contain the global optimal solution of the EP. Before that, for each $r \in I$, $s \in J_i$, we write

$$\tau_{rs} := \begin{cases} \frac{\frac{\overline{UB}^k - \underline{LB}^k}{\delta_r A_r} + \alpha_{rs} B_{rs} \underline{z}_{rs}^k}{\alpha_{rs} B_{rs}}, & \text{if } \delta_r > 0, \\ \frac{\frac{\overline{UB}^k - \underline{LB}^k}{\delta_r A_r} + \alpha_{rs} B_{rs} \bar{z}_{rs}^k}{\alpha_{rs} B_{rs}}, & \text{if } \delta_r < 0, \end{cases}$$

$$\overline{\mathcal{T}}_1^k(r,s) := \left\{ z^k \in \mathbb{R}^{\hat{T}} | \underline{z}_{ij}^k \leq z_{ij}^k \leq \bar{z}_{ij}^k,\ i \neq r;\ \tau_{rs} < z_{rs}^k \leq \bar{z}_{rs}^k,\ (i,j) = (r,s) \right\} \subseteq \mathcal{T}^k,$$

and

$$\overline{\mathcal{T}}_2^k(r,s) := \left\{ z^k \in \mathbb{R}^{\hat{T}} | \underline{z}_{ij}^k \leq z_{ij}^k \leq \bar{z}_{ij}^k,\ i \neq r;\ \underline{z}_{rs}^k \leq z_{rs}^k < \tau_{rs},\ (i,j) = (r,s) \right\} \subseteq \mathcal{T}^k.$$

The following theorem, though its proof follows the standard technique in branch-and-bound methods (see, e.g., [26, 39]), is provided here for the sake of logical completeness.

Theorem 3.1. Consider a sub-rectangle $\mathcal{T}^k = [\underline{z}^k, \bar{z}^k] \subseteq \mathcal{T}^0$ at iteration k of the optimization procedure. If $\underline{LB}^k > \overline{UB}^k$, then $\mathcal{T}^k$ contains no global optimal solution to the EP. If $\underline{LB}^k \leq \overline{UB}^k$ and $r \in I$ and $s \in J_i$ exist such that either (i) $\delta_r > 0$ and $\underline{z}_{rs}^k \leq \tau_{rs} < \bar{z}_{rs}^k$, or (ii) $\delta_r < 0$ and $\underline{z}_{rs}^k < \tau_{rs} \leq \bar{z}_{rs}^k$, then the global optimum of the EP cannot lie within the restricted sub-rectangles $\bar{\mathcal{T}}_1^k(r,s)$ or $\bar{\mathcal{T}}_2^k(r,s)$, respectively.

Proof. The proof is shown in [39], so we omit it here. □

4. Algorithm, global convergence, and computational complexity

In this section, we will present an outcome branch-and-bound algorithm for solving the GLFP problem and analyze its convergence and computational complexity.

4.1. An efficient outcome branch-and-bound algorithm

Definition 4.1. Let (z^k, x^k) and v^* be a feasible solution and the global minimum of the EP, respectively. For a given $\varepsilon > 0$, if $F(z^k) - v^* \leq \varepsilon$, then (z^k, x^k) is a global ε-optimal solution of the EP.

Next, we present Algorithm 1 for solving the GLFP problem on the basis of the results above.

Algorithm 1 A novel branch-and-bound algorithm for solving the GLFP problem.

Input. The GLFP problem.

Output. The global optimal solution (z^k, x^k) and the global optimal value $F(z^k) = UB^k$ of the GLFP problem.

Step 0. Given the tolerance error $\varepsilon > 0$, define the initial rectangle

$$\mathcal{T}^0 = \left\{z \in \mathbb{R}^{\hat{T}} | \underline{z}^0_{ij} \le z_{ij} \le \bar{z}^0_{ij}, i \in I, j \in J_i\right\}.$$

The optimal solution (z^0, x^0, S^0) and the initial lower bound $LB^0 = \min_{z \in \mathcal{T}^0} h(z)$ can then be obtained by computing the LRP over $\mathcal{T}^0$. Compute the initial upper bound $UB^0 = F(z^0)$. Let the rectangle set be $H = \{\mathcal{T}^0\}$ and set $k = 0$.

Step 1. (Terminative rule) If $UB^k - LB^k < \varepsilon$, then the algorithm terminates with (z^k, x^k) being a global ε-optimal solution of EP, and $UB^k = F(z^k)$ being an optimal value. Otherwise, let $UB^{k+1} \leftarrow UB^k$, $(z^{k+1}, x^{k+1}) \leftarrow (z^k, x^k)$. Then go to Step 2.

Step 2. (Branching operation) Subdivide the rectangle $\mathcal{T}^k$ into $\mathcal{T}^{k,1}$ and $\mathcal{T}^{k,2}$ by the standard bisection rule, such that $\mathcal{T}^k = \mathcal{T}^{k,1} \cup \mathcal{T}^{k,2}$, where

$$\mathcal{T}^k := \left\{z^k \in \mathbb{R}^{\hat{T}} | \underline{z}^k_{ij} \le z^k_{ij} \le \bar{z}^k_{ij},\ i \in I, j \in J_i\right\} \subseteq \mathcal{T}^0,$$

$$\langle i_0, j_0 \rangle := \arg\max\left\{\bar{z}^k_{ij} - \underline{z}^k_{ij},\ i \in I, j \in J_i\right\},\ z^k_{i_0,j_0} := \frac{\underline{z}^k_{ij} + \bar{z}^k_{ij}}{2}.$$

$$\mathcal{T}^{k,1} = \left\{z^k \in \mathbb{R}^{\hat{T}} | \underline{z}^k_{ij} \le z^k_{ij} \le z^k_{i_0,j_0},\ i \in I, j \in J_i\right\},$$

and

$$\mathcal{T}^{k,2} = \left\{z^k \in \mathbb{R}^{\hat{T}} | z^k_{i_0,j_0} \le z^k_{ij} \le \bar{z}^k_{ij},\ i \in I, j \in J_i\right\}.$$

Denote the set of new subdivided sub-rectangles by $\bar{\mathcal{T}}^k = \{\mathcal{T}^{k,1}, \mathcal{T}^{k,2}\}$.

Step 3. (Regional reduction) For each $\mathcal{T}^{k,i}$, where $i = 1, 2$, use the outcome space region reduction technique to compress its interval, and still denote $\bar{\mathcal{T}}^k$ as the set of the each remaining sub-rectangles.

Step 4. (Updating bounds) If $\bar{\mathcal{T}}^k \neq \emptyset$, then we get a new lower bound LB^k and the solution (z^k, x^k, S^k) by computing the LRP over $\bar{\mathcal{T}}^k$. Obviously, (z^k, x^k) is a feasible solution to the EP and compute $UB^k = F(z^k)$. If $UB^k < UB^{k+1}$, then update the upper bound $UB^{k+1} \leftarrow UB^k$, $(z^{k+1}, x^{k+1}) \leftarrow (z^k, x^k)$. Let $H = H \cup \bar{\mathcal{T}}^k \backslash \mathcal{T}^k$. Select a sub-rectangle $\mathcal{T}^{k+1}$ satisfying that $\mathcal{T}^{k+1} = \arg\min_{\mathcal{T} \in H} LB^k(\mathcal{T})$, set $LB^{k+1} \leftarrow LB^k(\mathcal{T}^{k+1})$.

Step 5. $k \leftarrow k + 1$, and return to Step 1.

4.2. Global convergence

Theorem 4.1. If Algorithm 1 terminates after a finite number of iterations, then (z^k, x^k) is an ε-optimal solution of the EP. If Algorithm 1 produces an infinite solution sequence $\left\{(z^k, x^k)\right\}$, then accumulation point (z^*, x^*) of $\left\{(z^k, x^k)\right\}$ is an ε-optimal solution to the EP.

Proof. Suppose that Algorithm 1 terminates at the k-th iteration. According to the termination conditions of Algorithm 1, we have

$$UB^k - LB^k \le \varepsilon,$$

and (z^k, x^k, S^k) is a feasible solution of the LRP over $\mathcal{T}^k$. Thus, (z^k, x^k) is a feasible solution of the EP over $\mathcal{T}^k$.

