



Two-Level Linear Relaxation Method for Generalized Linear Fractional Programming

Hong-Wei Jiao^{1,2} · You-Lin Shang²

Received: 17 March 2020 / Revised: 29 September 2021 / Accepted: 31 October 2021

© Operations Research Society of China, Periodicals Agency of Shanghai University, Science Press, and Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

This paper presents an efficient algorithm for globally solving a generalized linear fractional programming problem. For establishing this algorithm, we firstly construct a two-level linear relaxation method, and by utilizing the method, we can convert the initial generalized linear fractional programming problem and its subproblems into a series of linear programming relaxation problems. Based on the branch-and-bound framework and linear programming relaxation problems, a branch-and-bound algorithm is presented for globally solving the generalized linear fractional programming problem, and the computational complexity of the algorithm is given. Finally, numerical experimental results demonstrate the feasibility and efficiency of the proposed algorithm.

Keywords Generalized linear fractional programming · Global optimization · Two-level linear relaxation method · Branch-and-bound

Mathematics Subject Classification 90C26 · 90C32

This work was supported by the National Natural Science Foundation of China (Nos. 11871196, 12071133 and 12071112), the China Postdoctoral Science Foundation (No. 2017M622340), the Key Scientific and Technological Research Projects of Henan Province (Nos. 202102210147 and 192102210114), the Science and Technology Climbing Program of Henan Institute of Science and Technology (No. 2018JY01).

✉ Hong-Wei Jiao
jiaohongwei@126.com

You-Lin Shang
mathshang@sina.com

¹ Postdoctoral Research Base, Henan Institute of Science and Technology, Xinxiang 453003, Henan, China

² Postdoctoral Research Station of Control Science and Engineering, Henan University of Science and Technology, Luoyang 471023, Henan, China

1 Introduction

In this paper, we aim to solve the generalized linear fractional programming (GLFP) problem with the following general form:

$$(\text{GLFP}) : \begin{cases} \max g(x) = \sum_{j=1}^p \alpha_j \prod_{t=1}^{T_j} \left(\frac{c_{jt}^\top x + d_{jt}}{e_{jt}^\top x + f_{jt}} \right)^{\gamma_{jt}} \\ \text{s. t. } x \in D = \{x \in \mathbb{R}^N \mid Ax \leq b\}, \end{cases}$$

where p and T_j are all natural numbers, α_j and γ_{jt} ($j = 1, \dots, p, t = 1, \dots, T_j$) are all arbitrary real numbers, D is a nonempty bounded polyhedron set. $c_{jt}^\top x + d_{jt}$ and $e_{jt}^\top x + f_{jt}$ ($j = 1, \dots, p, t = 1, \dots, T_j$) are all affine functions defined on D such that $c_{jt}^\top x + d_{jt} > 0$ and $e_{jt}^\top x + f_{jt} > 0$ for all $x \in D$.

During the past several decades, intensive researches have been developed for solving special cases of GLFP problem. In part, this is because these problems exist very important applications in various disciplines, for example, the linear sum-of-ratios problems are encountered in various applications such as chance-constrained optimization, portfolio optimization, engineering design, risk management, multi-stage stochastic shipping, data envelopment analysis [1,2]. Another reason for the interest in these problems is that they generally possess multiple local optimal solutions which are not globally optimal.

Up to today, although some researchers have developed some deterministic algorithms for solving special cases of GLFP problem, to our knowledge, little work has been done for globally finding the optimal solution of the considered problem in this paper. For example, when $p = 1$, $e_{jt}^\top x + f_{jt} = 1$, and $\gamma_{jt} = 1$, GLFP problem is reduced to a linear multiplicative programming problem; when $p \geq 2$, $T_j \geq 2$, $e_{jt}^\top x + f_{jt} = 1$, and $\gamma_{jt} = 1$ ($t = 1, \dots, T_j$), GLFP problem is reduced to a generalized linear multiplicative programming problem; for these problems, several algorithms can be obtained in Refs. [3–7]. When $p \geq 2$, $T_j = 1$, and $\gamma_{jt} = 1$, GLFP problem is reduced to a linear sum-of-ratios problem; several algorithms can be found in Refs. [8–10]. When $e_{jt}^\top x + f_{jt} = 1$, and $c_{jt}^\top x + d_{jt}$ is reducible to a single variable, then GLFP problem is reduced to a generalized polynomial problem; for this problem, several deterministic algorithms have been presented in Refs. [11–16]. In addition, by introducing some new variables and constraints, Jiao et al. [17] have developed a feasible algorithm for globally solving the generalized linear fractional programming with nonlinear constraints. Recently, by utilizing equivalent transformation and direct linear relaxation, Liu et al. [18] and Zhang et al. [19] present two out-space branch-and-bound algorithms for the linear multiplicative and linear sum-of-ratios problem, respectively; by using range division and new linearizing technique, Shen et al. [20] propose a polynomial time approximation algorithm for solving the sum or product of linear fractional functions; based on reducing the original nonconvex problem as a standard geometric programming problem by utilizing simple transformation and contraction strategies, Shen et al. [21] present a practicable contraction approach for

solving the sum of the generalized polynomial ratios problem (P) with generalized polynomial constraints.

The purpose of this paper is to develop a unified algorithm to some variants of GLFP problem; our study is mainly motivated by Ref. [2]. In [2] and [22], the authors also point out that GLFP problem has been little studied in the literatures. In order to find the global optima of GLFP problem, we use an equivalent transformation and a two-level linear relaxation method to systematically convert GLFP problem into a series of linear programming relaxation problems. The solutions of these converted linear programming relaxation problems can be sufficiently close to the global optimal solution of GLFP problem by a successive refinement process. Based on the branch-and-bound framework, we design a branch-and-bound global optimization algorithm for GLFP problem. Numerical experimental results show that the proposed algorithm can solve all test examples for obtaining global optimal solutions within a given convergent tolerance.

This paper is organized as follows. In Sect. 2, a two-level linear relaxation method is constructed for deriving the linear programming relaxation problem of GLFP problem. A branch-and-bound algorithm in which the linear programming relaxation problems are embedded is established in Sect. 3, its global convergence is proved, and its computational complexity is derived. Numerical experimental results are reported in Sect. 4. Finally, some concluding remarks are described in Sect. 5.

2 Two-Level Linear Relaxation Method

The significant operation in the development of a branch-and-bound algorithm for GLFP problem is the construction of upper bounds for the initial problem, as well as for its refinement subproblems. The upper bounds for the optimal values of GLFP problem and its partitioned subproblems can be obtained by solving a series of linear programming relaxation problems. The proposed strategy for deriving these linear programming relaxation problems is to overestimate the objective function $g(x)$ with a linear function. The detailed deriving process is described as follows.

First of all, for each $i = 1, \dots, N$, since D is a nonempty bounded polyhedron set, by solving the following linear programming problems:

$$x_i^0 = \begin{cases} \min x_i \\ \text{s. t. } Ax \leq b, \\ c_{jt}^\top x + d_{jt} > 0, j = 1, \dots, p, t = 1, \dots, T_j, \\ e_{jt}^\top x + f_{jt} > 0, j = 1, \dots, p, t = 1, \dots, T_j, \end{cases}$$

and

$$\bar{x}_i^0 = \begin{cases} \max x_i \\ \text{s. t. } Ax \leq b, \\ c_{jt}^\top x + d_{jt} > 0, j = 1, \dots, p, t = 1, \dots, T_j, \\ e_{jt}^\top x + f_{jt} > 0, j = 1, \dots, p, t = 1, \dots, T_j, \end{cases}$$

we can obtain an initial rectangle X^0 , which is denoted by $X^0 = [\bar{x}^0, \bar{x}^0] = [\bar{x}_i^0, \bar{x}_i^0]_{N \times 1}$. It is obvious that the initial rectangle X^0 contains the feasible region of GLFP problem, and for all $x \in X^0$, we have that $c_{jt}^\top x + d_{jt} > 0$ and $e_{jt}^\top x + f_{jt} > 0$ hold.

Next, we will show how to construct the linear overestimating function of the objective function $g(x)$. For the function $g(x)$, without loss of generality, we assume $\alpha_j > 0$ ($j = 1, \dots, m$) and $\alpha_j < 0$ ($j = m+1, \dots, p$). Then the function $g(x)$ can be rewritten as follows:

$$g(x) = \sum_{j=1}^m \alpha_j \prod_{t=1}^{T_j} \left(\frac{c_{jt}^\top x + d_{jt}}{e_{jt}^\top x + f_{jt}} \right)^{\gamma_{jt}} + \sum_{j=m+1}^p \alpha_j \prod_{t=1}^{T_j} \left(\frac{c_{jt}^\top x + d_{jt}}{e_{jt}^\top x + f_{jt}} \right)^{\gamma_{jt}}.$$