By the lower bound and upper bound updating methods and the structure characteristics of Algorithm 1, it follows that

$$LB^k \le v^* \le F(z^k) = UB^k,$$

where v^* is the global minimum of the EP. Therefore, $F(z^k) - v^* \le \varepsilon$. This implies that (z^k, x^k) is an ε-optimal solution of the EP.

Suppose that Algorithm 1 is infinite. It will then produce an infinite feasible solution sequence $\{(z^k, x^k, S^k)\}$ by computing the LRP over $\mathcal{T}^k$. It is obvious that $\{(z^k, x^k)\}$ is a feasible solution sequence of the EP over $\mathcal{T}^k$. Since $x^k \in \chi$ and χ is a nonempty and bounded set, it holds that the accumulation point of $\left\{x^k\right\}$ exists. Without loss of generality, assume that x^* is a cluster point of the sequence $\{x^k\}$, i.e., $\lim_{k\to\infty} x^k = x^*$. For each $i \in I$ and $j \in J_i$, by the continuity of the affine function $\frac{c_{ij}^\top x^k + e_{ij}}{d_{ij}^\top x^k + f_{ij}}$ and the exhaustiveness of the branching rule, we have

$$\lim_{k\to\infty} \frac{c_{ij}^\top x^k + e_{ij}}{d_{ij}^\top x^k + f_{ij}} = \lim_{k\to\infty} z_{ij}^k = \lim_{k\to\infty} \bigcap_k \left[\underline{z}_{ij}^k, \overline{z}_{ij}^k\right] = z_{ij}^*.$$

and $\lim_{k\to\infty} S^k = \lim_{k\to\infty} (\frac{d_{ij}^\top x^k + f_{ij} - z^k}{2})^2 = (\frac{d_{ij}^\top x^* + f_{ij} - z^*}{2})^2 = S^*$. It means that (z^*, x^*) is a feasible solution of the EP over $\mathcal{T}^k$.

Because $\{LB^k\}$ is a nondecreasing lower-bound sequence with $LB^k < v^*$, we have

$$\lim_{k\to\infty} LB^k = \lim_{k\to\infty} h(z^k) \le v^* \le F(z^*). \tag{4.1}$$

Moreover, $|\overline{z}_{ij}^k - \underline{z}_{ij}^k| \to 0$, $k \to \infty$, from Theorem 2.2 and the function continuity of $h(z)$ and $F(z)$, it holds that

$$\exp(\vartheta_i(z^k)) - h_i^l(z^k) \to 0,\ h_i^u(z^k) - \exp(\vartheta_i(z^k)) \to 0,\ i \in I,$$

which implies that

$$\sum_{i=1,\delta_i>0}^{p} \delta_i \left(\exp(\vartheta_i(z^k)) - h_i^l(z^k)\right) + \sum_{i=1,\delta_i<0}^{p} \delta_i \left(h_i^u(z^k) - \exp(\vartheta_i(z^k))\right) \to 0.$$

Thus,

$$F(z^*) = \lim_{k\to\infty} F(z^k) = \lim_{k\to\infty} h(z^k) = h(z^*),$$

which together with (4.1) implies that

$$\lim_{k\to\infty} LB^k = h(z^*) = v^* = F(z^*) = \lim_{k\to\infty} F(z^k) = \lim_{k\to\infty} UB^k.$$

Hence, the accumulation point (z^*, x^*, S^*) of the infinite solution sequence $\left\{(z^k, x^k, S^k)\right\}$ exists, and (z^*, x^*) is a global optimal solution to the EP. The proof is completed. □

4.3. Computational complexity

In this subsection, we will provide an estimate of the number of iterations in the worst case. Let

$$d(\mathcal{T}^k) := \max_{i\in I, j\in J_i} \left\{\bar{z}_{ij}^k - \underline{z}_{ij}^k\right\},$$

$$\gamma := \max_{i\in I} \{A_i\},\ \delta := \max_{i\in I} \{|\delta_i|\},\ \lambda := \max_{i\in I, j\in J_i} \left\{\frac{2\alpha_{ij}}{\underline{z}_{ij}^k}\right\}. \tag{4.2}$$

Theorem 4.2. Let $\varepsilon > 0$ be a prescribed tolerance threshold. Consider the rectangle $\mathcal{T}^k \subseteq \mathcal{T}^0$ generated at the the k-th iteration of Algorithm 1, as defined in (2.1). If Algorithm 1 iterates to step k with $d(\mathcal{T}^k) < \frac{\varepsilon}{\delta\gamma\lambda\hat{T}}$, then for any feasible solution (z^k, x^k, S^k) to the LRP over $\mathcal{T}^k$, $F(z^k) - h(z^k) \le \varepsilon$.

Proof. Assume that Algorithm 1 operates at k-th iteration, generating the tuple (z^k, x^k, S^k). By construction, this tuple is feasible for the LRP, which implies that (z^k, x^k) is also feasible for the EP. For each $i \in I$, observe that $\vartheta_i(z^k) = \sum_{j=1}^{T_i} \alpha_{ij} \ln z_{ij}^k$, $\vartheta_i^l(z^k) = \sum_{j=1}^{T_i} \alpha_{ij} \ln \underline{z}_{ij}^k$, and $\vartheta_i^u(z^k) = \sum_{j=1}^{T_i} \alpha_{ij} \ln \bar{z}_{ij}^k$. By $\underline{z}_{ij}^k \le z_{ij}^k \le \bar{z}_{ij}^k$, it follows that $\vartheta_i(z^k) \in [\vartheta_i^l(z^k), \vartheta_i^u(z^k)]$. By computations, we have

$$F(z^k) - h(z^k) = \sum_{i=1,\delta_i>0}^{p} \delta_i(\Omega_i^{1,k} + \Omega_i^{2,k}) + \sum_{i=1,\delta_i<0}^{p} \delta_i(\Omega_i^{3,k} + \Omega_i^{4,k}),$$

where $\Omega_i^{1,k}$, $\Omega_i^{2,k}$, $\Omega_i^{3,k}$, $\Omega_i^{4,k}$ are defined in (2.11).

We next give the upper bounds for $\Omega_i^{1,k}$, $\Omega_i^{2,k}$, $\Omega_i^{3,k}$, and $\Omega_i^{4,k}$. Since $\Omega_i^{1,k}$ is convex in $\vartheta_i(z^k)$, its maximum over $[\vartheta_i^l(z^k), \vartheta_i^u(z^k)]$ is reached at the boundary:

$$\Omega_i^{1_{\max},k} = \exp(\vartheta_i^l(z^k)) - A_i(1 + \vartheta_i^l(z^k) - \ln A_i).$$

Similarly, $\Omega_i^{4,k}$ is concave and reaches its maximum at $\vartheta_i(z^k) = \ln A_i$:

$$\begin{aligned}\Omega_i^{4_{\max},k} &= A_i\left(\ln A_i - \vartheta_i^l(z^k)\right) + \exp\left(\vartheta_i^l(z^k)\right) - \exp\left(\ln A_i\right)\\ &= \exp\left(\vartheta_i^l(z^k)\right) - A_i\left(1 + \vartheta_i^l(z^k) - \ln A_i\right).\end{aligned}$$

Note that $\Omega_i^{1_{\max},k} = \Omega_i^{4_{\max},k}$ and $\Omega_i^{4,k} \le \Omega_i^{4_{\max},k}$. Using $B_{ij} \le \frac{1}{\underline{z}_{ij}^k}$ and the properties of the exponential function, we derive

$$\begin{aligned}\Omega_i^{1,k} \le \Omega_i^{1_{\max},k} &= \Omega_i^{4_{\max},k}\\ &\le \max_{z^k\in\mathcal{T}^k} \left\{\vartheta_i(z^k)) - A_i(1 + \vartheta_i(z^k) - \ln A_i\right\}\\ &= \max_{z^k\in\mathcal{T}^k} \left\{A_i(\vartheta_i(z^k) - \vartheta_i^l(z^k)) + \exp(\vartheta_i^l(z^k)) - \exp(\vartheta_i(z^k))\right\}\\ &\le \max_{z^k\in\mathcal{T}^k} \left\{\exp(\vartheta_i^u(z^k)) - \exp(\vartheta_i^l(z^k)) + \exp(\vartheta_i^l(z^k)) - \exp(\vartheta_i(z^k))\right\}\\ &\le \max_{z^k\in\mathcal{T}^k} \left\{\exp(\vartheta_i^u(z^k)) - \exp(\vartheta_i^l(z^k))\right\}\end{aligned}$$

$$\leq A_i \sum_{j=1}^{T_i} \alpha_{ij} B_{ij} \left(\bar{z}_{ij}^k - \underline{z}_{ij}^k \right)$$

$$\leq \gamma \sum_{j=1}^{T_i} \frac{\alpha_{ij} \left(\bar{z}_{ij}^k - \underline{z}_{ij}^k \right)}{\underline{z}_{ij}^k},$$

and $\Omega_i^{4,k} \leq \gamma \sum_{j=1}^{T_i} \frac{\alpha_{ij}\left(\bar{z}_{ij}^k - \underline{z}_{ij}^k\right)}{\underline{z}_{ij}^k}$.