For each term $\prod_{t=1}^{T_j} \left(\frac{c_{jt}^\top x + d_{jt}}{e_{jt}^\top x + f_{jt}} \right)^{\gamma_{jt}}$, since $c_{jt}^\top x + d_{jt} > 0$ and $e_{jt}^\top x + f_{jt} > 0$ for all $x \in X^0$, we have

$$\prod_{t=1}^{T_j} \left(\frac{c_{jt}^\top x + d_{jt}}{e_{jt}^\top x + f_{jt}} \right)^{\gamma_{jt}} = \exp \left(\sum_{t=1}^{T_j} \gamma_{jt} [\ln(c_{jt}^\top x + d_{jt}) - \ln(e_{jt}^\top x + f_{jt})] \right).$$

Then, we can derive an equivalent form of the function $g(x)$ as follows:

$$g(x) = \sum_{j=1}^m \alpha_j \exp \left(\sum_{t=1}^{T_j} \gamma_{jt} [\ln(c_{jt}^\top x + d_{jt}) - \ln(e_{jt}^\top x + f_{jt})] \right) + \sum_{j=m+1}^p \alpha_j \exp \left(\sum_{t=1}^{T_j} \gamma_{jt} [\ln(c_{jt}^\top x + d_{jt}) - \ln(e_{jt}^\top x + f_{jt})] \right).$$

To derive the linear upper bounding function (LUBF) of the objective function $g(x)$, we present a two-level linear relaxation method. In the first-level linear relaxation process, we will derive the linear lower bounding function (LLBF) and LUBF of the function $\sum_{t=1}^{T_j} \gamma_{jt} [\ln(c_{jt}^\top x + d_{jt}) - \ln(e_{jt}^\top x + f_{jt})]$, respectively. Then, in the second-level linear relaxation process, an LUBF of the objective function $g(x)$ will be constructed ultimately. The detailed two-level linear relaxation process is described as follows:

First-level linear relaxation:

For any $x \in X = \{x \in \mathbb{R}^N | \underline{x} \leq x \leq \bar{x}\} \subseteq X^0$, for each $j = 1, \dots, p, t = 1, \dots, T_j$, we let

$$\begin{aligned}\xi_{jt}^l &= \min_{x \in X} c_{jt}^\top x + d_{jt}, \quad \xi_{jt}^u = \max_{x \in X} c_{jt}^\top x + d_{jt}, \quad k_{jt}^1 = \frac{\ln \xi_{jt}^u - \ln \xi_{jt}^l}{\xi_{jt}^u - \xi_{jt}^l}, \\ \eta_{jt}^l &= \min_{x \in X} e_{jt}^\top x + f_{jt}, \quad \eta_{jt}^u = \max_{x \in X} e_{jt}^\top x + f_{jt}, \quad k_{jt}^2 = \frac{\ln \eta_{jt}^u - \ln \eta_{jt}^l}{\eta_{jt}^u - \eta_{jt}^l}, \\ \phi_{jt}(x) &= \ln(c_{jt}^\top x + d_{jt}) - \ln(e_{jt}^\top x + f_{jt}), \\ \phi_{jt}^u(x) &= [k_{jt}^1(c_{jt}^\top x + d_{jt}) - 1 - \ln k_{jt}^1] - [k_{jt}^2(e_{jt}^\top x + f_{jt} - \eta_{jt}^l) + \ln \eta_{jt}^l], \\ \phi_{jt}^l(x) &= [k_{jt}^1(c_{jt}^\top x + d_{jt} - \xi_{jt}^l) + \ln \xi_{jt}^l] - [k_{jt}^2(e_{jt}^\top x + f_{jt}) - 1 - \ln k_{jt}^2].\end{aligned}$$

Since $\ln(Y)$ is a concave and monotone increasing function of Y over $[Y^l, Y^u]$, its affine convex envelope is $k(Y - Y^l) + \ln Y^l$, where $k = \frac{\ln Y^u - \ln Y^l}{Y^u - Y^l}$. Moreover, since the tangential approximation function of $\ln(Y)$ is parallel with $k(Y - Y^l) + \ln Y^l$, thus the tangential approximation point will occur at $Y = \frac{1}{k}$, and the corresponding tangential approximation function is $kY - 1 - \ln k$. Hence, by geometric property of the function $\ln(Y)$, we have

$$k(Y - Y^l) + \ln Y^l \leq \ln(Y) \leq kY - 1 - \ln k.$$

Then, by the above discussions, we can derive the LLBF and LUBF of the functions $\ln(c_{jt}^\top x + d_{jt})$ and $\ln(e_{jt}^\top x + f_{jt})$, respectively. So that we can get that

$$\begin{aligned}k_{jt}^1(c_{jt}^\top x + d_{jt} - \xi_{jt}^l) + \ln \xi_{jt}^l &\leq \ln(c_{jt}^\top x + d_{jt}) \leq k_{jt}^1(c_{jt}^\top x + d_{jt}) - 1 - \ln k_{jt}^1, \\ k_{jt}^2(e_{jt}^\top x + f_{jt} - \eta_{jt}^l) + \ln \eta_{jt}^l &\leq \ln(e_{jt}^\top x + f_{jt}) \leq k_{jt}^2(e_{jt}^\top x + f_{jt}) - 1 - \ln k_{jt}^2.\end{aligned}$$