The maximum of $\Omega_i^{2,k}$ occurs at $z_{ij}^k = \frac{1}{B_{ij}}$:

$$\Omega_i^{2\max,k} = A_i \sum_{j=1}^{T_i} \alpha_{ij} \left(-\ln B_{ij} - 1 + B_{ij} \underline{z}_{ij}^k - \ln \underline{z}_{ij}^k \right).$$

The maximum of $\Omega_i^{3,k}$ is reached at $z_{ij}^k \in \{\underline{z}_{ij}^k, \bar{z}_{ij}^k\}$:

$$\Omega_i^{3\max,k} = A_i \sum_{j=1}^{T_i} \alpha_{ij} \left(B_{ij} \underline{z}_{ij}^k - 1 - \ln B_{ij} - \ln \underline{z}_{ij}^k \right).$$

Observe that $\Omega_i^{2\max,k} = \Omega_i^{3\max,k}$ and $\Omega_i^{2,k} \leq \Omega_i^{2\max,k}$. Hence,

$$\begin{aligned}
\Omega_i^{3,k} \leq \Omega_i^{3\max,k} &= \Omega_i^{2\max,k} \\
&\leq \max_{z^k \in \mathcal{T}^k} A_i \sum_{j=1}^{T_i} \alpha_{ij} \left(B_{ij} \underline{z}_{ij}^k - 1 - \ln B_{ij}^k - \ln \underline{z}_{ij}^k \right) \\
&= \max_{z^k \in \mathcal{T}^k} A_i \sum_{j=1}^{T_i} \alpha_{ij} \left(\ln z_{ij}^k - B_{ij} z_{ij}^k + B_{ij} \underline{z}_{ij}^k - \ln \underline{z}_{ij}^k \right) \\
&\leq A_i \sum_{j=1}^{T_i} \alpha_{ij} B_{ij} \left(\bar{z}_{ij}^k - \underline{z}_{ij}^k \right) \\
&\leq \gamma \sum_{j=1}^{T_i} \frac{\alpha_{ij} \left(\bar{z}_{ij}^k - \underline{z}_{ij}^k \right)}{\underline{z}_{ij}^k},
\end{aligned}$$

and $\Omega_i^{2,k} \leq \gamma \sum_{j=1}^{T_i} \frac{\alpha_{ij}\left(\bar{z}_{ij}^k - \underline{z}_{ij}^k\right)}{\underline{z}_{ij}^k}$.

Therefore, by $d(\mathcal{T}^k) < \frac{\varepsilon}{\delta \gamma \lambda \hat{T}}$, it holds that

$$\begin{aligned}
F(z^k) - h(z^k) &= \sum_{i=1, \delta_i > 0}^{p} \delta_i \left(\Omega_i^{1,k} + \Omega_i^{2,k} \right) + \sum_{i=1, \delta_i < 0}^{p} \delta_i \left(\Omega_i^{3,k} + \Omega_i^{4,k} \right) \\
&\leq \sum_{i=1}^{p} \delta \gamma \left(\sum_{j=1}^{T_i} \frac{2\alpha_{ij}}{\underline{z}_{ij}^k} \left(\bar{z}_{ij}^k - \underline{z}_{ij}^k \right) \right)
\end{aligned}$$

$$
\begin{aligned}
&\leq \sum_{i=1}^{p} \delta\gamma \sum_{j=1}^{T_i} \lambda d(\mathcal{T}^k) \\
&\leq \delta\gamma\lambda\hat{T} \cdot \frac{\varepsilon}{\delta\gamma\lambda\hat{T}} \\
&= \varepsilon.
\end{aligned}
$$

This completes the proof. □

By Theorem 4.2, we derive the worst-case computational complexity of Algorithm 1, characterizing its time and space complexity asymptotically.

Theorem 4.3. Let $\varepsilon > 0$ be a prescribed tolerance threshold. Suppose Algorithm 1 runs for at most $\left\lfloor \prod_{i=1}^{p}\prod_{j=1}^{T_i} \frac{\delta\gamma\lambda\hat{T}(\bar{z}_{ij}^0 - \underline{z}_{ij}^0)}{\varepsilon} - 1 \right\rfloor$ iterations, where δ, γ, and λ are defined in (4.2), with $\hat{T} = \sum_{i=1}^{p} T_i$, and $\underline{z}_{ij}^0$, $\bar{z}_{ij}^0$, which define the lower and upper initial search bounds, respectively. Algorithm 1 then guarantees convergence to an ε-optimal solution of the EP.

Proof. After Algorithm 1 completes k generations and terminates, according to Theorem 4.2, we have

$$
\bar{z}_{ij}^k - \underline{z}_{ij}^k \leq \frac{\varepsilon}{\delta\gamma\lambda\hat{T}}, \; i \in I, j \in J_i, \tag{4.3}
$$

which is a sufficient condition for the termination of Algorithm 1. By the branching method used in Algorithm 1, when the initial rectangle $\mathcal{T}^0$ operates for k generations, we can generate $k+1$ sub-rectangles, denoted as $\mathcal{T}^\iota$, $\iota \in \{1, \dots, k+1\}$, and the longest edge of $\mathcal{T}^\iota$, denoted as $\bar{z}_{ij}^\iota - \underline{z}_{ij}^\iota$. Then, all values of $\mathcal{T}^\iota$ satisfy (4.3) , which is the worst case for Algorithm 1's termination. This implies that if we take $\frac{\varepsilon}{\delta\gamma\lambda\hat{T}}$ as one side of the rectangle, with its volume denoted as $V(\mathcal{T}^\iota)$, then, the total volume of $k+1$ sub-rectangles no less than $V(\mathcal{T}^0)$, which is formed by the initial interval $[\underline{z}_{ij}^0, \bar{z}_{ij}^0]$, i.e.,

$$
V(\mathcal{T}^0) =: \prod_{i=1}^{p}\prod_{j=1}^{T_i} (\bar{z}_{ij}^0 - \underline{z}_{ij}^0) \leq (k+1) \prod_{i=1}^{p}\prod_{j=1}^{T_i} \frac{\varepsilon}{\delta\gamma\lambda\hat{T}} := (k+1)V(\mathcal{T}^\iota).
$$

This implies that

$$
k \geq \prod_{i=1}^{p}\prod_{j=1}^{T_i} \frac{\delta\gamma\lambda\hat{T}(\bar{z}_{ij}^0 - \underline{z}_{ij}^0)}{\varepsilon} - 1.
$$

Let $k = \left\lfloor \prod_{i=1}^{p}\prod_{j=1}^{T_i} \frac{\delta\gamma\lambda\hat{T}(\bar{z}_{ij}^0 - \underline{z}_{ij}^0)}{\varepsilon} - 1 \right\rfloor$. After Algorithm 1 runs k generations, all sub-rectangles will be deleted. Therefore, Algorithm 1 terminates by at most k iterations. The proof is completed. □

5. Numerical experiments

This section compares the performance of Algorithm 1 with several existing methods [22, 23, 26, 37, 40], highlighting its efficacy and computational advantages. We use MATLAB 2017b to implement all algorithms and run all computational experiments on an Intel(R) Core(TM) i7-9700M central processing unit (CPU) with 3.0 GHz and 8 GB of memory. For each (p, m, n), average numerical results were obtained from 10 distinct randomly generated instances of different scales. The SCIP Optimization Suite (v5.0.1) [42] was included as a benchmark solver. To ensure a fair comparison, all methods, including SCIP, were subject to a uniform time limit of 3600 seconds and a feasibility tolerance ε of 10^{-6}. Beyond these two conditions, SCIP was used with its default configuration, without any problem-specific tuning. The notation used in Tables 1–7, and Figures 1–2 is given as follows:

- n is the number of variables;
- m is the number of constraints;
- N is the number of iterations;
- T is the CPU time in seconds;
- $Avg.Iter$ is the average number of iterations;
- $Avg.Time$ is the average CPU time in seconds;
- $Opt.V$ is the average of the optimal values;
- " – " indicates that no global optimal solution was found within 3600 s.

We start by introducing some specific examples of GLFPs; see Problems A.1-A.12 in Appendix A. Applying Algorithm 1, we obtain globally optimal solutions for these problems, with the corresponding computational results summarized in Table 1. The data demonstrate that Algorithm 1 reliably converges to the global optimum in finite time across all test cases. Compared with existing algorithms of [22, 23, 26, 37, 40], Algorithm 1 is capable of achieving the globally optimal solution in a shorter time and with fewer iterations, as illustrated by Problems A.1, A.2, A.8, A.9, A.11 and A.12 Algorithm 1 is superior to algorithms of [22,23,37,40] especially for Problems A,1, A.8, A.9, A.11, A.12. For Problem A.10 the number of iterations of the algorithm in [40] is fewer than that in Algorithm 1, but its CPU time is longer than that of Algorithm 1, which might be influenced by the more constraints in the GLFP of [40]. In Problems A.3, A.4, A.5 and A.7, both Algorithm 1 and the algorithm in [26] get the same optimal solution with zero iterations. Problem A.6shows Algorithm 1's CPU time exceeding that of the algorithm in [26], yet Algorithm 1 is faster and uses fewer iterations than those in [37, 40].

Next, we evaluate the performance of Algorithm 1 on Problems 1–4, with the computational results detailed in Tables 2–7. To further benchmark efficacy, we compare Algorithm 1 against the commercial solver SCIP (v5.0.1) [42] on Problems 3 and 4.

Problem 1. *([22])*

$$\begin{cases} \min \ \prod_{i=1}^{p} c_i^\top x \\ s.t.\ Ax \le b,\ 0 \le x_j \le 1,\ j = 1, \ldots, n, \end{cases}$$

where $c_i \in \mathbb{R}^n$ *is generated randomly in* $[0, 1]$, $i = 1, \ldots, p$, $A = [a_{uj}] \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, a_{uj} *is randomly generated in* $[-1, 1]$, $b_u = \sum_{j=1}^{n} a_{ij} + 2\pi$, *and* π *is generated randomly in* $[0, 1]$.

Table 1. Numerical comparisons of the algorithms in [22, 23, 26, 37, 40] and Algorithm 1 on Problems A.1-A.12

Problems	Algorithms	Optimal value	Optimal solution	N	T
A.1	Algorithm 1	10.0	(2;8)	0	0.0053
	[23]	10.0	(2;8)	0	0.10
	[22]	10.0	(2;8)	3	0.12
A.2	Algorithm 1	0.9012	(8.0;0.0;1.0)	6	0.0717
	[26]	0.9012	(8.0;0.0;1.0)	8	0.08
	[22]	0.9012	(0.0;8.0;1.0)	9	0.09
A.3	Algorithm 1	997.6613	(1.0;1.0)	0	0.0063
	[26]	997.6613	(1.0;1.0)	0	0.01
	[22]	997.6613	(1.0;1.0)	0	0.01
A.4	Algorithm 1	9504.0	(1.0;2.0;1.0;1.0)	0	0.0059
	[26]	9504.0	(1.0;2.0;1.0;1.0)	0	0.01
	[22]	9504.0	(1.0;2.0;1.0;1.0)	4	0.14
A.5	Algorithm 1	263.7889	(1.25;1.0)	0	0.0055
	[26]	263.7889	(1.25;1.0)	0	0.01
	[22]	263.7889	(1.25;1.0)	0	0.01
A.6	Algorithm 1	0.5333	(0.0;0.0)	4	0.0690
	[26]	0.5333	(0.0;0.0)	5	0.06
	[40]	0.5333	(0.0;0.0)	8	0.4800
	[37]	0.5333	(0.0;0.0)	50	0.6326
A.7	Algorithm 1	5.0987	(1.5;1.5)	0	0.0093
	[26]	5.0987	(1.5;1.5)	0	0.01
A.8	Algorithm 1	0.3360	(1.0;1.0)	0	0.0063
	[40]	0.3360	(1.0;1.0)	1	0.3028
	[37]	0.3360	(1.0;1.0)	23	0.3285
A.9	Algorithm 1	3.0000	(0;0;0)	1	0.0078
	[37]	3.0000	(0;0;0)	19	0.2423
A.10	Algorithm 1	3.0029	(0;0.3333;0)	23	0.2682
	[40]	3.0029	(0;0.3333;0)	7	0.2765
	[37]	3.0029	(0;0.3333;0)	41	0.5300
A.11	Algorithm 1	4.0907	(1.1111;0;0)	16	0.1839
	[37]	4.0907	(1.1111;0;0)	47	0.5755
A.12	Algorithm 1	3.7109	(0;1.6667;0)	0	0.0073
	[37]	3.7109	(0;1.6667;0)	29	0.3790

As illustrated in Table 2, for small and medium-sized cases, Algorithm 1 requires less computation time to obtain the global optimal solutions and optimal values compared with the algorithm presented in [22], with fewer iterations in most cases. Furthermore, Table 3 shows that Algorithm 1 performs better than the algorithm in [22] when the problem becomes large, such as $n \geq 1000$. Hence, Algorithm 1 outperforms the algorithm in [22] for small and large-scale problems, as these results suggest.

Table 2. Computational results for Problem 1.

(p, m, n)	Algorithm 1		Algorithm in [22]	
	Avg.Time	Avg.Iter	Avg.Time	Avg.Iter
(4, 10, 20)	0.2596	17.0	3.0331	37.2
(4, 20, 40)	0.4101	24.1	3.7998	57.9
(4, 40, 80)	0.5333	25.0	5.0930	41.9
(4, 60, 120)	1.2071	28.7	7.0007	32.0
(4, 80, 160)	2.0042	29.6	9.8870	26.8
(4, 100, 200)	3.5820	32.7	16.6237	27.2
(5, 10, 20)	0.4663	22.7	6.8133	96.9
(5, 20, 40)	0.5797	37.3	5.1644	68.7
(5, 40, 80)	1.4521	69.1	10.3765	78.8
(5, 60, 120)	1.8752	51.2	14.8812	65.9
(5, 80, 160)	3.7175	59.9	20.2917	51.1
(5, 100, 200)	4.8400	46.3	38.6381	52.8
(6, 10, 20)	0.8378	54.3	12.4855	159.7
(6, 20, 40)	1.5304	95.2	9.9100	116.7
(6, 40, 80)	2.1361	102.8	19.9002	144.4
(6, 60, 120)	2.4773	66.1	23.4874	98.2
(6, 80, 160)	4.7217	78.2	42.5319	98.7
(6, 100, 200)	7.7401	76.7	43.9363	63.7
(7, 10, 20)	1.1113	69.3	18.3431	302.6
(7, 20, 40)	1.3346	77.5	23.0972	289.2
(7, 40, 80)	3.3852	132.8	33.8311	228.8
(7, 60, 120)	6.9436	161.3	50.3251	208.5
(7, 80, 160)	12.9583	176.1	66.4867	141.9
(7, 100, 200)	21.6252	189.1	96.7868	119.7

Table 3. Computational results for Problem 1.

(p, m, n)	Algorithm 1		Algorithm in [22]	
	Avg.Time	Avg.Iter	Avg.Time	Avg.Iter
(2, 10, 1000)	1.6568	30.6	11.1218	38.5
(2, 10, 2000)	3.0183	31.1	25.0182	43.5
(3, 10, 1000)	2.9143	52.2	35.6843	177.2
(3, 10, 2000)	10.5400	102.3	131.6528	293.1
(4, 10, 1000)	9.5725	193.3	219.1879	939.3
(4, 10, 2000)	57.1688	557.8	457.9584	928.9

Problem 2. *([22])*

$$\begin{cases} \min \ \prod_{i=1}^{p} (c_i^\top x + e_{0i})^{\alpha_i} \\ s.t.\ Ax \le b,\ 0 \le x_j \le 1,\ j = 1, \ldots, n, \end{cases}$$

where $c_i \in \mathbb{R}^n$ *is generated randomly in* $(0, 1)$*,* $i = 1, \ldots, p$*, and* $A = [a_{uj}] \in \mathbb{R}^{m \times n}$*,* $b \in \mathbb{R}^m$*,* a_{uj} *is randomly generated in* $[-1, 1]$*,* $b_u = \sum_{j=1}^{n} a_{ij} + 2\pi$*, and* π *is generated randomly in* $(0, 1)$*.*

Table 4. Computational results for Problem 2 with $\alpha_{ij} = 0.5$.