Then, by the above inequalities, it follows that

$$\begin{aligned}&\ln(c_{jt}^\top x + d_{jt}) - \ln(e_{jt}^\top x + f_{jt}) \\ &\leq k_{jt}^1(c_{jt}^\top x + d_{jt}) - 1 - \ln k_{jt}^1 - [k_{jt}^2(e_{jt}^\top x + f_{jt} - \eta_{jt}^l) + \ln \eta_{jt}^l],\end{aligned}$$

and

$$\begin{aligned}&\ln(c_{jt}^\top x + d_{jt}) - \ln(e_{jt}^\top x + f_{jt}) \\ &\geq k_{jt}^1(c_{jt}^\top x + d_{jt} - \xi_{jt}^l) + \ln \xi_{jt}^l - [k_{jt}^2(e_{jt}^\top x + f_{jt}) - 1 - \ln k_{jt}^2].\end{aligned}$$

Therefore, by the above inequalities, we have

$$\phi_{jt}^l(x) \leq \phi_{jt}(x) \leq \phi_{jt}^u(x). \quad (1)$$

For each $j = 1, \dots, p$, we let

$$\begin{aligned} Z_j(x) &= \sum_{t=1}^{T_j} \gamma_{jt} \phi_{jt}(x) = \sum_{t=1}^{T_j} \gamma_{jt} \left[\ln(c_{jt}^\top x + d_{jt}) - \ln(e_{jt}^\top x + f_{jt}) \right], \\ Z_j^l(x) &= \sum_{t=1, \gamma_{jt} > 0}^{T_j} \gamma_{jt} \phi_{jt}^l(x) + \sum_{t=1, \gamma_{jt} < 0}^{T_j} \gamma_{jt} \phi_{jt}^u(x), \\ Z_j^u(x) &= \sum_{t=1, \gamma_{jt} > 0}^{T_j} \gamma_{jt} \phi_{jt}^u(x) + \sum_{t=1, \gamma_{jt} < 0}^{T_j} \gamma_{jt} \phi_{jt}^l(x), \\ U(x) &= \sum_{j=1}^m \alpha_j \exp(Z_j^u(x)) + \sum_{j=m+1}^p \alpha_j \exp(Z_j^l(x)). \end{aligned}$$

Thus, by (1) we can obtain that

$$Z_j^l(x) \leq Z_j(x) \leq Z_j^u(x).$$

By the above inequality, we can get that

$$g(x) = \sum_{j=1}^p \alpha_j \exp(Z_j(x)) \leq \sum_{j=1}^m \alpha_j \exp(Z_j^u(x)) + \sum_{j=m+1}^p \alpha_j \exp(Z_j^l(x)) = U(x). \quad (2)$$

Hence, we can obtain the first-level relaxation overestimating function $U(x)$ of the function $g(x)$ over X as follows:

$$U(x) = \sum_{j=1}^m \alpha_j \exp(Z_j^u(x)) + \sum_{j=m+1}^p \alpha_j \exp(Z_j^l(x)). \quad (3)$$

Second-level linear relaxation:

Using the similar method as the first-level linear relaxation, we can derive the LUBF of the function $U(x)$. Since $\exp(Z)$ is a convex and monotone increasing function of Z over $[Z^l, Z^u]$, its affine concave envelope is $B(Z - Z^l) + \exp(Z^l)$, where $B = \frac{\exp(Z^u) - \exp(Z^l)}{Z^u - Z^l}$. Moreover, since the tangential approximation function of $\exp(Z)$ is parallel with $B(Z - Z^l) + \exp(Z^l)$, thus the tangential approximation point will occur at $\ln B$, and the corresponding tangential approximation function is $B(1 + Z - \ln B)$. Hence, by geometric property of the function $\exp(Z)$, we have