(p, m, n)	Algorithm 1			Algorithm in [22]		
	Avg.Time	Avg.Iter	Opt.V	Avg.Time	Avg.Iter	Opt.V
(2, 10, 50)	0.3928	21.8	0.4945	0.5707	28.1	0.4945
(2, 20, 100)	0.5145	26.2	0.5579	0.8967	32.1	0.5579
(2, 30, 150)	0.7945	31.0	0.7279	1.5607	39.0	0.7279
(2, 40, 200)	1.2551	33.1	0.6844	2.7118	39.9	0.6844
(3, 10, 50)	0.5522	31.7	0.3350	1.1257	60.8	0.3350
(3, 20, 100)	1.0425	49.3	0.6779	1.6268	58.8	0.6779
(3, 30, 150)	1.5719	58.1	0.4994	2.9726	71.8	0.4994
(3, 40, 200)	2.2203	57.7	0.7199	4.9715	72.1	0.7199
(4, 10, 50)	0.7698	37.5	13.3934	1.2385	47.4	13.3934
(4, 20, 100)	2.5817	117.8	0.4361	8.0883	288.4	0.4361
(4, 30, 150)	3.6565	128.5	0.5120	10.3702	258.4	0.5120
(4, 40, 200)	3.8522	101.9	0.5015	11.6464	177.2	0.5015

For Problem 2, the average CPU time, average number of iterations, and average optimal values for Algorithm 1 and the algorithm of [22] are shown through varying the variables p, m, n, and α_{ij} in Table 4 and Table 5. Table 4 and Table 5 show that Algorithm 1 requires less computation time and shorter iterations than the one in [22]. Specially, it can be observed from Table 5 that when the problem become larger, such as $(p, m) = (5,10)$ $(n \geq 8000)$, the algorithm in [22] is unable to achieve the optimal value within 3600 s, whereas Algorithm 1 is capable of achieving the optimal solution within a short time. This comparison indicates that Algorithm 1 has superior performance to the algorithm in [22], particularly for large-scale problems.

Table 5. Computational results for Problem 2 with α_{ij}=0.5.

(p, m, n)	Algorithm 1			Algorithm in [22]		
	Avg.Time	Avg.Iter	Opt.V	Avg.Time	Avg.Iter	Opt.V
(2, 10, 1000)	1.8491	36.1	0.3628	3.3935	40.6	0.3628
(2, 10, 2000)	3.7187	39.0	0.5624	6.4321	39.8	0.5624
(2, 10, 3000)	7.9804	49.7	0.4047	14.1641	51.2	0.4047
(2, 10, 5000)	17.0612	57.4	0.4955	33.2540	62.5	0.4955
(2, 10, 8000)	19.6805	47.2	0.2780	34.5936	46.4	0.2780
(2, 10, 10000)	49.1859	73.1	0.2951	96.0236	77.9	0.2951
(3, 10, 1000)	3.5106	64.5	0.2964	7.5364	74.0	0.2964
(3, 10, 2000)	5.3749	58.3	0.4263	9.5797	62.1	0.4263
(3, 10, 3000)	12.1256	69.5	0.3689	32.5706	94.6	0.3689
(3, 10, 5000)	18.1653	70.1	0.3199	35.9152	80.9	0.3199
(3, 10, 8000)	38.7729	74.0	0.2505	86.77815	92.8	0.2505
(3, 10, 10000)	91.4636	98.4	0.2442	148.3293	104.6	0.2442
(4, 10, 1000)	4.6530	74.9	0.2708	8.3739	91.7	0.2708
(4, 10, 2000)	9.7803	97.2	0.6583	21.4960	127.3	0.6583
(4, 10, 3000)	24.6549	93.6	1.0720	28.5296	102.1	1.0720
(4, 10, 5000)	67.4885	195.1	0.8474	81.6866	157.6	0.8474
(4, 10, 8000)	146.9179	2704	0.1630	985.0867	727.3	0.1630
(4, 10, 10000)	245.9049	316.7	0.2371	1243.8572	824.4	0.2371
(5, 10, 1000)	10.0476	169.1	0.2246	50.1596	438.5	0.2246
(5, 10, 2000)	18.4853	173.5	0.2767	89.1863	435.6	0.2767
(5, 10, 3000)	56.1142	281.4	0.1665	227.1090	616.3	0.1665
(5, 10, 5000)	118.7390	352.0	0.1166	1385.5302	1190.0	0.1166
(5, 10, 8000)	802.2497	660.3	0.1349	-	-	-
(5, 10, 10000)	1040.0678	1369.3	0.0912	-	-	-

Problem 3. *([26])*

$$\begin{cases} \min \ \prod_{i=1}^{p} \left(\frac{c_i^\top x + e_{0i}}{d_i^\top x + f_{0i}} \right)^{\alpha_i} \\ s.t. \ Ax \le b, \ x_j \ge 0, \ j = 1, \dots, n, \end{cases}$$

where $c_i, d_i \in \mathbb{R}^n$ are generated randomly in $(0, 1)$; e_{0i} and $f_{0i} \in \mathbb{R}^p$ are generated randomly in $(0, 1)$; α_i is generated randomly in $[-1, 1]$, for $i = 1, \dots, p$; $A = [a_{uj}] \in \mathbb{R}^{m\times n}$; $b \in \mathbb{R}^m$, a_{uj} is randomly generated in $(0, 1)$; and b_u is randomly generated in $(1, 16)$, for $u = 1, \dots, m$, $j = 1, \dots, n$.

Table 6. Computational results for Problem 3 with α_{ij}=0.5.

(p, m, n)	Algorithm 1			Algorithm in [26]			SCIP [42]	
	Avg.Time	Avg.Iter	Opt.V	Avg.Time	Avg.Iter	Opt.V	Avg.Time	Opt.V
(2, 10, 50)	0.3776	18.3	0.3752	0.8311	27.0	0.3752	1168.0263	0.3752
(2, 20, 100)	0.4186	21.4	0.3134	1.1785	43.6	0.3134	2409.5348	0.4651
(2, 30, 150)	0.5430	22.3	0.4651	1.5908	39.7	0.4651	-	-
(2, 40, 200)	0.7876	22.4	0.2421	2.2077	40.3	0.2421	-	-
(2, 10, 1000)	3.3465	45.6	0.2458	4.3588	46.9	0.2458	-	-
(2, 10, 2000)	8.7498	55.2	0.2042	13.2679	66.3	0.2042	-	-
(2, 10, 3000)	15.3142	58.7	0.2243	35.9812	104.2	0.2243	-	-
(2, 10, 4000)	34.4107	85.4	0.1287	51.4714	107.9	0.1287	-	-
(2, 10, 5000)	46.7653	86.8	0.1328	165.8337	200.6	0.1328	-	-
(2, 10, 7000)	130.6360	137.7	0.1489	211.9754	203.9	0.1489	-	-
(2, 10, 10000)	210.1779	145.2	0.0999	444.8836	195.2	0.0999	-	-
(3, 10, 50)	1.1328	85.6	0.2296	2.8848	171.3	0.2296	-	-
(3, 20, 100)	2.9714	151.7	0.3162	3.1969	142.3	0.3162	-	-
(3, 30, 150)	4.4305	166.1	0.2577	9.4298	256.0	0.2577	-	-
(3, 40, 200)	6.2807	159.8	0.2341	21.8577	370.0	0.2341	-	-
(3, 10, 1000)	29.2481	349.1	0.3013	37.6139	399.6	0.3013	-	-
(3, 10, 2000)	42.1558	242.4	0.4627	91.9684	461.5	0.4627	-	-
(3, 10, 3000)	136.0869	342.3	0.0536	307.6231	942.7	0.0536	-	-
(3, 10, 4000)	342.1486	702.3	0.1291	544.9074	1027.7	0.1291	-	-
(3, 10, 5000)	892.9827	1201.2	0.1776	1244.4783	1876.5	0.1776	-	-
(3, 10, 7000)	1693.6653	1607.7	0.2628	4170.4162	3605.0	0.2628	-	-
(3, 10, 10000)	2460.9515	1440.2	0.2217	4999.2831	2819.6	0.2217	-	-
(4, 10, 50)	4.2109	195.3	0.2730	11.4525	604.9	0.2730	-	-
(4, 20, 100)	11.2016	544.0	0.3040	39.2108	1214.0	0.3040	-	-
(4, 30, 150)	48.5485	1460.6	0.2301	101.0215	2127.5	0.2301	-	-
(4, 40, 200)	91.9194	1966.5	0.1612	180.5980	2522.5	0.1612	-	-
(4, 10, 1000)	188.2229	2230.1	0.1248	692.2981	5771.0	0.1248	-	-
(4, 10, 2000)	705.1371	3502.2	0.1275	1519.5548	6009.7	0.1275	-	-
(4, 10, 3000)	1056.6647	1935.9	0.0856	2135.0781	5102.2	0.0856	-	-
(4, 10, 4000)	2131.9208	3617.5	0.2169	4251.8661	8602.7	0.2169	-	-

For Problem 3, Table 6 demonstrates that Algorithm 1 achieves significantly lower average CPU times than both the method in [26] and SCIP [42] as dimensions scale: from $(p, m, n) = (2, 10, 50)$ to $(3, 10, 10000)$ and from $(4, 10, 50)$ to $(4, 10, 4000)$. Crucially, SCIP fails to return optimal solutions within 3600 seconds for large-scale GLFPs, including instances where $(p, m, n) = (2, 30, 150)$ or $(2, 10, 1000)$. Figure 1 also shows that for fixed n or p, Algorithm 1 maintains consistently lower computational times than [26] under increasing variable counts, with its CPU time's growth rate substantially slower. These results confirm the superior efficacy of our constraint relaxation approach over both [26] and SCIP.

Problem 4. *([37])*

$$\begin{cases} \min \ \sum_{i=1}^{p} \prod_{j=1}^{T_i} \left(\frac{c_{ij}^{\top} x + e_{ij}}{d_{ij}^{\top} x + f_{ij}} \right)^{\alpha_{ij}} \\ s.t.\ Ax \le b,\ x_s \ge 0,\ s = 1, \ldots, n, \end{cases}$$

where c_{ij} *and* $d_{ij} \in \mathbb{R}^n$ *are generated randomly in* $(0, 1)$*;* e_{ij} *and* $f_{ij} \in \mathbb{R}^{\hat{T}}$ *are generated randomly in* $[0, 1]$*, where* $\hat{T} = \sum_{i=1}^{p} T_i$*;* α_{ij} *is generated randomly in* $[-1, 1]$*, for* $i = 1, \ldots, p,\ j = 1, \ldots, T_i$*, and* $A = [a_{us}] \in \mathbb{R}^{m \times n}$*,* $b \in \mathbb{R}^m$*;* a_{us} *is randomly generated in* $(-1, 1)$*;* b_u *is randomly generated in* $(0, 16)$*;* $u = 1, \ldots, m,\ s = 1, \ldots, n$*; and the error parameter* ϵ *is set to* 10^{-2}.

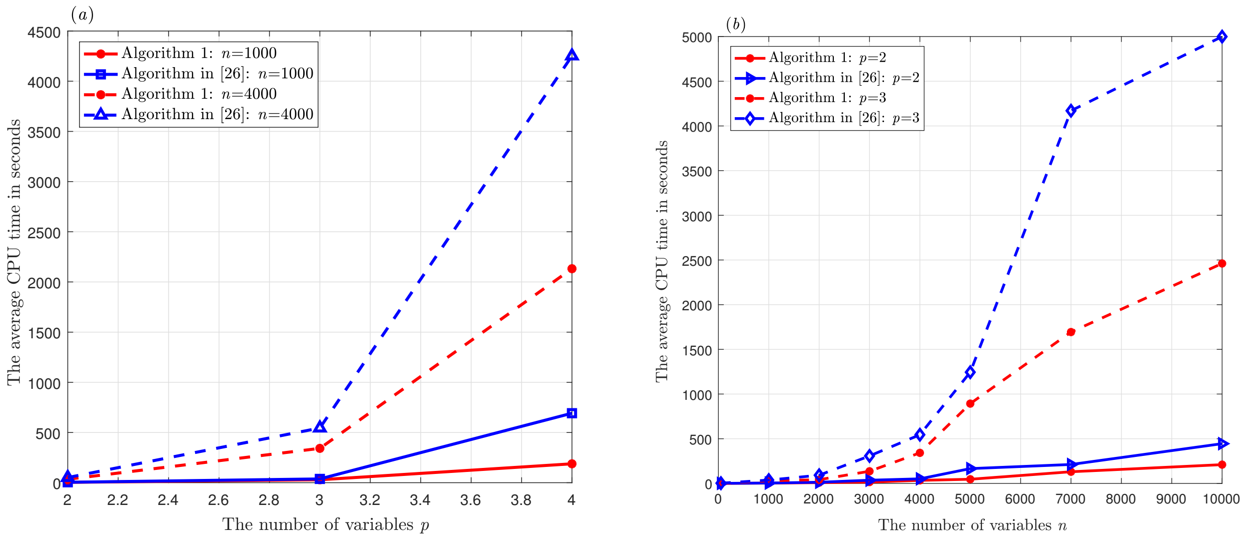

Figure 1. Average CPU time vs. the number of terms (p) for a fixed n, and vs. the number of variables (n) for a fixed p and $m = 10$, for Problem 3.

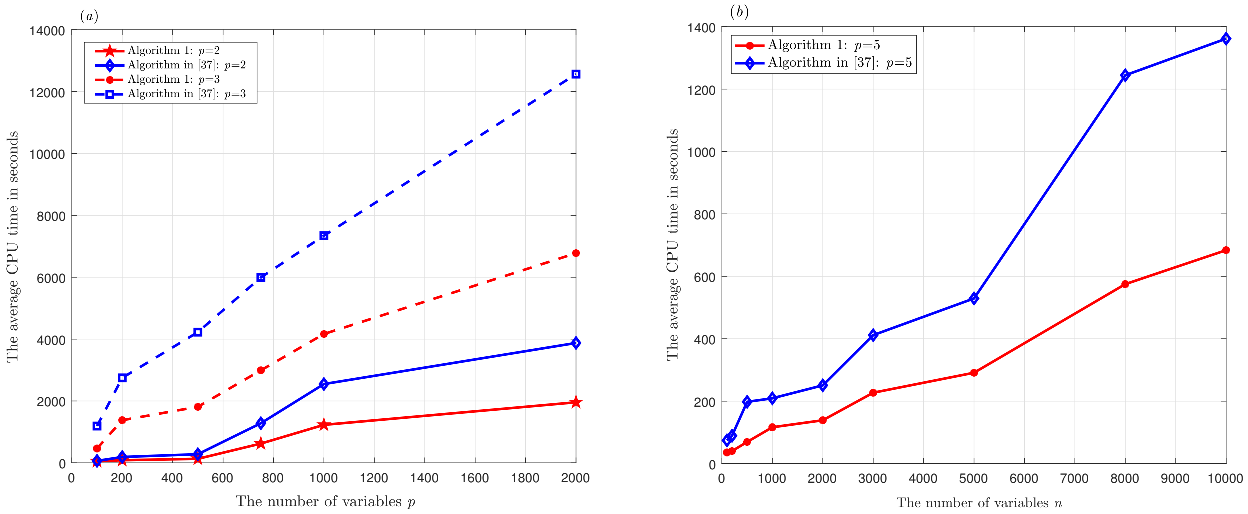

Figure 2. Average CPU time vs. the number of variables n, comparing Algorithm 1 with the Algorithm in [37] for a fixed p, for Problem 4.

Table 7. Computational results for Problem 4.