$$B(1 + Z - \ln B) \leq \exp(Z) \leq B(Z - Z^l) + \exp(Z^l).$$

For any $x \in X = \{x \in \mathbb{R}^N \mid \underline{x} \leq x \leq \bar{x}\} \subseteq X^0$, we let

$$\begin{aligned} Z_j^{ll} &= \min_{x \in X} Z_j^l(x), \quad Z_j^{lu} = \max_{x \in X} Z_j^l(x), \quad K_j^1 = \frac{\exp(Z_j^{lu}) - \exp(Z_j^{ll})}{Z_j^{lu} - Z_j^{ll}}, \\ Z_j^{ul} &= \min_{x \in X} Z_j^u(x), \quad Z_j^{uu} = \max_{x \in X} Z_j^u(x), \quad K_j^2 = \frac{\exp(Z_j^{uu}) - \exp(Z_j^{ul})}{Z_j^{uu} - Z_j^{ul}}. \end{aligned}$$

By the above discussion, we can derive the LUBF of the function $\exp(Z_j^u(x))$ and the LLBF of the function $\exp(Z_j^l(x))$, respectively. So that we can get that

$$K_j^2(Z_j^u(x) - Z_j^{ul}) + \exp(Z_j^{ul}) \geq \exp(Z_j^u(x))$$

and

$$\exp(Z_j^l(x)) \geq K_j^1[1 + Z_j^l(x) - \ln K_j^1].$$

Then, according to the above discussions, for each $j = 1, \dots, m$, substitute the term $\exp(Z_j^u(x))$ by $K_j^2(Z_j^u(x) - Z_j^{ul}) + \exp(Z_j^{ul})$ in (3); for each $j = m+1, \dots, p$, substitute the term $\exp(Z_j^l(x))$ by $K_j^1[1 + Z_j^l(x) - \ln K_j^1]$ in (3), respectively. We may derive the LUBF of the function $U(x)$ as follows:

$$g^u(x) = \sum_{j=1}^m \alpha_j \{K_j^2(Z_j^u(x) - Z_j^{ul}) + \exp(Z_j^{ul})\} + \sum_{j=m+1}^p \alpha_j \{K_j^1[1 + Z_j^l(x) - \ln K_j^1]\}.$$

Obviously, we have

$$U(x) \leq g^u(x). \quad (4)$$

Combining (2) and (4), we can follow that

$$g(x) \leq U(x) \leq g^u(x). \quad (5)$$

Consequently, we can construct the corresponding approximation linear programming relaxation problem (LPRP) of GLFP problem over X as follows:

$$(\text{LPRP}) : \begin{cases} \max g^u(x) \\ \text{s.t. } Ax \leq b, \quad x \in X, \end{cases}$$

where

$$g^u(x) = \sum_{j=1}^m \alpha_j \{K_j^2(Z_j^u(x) - Z_j^{ul}) + \exp(Z_j^{ul})\} + \sum_{j=m+1}^p \alpha_j \{K_j^1[1 + Z_j^l(x) - \ln K_j^1]\}.$$

From the deriving process of the LPRP, for any $X \subseteq X^0$, the LPRP can provide a reliable upper bound for the global optimal value of GLFP problem in the branch-and-bound algorithm.

3 Algorithm and Its Global Convergence

In this section, based on the above linear programming relaxation problem, we propose a branch-and-bound algorithm for globally solving GLFP problem, and main operations of the proposed algorithm are given as follows.

First of all, we select a rectangle bisection operation of maximum edge, which is sufficient to ensure the global convergence of the proposed algorithm. The detailed branching method is described as follows. Assume that $X^k = [\underline{x}^k, \bar{x}^k] \subseteq X^0$ is the selected rectangle, where the rule of selecting the rectangle is given in Step 5 of the proposed algorithm, let $\sigma \in \arg \max \{\bar{x}_i^k - \underline{x}_i^k : i = 1, 2, \dots, N\}$, partition the interval $[\underline{x}_\sigma^k, \bar{x}_\sigma^k]$ into two subintervals $[\underline{x}_\sigma^k, \frac{\underline{x}_\sigma^k + \bar{x}_\sigma^k}{2}]$ and $[\frac{\underline{x}_\sigma^k + \bar{x}_\sigma^k}{2}, \bar{x}_\sigma^k]$, so that we can partition X^k into two sub-rectangles.

Secondly, by solving a sequence of LPRP of GLFP problem, we may update the upper bound of the optimal value of GLFP problem. At the same time, since the optimal solutions of these linear programming relaxation problems are always feasible to GLFP problem, by calculating the objective functional values of these feasible points, we may update the lower bound of the optimal value of GLFP problem.