(p, m, n, T_i)	Algorithm 1			Algorithm in [37]			SCIP [42]	
	Avg.Time	Avg.Iter	Opt.V	Avg.Time	Avg.Iter	Opt.V	Avg.Time	Opt.V
$(2, 50, 100, (2, 2))$	52.2324	1200.6	0.5312	63.8149	1180.8	0.5312	-	-
$(2, 50, 200, (2, 2))$	85.4465	1464.6	0.2984	190.0560	2108.7	0.2984	-	-
$(2, 50, 500, (2, 2))$	130.3383	991.5	0.6119	279.2672	1311.5	0.6119	-	-
$(2, 50, 750, (2, 2))$	623.2772	3471.6	0.1747	1281.0841	4363.68	0.1747	-	-
$(2, 50, 1000, (2, 2))$	1228.5585	3919.1	0.4738	2547.3767	5570.0	0.4738	-	-
$(2, 50, 2000, (2, 2))$	1958.0393	2911.9	0.1558	3875.3023	3502.6	0.1558	-	-
$(3, 50, 100, (2, 2, 2))$	464.2858	6908.2	1.3332	1190.0045	12205.0	1.1332	-	-
$(3, 50, 200, (2, 2, 2))$	1378.8555	13839.2	0.7035	2747.1914	19164.0	0.7035	-	-
$(3, 50, 500, (2, 2, 2))$	1811.9399	11253.5	0.4335	4224.8549	15688.8	0.4335	-	-
$(3, 50, 750, (2, 2, 2))$	2991.0653	14001.0	0.6483	5993.7798	16002.6	0.6483	-	-
$(3, 50, 1000, (2, 2, 2))$	4164.9419	13142.3	0.5878	7342.1465	12820.2	0.5878	-	-
$(3, 50, 2000, (2, 2, 2))$	6776.9837	8654.3	0.8104	12565.7074	10058.3	0.8104	-	-
$(5, 50, 100, (1, 1, 1, 1, 1))$	35.6169	733.8	2.8027	74.2599	1171.9	2.8027	-	-
$(5, 50, 200, (1, 1, 1, 1, 1))$	39.9546	492.9	2.5358	88.9999	733.9	2.5358	-	-
$(5, 50, 500, (1, 1, 1, 1, 1))$	69.2629	340.1	2.2246	197.7948	576.8	2.2246	-	-
$(5, 50, 1000, (1, 1, 1, 1, 1))$	116.3252	327.9	2.2101	209.1596	365.5	2.2101	-	-
$(5, 50, 2000, (1, 1, 1, 1, 1))$	138.7250	162.3	1.6987	250.5667	194.6	1.6987	-	-
$(5, 50, 3000, (1, 1, 1, 1, 1))$	226.9995	159.9	1.8427	411.4861	221.0	1.8427	-	-
$(5, 50, 5000, (1, 1, 1, 1, 1))$	291.0383	119.4	1.8494	529.2023	159.4	1.8494	-	-
$(5, 50, 8000, (1, 1, 1, 1, 1))$	574.9253	111.6	1.7759	1244.2626	190.5	1.7759	-	-
$(5, 50, 10000, (1, 1, 1, 1, 1))$	683.7482	112.0	1.7191	1361.5168	153.8	1.7191	-	-

For Problem 4, Table 7 demonstrates that Algorithm 1 achieves the global optimum with significantly reduced computational effort compared with both the method in [37] and the SCIP solver [42], exhibiting fewer iterations and lower CPU time. Crucially, SCIP fails to converge within the 3600-second time limit for all tested instances, As evidenced by the last two columns. Furthermore, Algorithm 1 reduces the average CPU runtime of [37] by at least 50% while demonstrating superior scalability: As the problem's dimension increases, it has a lower probability of exceeding the time limit. Figure 2 reveals distinct complexity profiles when fixing the parameter p: Algorithm 1 exhibits linear time growth in n with a moderate slope, whereas [37] displays linear growth with a steeper gradient. This contrast underscores our method's enhanced efficiency for large-scale GLFPs.

As shown numerically in Tables 5–7 and Figures 1(b) and 2(b), the average CPU time of the proposed algorithm exhibits clearly distinct scaling behavior with respect to the number of variables n and the parameter p.

Holding p, m, and T_i fixed, the CPU time grows nearly linearly with n. For example, with $(p, m) = (3, 10)$, Table 6 shows a gradual increase in time as n rises from 50 to 10,000, a trend that is also visible in Figure 1(b). Similarly, under a fixed $(p, m, T_i) = (5, 50, (1, 1, 1, 1, 1))$, Table 7 and Figure 2(b) confirm an approximately linear growth with n.

By contrast, when n, m, and T_i are fixed, the CPU time increases sharply with p, suggesting nearly exponential complexity. For instance, at $(m, n) = (10, 1000)$, time rises rapidly from $p = 2$ to $p = 3$, exceeding 3600 s at $p = 4$ (Table 6), a trend that is also visible in Figure 1(a).

The following aspects of the algorithm help explain these computational characteristics.

- **Reduced search dimensionality**: Rather than branching in the original n-dimensional variable space, the algorithm operates in the outcome space $\hat{\mathcal{T}}$, where $\dim(\hat{\mathcal{T}}) \ll n$, greatly lowering combinatorial complexity.

- **Preemptive region reduction**: The technique from Section 3 removes non-optimal regions before branching, tightening the search space dynamically and avoiding costly exploration of irrelevant areas.

- **Tight relaxation bounds**: Theorems 2.3 and 2.4 ensure that the hybrid relaxation provides sharp lower bounds, enabling early pruning of large suboptimal regions and reducing iterations significantly below pessimistic worst-case estimates.

6. Conclusions

This paper has introduced an efficient outcome-space branch-and-bound algorithm for the global solution of large-scale GLFPs. The method is built on a hybrid relaxation strategy that effectively combines convex envelope and second-order cone constraints. Numerical experiments demonstrate the computational advantages of the proposed algorithm, particularly in solving high-dimensional instances.

Future work will explore extensions of the hybrid relaxation methodology to broader classes of nonconvex optimization problems. Promising research directions include the integration of machine learning techniques for adaptive relaxation selection, the development of advanced linear transformation strategies to improve the reformulation quality, and the implementation of parallel computing frameworks with efficient data structures to accelerate the branch-and-bound process.

Use of Generative-AI tools declaration

If there is nothing to disclose, Please state "The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article".

Author contributions

X. Jing (first author): conceptualization, methodology, data curation, visualization, writing-original draft; Y.L. Gao (second author, corresponding author): conceptualization, writing-review and editing; Xiaoli. Huang (third author): methodology, writing-original draft.

Acknowledgments

This work is supported by the National Natural Science Foundations of China (12461053); the Basic Discipline Research Projects supported by Nanjing Securities (NJZQJCXK202201); the Construction Project of First-Class Subjects in Ningxia Higher Education (NXYLXK2017B09); the Natural Science Foundations of Shaanxi Province, China (2025JC-YBMS-025); the Doctoral Research Startup Project of Yan'an University, China (YDBK2024-46).

Conflict of interest

The authors declare no conflict of interest.

References

1. M. Palmgren, M. Rönnqvist, P. Värbrand, A solution approach for log truck scheduling based on composite pricing and branch and bound, *Int. T. Oper. Res.*, **10** (2003), 433–447. https://doi.org/10.1111/1475-3995.00420
2. X. Li, Z. W. He, N. M. Wang, A branch-and-bound algorithm for the proactive resource-constrained project scheduling problem with a robustness maximization objective, *Comput. Oper. Res.*, **166** (2024), 106623. https://doi.org/10.1016/j.cor.2024.106623
3. M. Barkhagen, S. García, J. Gondzio, J. Kalcsics, J. Kroeske, S. Sabanis, et al., Optimising portfolio diversification and dimensionality, *J. Glob. Optim.*, **85** (2023), 185–234. https://doi.org/10.1007/s10898-022-01202-7
4. H. Konno, M. Inori, Bond portfolio optimization by bilinear fractional programming, *J. Oper. Res. Soc. Jpn.*, **32** (1989), 143–158. https://doi.org/10.15807/jorsj.32.143
5. H. Konno, T. Suzukia, D. Kobayashi, A branch and bound algorithm for solving mean-risk-skewness portfolio models, *Optim. Method. Soft.*, **10** (1998), 297–317. https://doi.org/10.1080/10556789808805716
6. C. D. Maranas, I. P. Androulakis, C. A. Floudas, A. J. Berger, J. M. Mulvey, Solving long-term financial planning problems via global optimization, *J. Econ. Dyn. Control*, **21** (1997), 1405–1425. https://doi.org/10.1016/S0165-1889(97)00032-8

7. B. Grimstad, B. Foss, R. Heddle, M. Woodman, Global optimization of multiphase flow networks using spline surrogate models, *Comput. Chem. Eng.*, **84** (2016), 237–254. https://doi.org/10.1016/j.compchemeng.2015.08.022