Thirdly, by detecting the feasibility of midpoints of each investigated sub-rectangle and calculating the objective functional values of these feasible points, we can also update the lower bound of the optimal value of GLFP problem.

Fourthly, similar to Ref. [23], for each investigated sub-rectangle, based on the linear relaxation of the objective function and the currently known lower bound, we may construct optimality-based interval reducing tactics to reduce the number of iterations needed to prove global optimality; the proposed reducing tactics are given as follows.

Without loss of generality, we rewrite the objective function of the LPRP over X as the following form:

$$g^u(x) = \sum_{i=1}^N \beta_i x_i + \delta.$$

Denote by LB_k the currently known lower bound of the global optimal value of GLFP problem, which is obtained after k iterations, and set

$$\begin{aligned} \text{RUB} &= \sum_{i=1, \beta_i > 0}^N \beta_i \bar{x}_i + \sum_{i=1, \beta_i < 0}^N \beta_i \underline{x}_i + \delta; \\ \omega_q &= \frac{LB_k - \text{RUB} + \max\{\beta_q \bar{x}_q, \beta_q \underline{x}_q\}}{\beta_q}, \quad q \in \{1, \dots, N\}; \\ \bar{X}_i &= \begin{cases} X_i, & i \neq q, i = 1, \dots, N, \\ [\underline{x}_q, \omega_q) \cap X_q, & i = q; \end{cases} \\ \bar{\bar{X}}_i &= \begin{cases} X_i, & i \neq q, i = 1, \dots, N, \\ (\omega_q, \bar{x}_q] \cap X_q, & i = q. \end{cases} \end{aligned}$$

Theorem 1 For any sub-rectangle $X \subseteq X^0$, we have the following conclusions:

- (i) If $\text{RUB} < LB_k$, then the investigated sub-rectangle X exists no global optimal solution of GLFP problem.

- (ii) If $\text{RUB} \geq \text{LB}_k$, then: for any $q \in \{1, 2, \dots, N\}$,
 if $\beta_q > 0$, then the sub-rectangle $\bar{X} = (\bar{X}_i)_{N \times 1}$ exists no global optimal solution of GLFP problem;
 if $\beta_q < 0$, then the sub-rectangle $\bar{\bar{X}} = (\bar{\bar{X}}_i)_{N \times 1}$ exists no global optimal solution of GLFP problem.

Proof (i) If $\text{RUB} < \text{LB}_k$, then for any $x \in X$, $g^u(x) \leq \text{RUB} < \text{LB}_k$, i.e., for any $x \in X$, we have that

$$g(x) \leq g^u(x) \leq \sum_{i=1, \beta_i > 0}^N \beta_i \bar{x}_i + \sum_{i=1, \beta_i < 0}^N \beta_i \underline{x}_i + \delta = \text{RUB} < \text{LB}_k.$$

Therefore, the investigated sub-rectangle X exists no global optimal solution of GLFP problem.

(ii) If $\text{RUB} \geq \text{LB}_k$, then, for each $q \in \{1, \dots, n\}$, if $\beta_q > 0$, for any $x \in \bar{X}$, we have $x_q < \omega_q$, i.e.,

$$x_q < \frac{\text{LB}_k - \text{RUB} + \max\{\beta_q \bar{x}_q, \beta_q \underline{x}_q\}}{\beta_q}.$$

Therefore, for any $x \in \bar{X}$, we can get that

$$\begin{aligned} g(x) &\leq g^u(x) \\ &\leq \sum_{i=1, i \neq q}^N \max\{\beta_i \bar{x}_i, \beta_i \underline{x}_i\} + \delta + \beta_q x_q \\ &< \sum_{i=1, i \neq q}^N \max\{\beta_i \bar{x}_i, \beta_i \underline{x}_i\} + \delta + (\text{LB}_k - \text{RUB} + \max\{\beta_q \bar{x}_q, \beta_q \underline{x}_q\}) \\ &= \text{RUB} + \text{LB}_k - \text{RUB} \\ &= \text{LB}_k. \end{aligned}$$

Thus, for any $x \in \bar{X}$, we have that

$$g(x) \leq g^u(x) < \text{LB}_k.$$

Hence, the sub-rectangle $\bar{X} = (\bar{X}_i)_{N \times 1}$ exists no global optimal solution of GLFP problem.

Similarly, we can also prove that, if $\beta_q < 0$, then the sub-rectangle $\bar{\bar{X}} = (\bar{\bar{X}}_i)_{N \times 1}$ exists no global optimal solution of GLFP problem.

By utilizing the conclusions of Theorem 1, we can give some reducing tactics to delete a part of the investigated sub-rectangle in which there does not exist the global optimal solution of GLFP problem, so that we can enhance the convergent speed of the proposed branch-and-bound algorithm.

3.1 Branch-and-bound algorithm

Denote by $UB(X^k)$ and $x^k = x(X^k)$ the optimal value and the optimal solution of the LPRP over X^k , respectively. The proposed branch-and-bound algorithm is described as follows.

Algorithm 1: Branch-and-bound algorithm

Step 1 (Initializing) Given the initial convergent error $\varepsilon > 0$, the initial set of all active node $\Omega_0 = \{X^0\}$ and the initial set of all deleted node $\Lambda = \emptyset$.

Solve the LPRP over X^0 to get its optimal solution x^0 and optimal value $UB(X^0)$, respectively. Denote by $UB_0 = UB(X^0)$ the initial upper bound, and denote by $LB_0 = g(x^0)$ the initial lower bound.

If $UB_0 - LB_0 \leq \varepsilon$, Algorithm 1 terminates with x^0 as the global ε -optimal solution of GLFP problem. Otherwise, let $k = 0$, and continue to the next step.

Step 2 (Branching) By using the proposed branching technique, we divide the rectangle X^k into two new sub-rectangles $X^{k,1}$ and $X^{k,2}$.

Step 3 (Interval reducing) Use the proposed reducing tactics to compress each sub-rectangle $X^{k,t}$, where $t = 1, 2$, and denote by Λ the set of all deleted sub-rectangles. Meanwhile, the remaining partition set is now $\Omega_k = \{X | X \in \Omega_{k-1} \cup \{X^{k,1}, X^{k,2}\}, X \notin \Lambda\}$.

Step 4 (Bounding) For each sub-rectangle $X^{k,t}$, where $t = 1, 2$, solve the LPRP to get its optimal solution $x^{k,t}$ and optimal value $UB(X^{k,t})$, if $UB(X^{k,t}) < LB_k$, set $\Omega_k = \Omega_k \setminus X^{k,t}$, and let $LB_k = \max\{LB_k, g(x^{k,t})\}$.

If the midpoint x^{mid} of $X^{k,t}$ ($t \in \{1, 2\}$) is feasible to GLFP problem, let $LB_k = \max\{LB_k, g(x^{\text{mid}})\}$.

Denote by x^k the best feasible solution such that $LB_k = g(x^k)$, and denote by the upper bound $UB_k = \max\{UB(X) | X \in \Omega_k\}$.

Step 5 (Terminating) Set $\Omega_{k+1} = \{X | UB(X) - LB_k > \varepsilon, X \in \Omega_k\}$. If $\Omega_{k+1} = \emptyset$, then Algorithm 1 terminates: select the rectangle X^{k+1} such that $X^{k+1} = \arg \max_{X \in \Omega_{k+1}} UB(X)$, and let $k := k + 1$, and return to Step 2.

3.2 Convergence analysis

In this subsection, the global convergence of Algorithm 1 is given by following Theorems 2 and 3.

First of all, following Theorem 2 ensures that the linear function $g^u(x)$ will infinitely approximate the objective function $g(x)$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$.

Theorem 2 For any $x \in X = \{x \in \mathbb{R}^N | \underline{x} \leq x \leq \bar{x}\} \subseteq X^0$, the difference of the functions $g^u(x)$ and $g(x)$ satisfies that $g^u(x) - g(x) \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$.

Proof For any $x \in X$, let

$$\Delta = g^u(x) - g(x) = [g^u(x) - U(x)] + [U(x) - g(x)] = \Delta^1 + \Delta^2. \quad (6)$$

From (5), we know that $\Delta^1 \geq 0$ and $\Delta^2 \geq 0$. Therefore, we only need to prove that $\Delta^1 \rightarrow 0$ and $\Delta^2 \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$.