8. F. Pecci, E. Abraham, I. Stoianov, Global optimality bounds for the placement of control valves in water supply networks, *Optim. Eng.*, **20** (2019), 457–495. https://doi.org/10.1007/s11081-018-9412-7

9. K. Bot, I. Laouali, A. Ruano, M. D. G. Ruano, Home energy management systems with branch-and-bound model-based predictive control techniques, *Energies*, **14** (2021), 5852. https://doi.org/10.3390/en14185852

10. S. A. Cetegen, M. D. Stuber, Optimal design of controlled environment agricultural systems under market uncertainty, *Comput. Chem. Eng.*, **149** (2021), 107285. https://doi.org/10.1016/j.compchemeng.2021.107285

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

12. H. Konno, Parametric simplex algorithms for solving a special class of nonconvex minimization problems, *J. Glob. Optim.*, **1** (1991), 65–82. https://doi.org/10.1007/BF00120666

13. P. P. Shen, B. D. Huang, L. F. Wang, Range division and linearization algorithm for a class of linear ratios optimization problems, *J. Comput. Appl. Math.*, **350** (2019), 324–342. https://doi.org/10.1016/j.cam.2018.10.038

14. R. M. Oliveira, P. A. V. Ferreira, An outcome space approach for generalized convex multiplicative programs, *J. Glob. Optim.*, **47** (2010), 107–118. https://doi.org/10.1007/s10898-009-9460-9

15. H. P. Benson, G. M. Boger, Outcome-space cutting-plane algorithm for linear multiplicative programming, *J. Optim. Theory Appl.*, **104** (2000), 301–322. https://doi.org/10.1023/A:1004657629105

16. P. Bonami, A. Lodi, J. Schweiger, A. Tramontaniet, Solving quadratic programming by cutting planes, *SIAM J. Optim.*, **29** (2019), 1076–1105. https://doi.org/10.1137/16M107428X

17. P. P. Shen, D. X. Wu, Y. F. Wang, An efficient spatial branch-and-bound algorithm using an adaptive branching rule for linear multiplicative programming, *J. Comput. Appl. Math.*, **426** (2023), 115100. https://doi.org/10.1016/j.cam.2023.115100

18. C. F. Wang, Y. P. Deng, P. P. Shen, A novel convex relaxation-strategy-based algorithm for solving linear multiplicative problems, *J. Comput. Appl. Math.*, **407** (2023), 114080. https://doi.org/10.1016/j.cam.2021.114080

19. E. A. Y. Envelope, Level set algorithm for solving convex multiplicative programming problems, *Appl. Math. Comput.*, **167** (2005), 1412–1417. https://doi.org/10.1016/j.amc.2004.08.028

20. S. Y. Liu, Y. F. Zhao, An effcient algorithm for globally solving generalized linear multiplicative programming, *J. Comput. Appl. Math.*, **296** (2016), 840–847. https://doi.org/10.1016/j.cam.2015.11.009

21. X. L. Huang, Y. L. Gao, An effcient outer space branch-and-bound algorithm for globally minimizing linear multiplicative problems, *AIMS Math.*, **8** (2023), 26045–26069. https://doi.org/10.3934/math.20231327

22. H. W. Jiao, W. J. Wang, J. B. Yin, Y. L. Shang, Image space branch-reduction-bound algorithm for globally minimizing a class of multiplicative problems, *RAIRO-Oper. Res.*, **56** (2022), 1533–1552. https://doi.org/10.1051/ro/2022061

23. H. W. Jiao, W. J. Wang, Y. L. Shang Outer Space branch-reduction-bund algorithm for solving generalizaed affine multiplicative problems, *J. Comput. Appl. Math.*, **419** (2023), 114784. https://doi.org/10.1016/j.cam.2022.114784

24. Y. L. Gao, C. X. Xu, Y. J. Yang, An outcome-space finite algorithm for solving linear multiplicative programming, *Appl. Math. Comput.*, **179** (2006), 494–505. https://doi.org/10.1016/j.amc.2005.11.111

25. B. Zhang, H. Y. Wang, Y. L. Gao, Output-space outer approximation branch-and-bound algorithm for a class of linear multiplicative programs, *J. Optim. Theory Appl.*, **202** (2024), 997–1026. https://doi.org/10.1007/s10957-024-02461-y

26. Y. L. Gao, B. Zhang, Output-space branch-and-bound reduction algorithm for generalized linear fractional-multiplicative programming problem, *Chaos Soliton. Fract.*, **175** (2023), 113924. https://doi.org/10.1016/j.chaos.2023.113924

27. R. W. Freund, F. Jarre, Solving the sum-of-ratios problem by an interior-point method, *J. Glob. Optim.*, **19** (2001), 83–102. https://doi.org/10.1023/A:1008316327038

28. R. Precup, A. Orzan, Dinkelbach type approximation algorithms for nonlinear fractional optimization problems, *Numer. Funct. Anal. Optim.*, **44** (2023), 954–969. https://doi.org/10.1080/01630563.2023.2217893

29. M. Jaberipour, E. Khorram, Solving the sum-of-ratios problems by a harmony search algorithm, *J. Comput. Appl. Math.*, **234** (2010), 733–742. https://doi.org/10.1016/j.cam.2010.01.013

30. B. D. Huang, P. P. Shen, An efficient branch and bound reduction algorithm for globally solving linear fractional programming problems, *Chaos Soliton. Fract.*, **182** (2024), 114757. https://doi.org/10.1016/j.chaos.2024.114757

31. Y. J. Wang, Z. A. Liang, A deterministic global optimization algorithm for generalized geometric programming, *Math. Comput.*, **168** (2005), 722–737. https://doi.org/10.1016/j.amc.2005.01.142

32. H. W. Jiao, B. B. Li, W. Yang. A criterion-space branch-reduction-bound algorithm for solving generalized multiplicative problems, *J. Glob. Optim.*, **89** (2024), 597–632. https://doi.org/10.1007/s10898-023-01358-w

33. P. P. Shen, X. D. Bai, W. M. Li, A new accelerating method for globally solving a class of nonconvex programming problems, *Nonlinear Anal.*, **71** (2009), 2866–2876. https://doi.org/10.1016/j.na.2009.01.142

34. H. W. Jiao, Y. L. Shang, Two-level linear relaxation method for generalized linear fractional programming, *J. Oper. Res. Soc. China*, **11** (2023), 569–594. https://doi.org/10.1007/s40305-021-00375-4

35. P. P. Shen, C. F. Wang, Global optimization for sum of generalized fractional functions, *J. Comput. Appl. Math.*, **214** (2008), 1–12. https://doi.org/10.1016/j.cam.2007.01.022

36. H. W. Jiao, Y. R. Guo, P. P. Shen, Global optimization of generalized linear fractional programming with nonlinear constraints, *Appl. Math. Comput.*, **183** (2006), 717–728. https://doi.org/10.1016/j.amc.2006.05.102

37. H. W. Jiao, B. B. Li, Y. L. Shang, An outer space approach to tackle generalized affine fractional program problems, *J. Optim. Theory Appl.*, **201** (2024), 1–35. https://doi.org/10.1007/s10957-023-02368-0

38. Z. S. Hou, S. Y. Liu, An efficient image space branch-reduction-bound algorithm to globally solve generalized fractional programming problems for large-scale real applications, *J. Comput. Appl. Math.*, **451** (2024), 116070. https://doi.org/10.1016/j.cam.2024.116070

39. H. W. Jiao, J. Q. Ma, Optimizing generalized linear fractional program using the image space branch-reduction-bound scheme, *Optimization*, **74** (2024), 1–32. https://doi.org/10.1080/02331934.2023.2253816

40. Z. S. Hou, S. Y. Liu, A spatial branch-reduction-bound algorithm for solving generalized linear fractional problems globally, *Chaos Soliton. Fract.*, **176** (2023), 114144. https://doi.org/10.1016/j.chaos.2023.114144

41. A. Charnes, W. W. Cooper, Programming with linear fractional functionals, *Nav. Res. Logist. Q.*, **9** (1962), 181–186.

42. A. Gleixner, L. Eifler, T. Gally, G. Gamrath, P. Gemander, R. L. Gottwald, et al., *The SCIP Optimization Suite, version 5.0.1*, Zuse Institute Berlin (ZIB), Berlin, Germany, 2017. Available from: `https://www.scipopt.org/index.php/download` [Accessed 2 July 2025].