First, we consider the difference $\Delta^1 = g^u(x) - U(x)$. Let

$$\begin{aligned}\Delta_{1j}^1 &= K_j^2(Z_j^u(x) - Z_j^{ul}) + \exp(Z_j^{ul}) - \exp(Z_j^u(x)), \\ \Delta_{2j}^1 &= \exp(Z_j^l(x)) - K_j^1[1 + Z_j^l(x) - \ln K_j^1],\end{aligned}$$

we have

$$\begin{aligned}\Delta^1 &= g^u(x) - U(x) \\ &= \sum_{j=1}^m \alpha_j \{K_j^2(Z_j^u(x) - Z_j^{ul}) + \exp(Z_j^{ul})\} + \sum_{j=m+1}^p \alpha_j \{K_j^1[1 + Z_j^l(x) - \ln K_j^1]\} \\ &\quad - [\sum_{j=1}^m \alpha_j \exp(Z_j^u(x)) + \sum_{j=m+1}^p \alpha_j \exp(Z_j^l(x))] \\ &= \sum_{j=1}^m \alpha_j [K_j^2(Z_j^u(x) - Z_j^{ul}) + \exp(Z_j^{ul}) - \exp(Z_j^u(x))] \\ &\quad - \sum_{j=m+1}^p \alpha_j [\exp(Z_j^l(x)) - K_j^1[1 + Z_j^l(x) - \ln K_j^1]] \\ &= \sum_{j=1}^m \alpha_j \Delta_{1j}^1 - \sum_{j=m+1}^p \alpha_j \Delta_{2j}^1.\end{aligned}$$

Since $\Delta_{1j}^1 = K_j^2(Z_j^u(x) - Z_j^{ul}) + \exp(Z_j^{ul}) - \exp(Z_j^u(x))$ is a concave function about $Z_j^u(x)$, it can attain the maximum value $\Delta_{1j}^{1,\max}$ at the point $\ln K_j^2$. Let

$$T_{1j} = \frac{\exp(Z_j^{uu} - Z_j^{ul}) - 1}{Z_j^{uu} - Z_j^{ul}},$$

then through computing we can derive the following form:

$$\Delta_{1j}^{1,\max} = \exp(Z_j^{ul})(1 - T_{1j} + T_{1j} \ln T_{1j}).$$

Since $|Z_j^{uu} - Z_j^{ul}| \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$, $T_{1j} \rightarrow 1$ as $|Z_j^{uu} - Z_j^{ul}| \rightarrow 0$, and $\Delta_{1j}^{1,\max} \rightarrow 0$ as $T_{1j} \rightarrow 1$, obviously, we have

$$\Delta_{1j}^{1,\max} \rightarrow 0 \text{ as } \|\bar{x} - \underline{x}\| \rightarrow 0. \quad (7)$$

Similarly, we can prove that the maximum value $\Delta_{2j}^{1,\max}$ of the function $\Delta_{2j}^1 = \exp(Z_j^l(x)) - K_j^1[1 + Z_j^l(x) - \ln K_j^1]$ satisfies

$$\Delta_{2j}^{1,\max} \rightarrow 0 \text{ as } \|\bar{x} - \underline{x}\| \rightarrow 0. \quad (8)$$

Therefore, by (7) and (8) we have

$$\Delta^1 = g^u(x) - U(x) = \sum_{j=1}^m \alpha_j \Delta_{1j}^1 - \sum_{j=m+1}^p \alpha_j \Delta_{2j}^1 \rightarrow 0 \text{ as } \|\bar{x} - \underline{x}\| \rightarrow 0. \quad (9)$$

Second, we consider the difference $\Delta^2 = U(x) - g(x)$. Let

$$\Delta_{1j}^2 = \exp(Z_j^u(x)) - \exp(Z_j(x))$$

and

$$\Delta_{2j}^2 = \exp(Z_j(x)) - \exp(Z_j^l(x)),$$

we have

$$\begin{aligned} \Delta^2 &= U(x) - g(x) \\ &= \sum_{j=1}^m \alpha_j \exp(Z_j^u(x)) - \sum_{j=1}^m \alpha_j \exp\left(\sum_{t=1}^{T_j} \gamma_{jt} [\ln(c_{jt}^\top x + d_{jt}) - \ln(e_{jt}^\top x + f_{jt})]\right) \\ &\quad + \sum_{j=m+1}^p \alpha_j \exp(Z_j^l(x)) \\ &\quad - \sum_{j=m+1}^p \alpha_j \exp\left(\sum_{t=1}^{T_j} \gamma_{jt} [\ln(c_{jt}^\top x + d_{jt}) - \ln(e_{jt}^\top x + f_{jt})]\right) \\ &= \sum_{j=1}^m \alpha_j [\exp(Z_j^u(x)) - \exp(\sum_{t=1}^{T_j} \gamma_{jt} \phi_{jt}(x))] \\ &\quad + \sum_{j=m+1}^p \alpha_j [\exp(Z_j^l(x)) - \exp(\sum_{t=1}^{T_j} \gamma_{jt} \phi_{jt}(x))] \\ &= \sum_{j=1}^m \alpha_j [\exp(Z_j^u(x)) - \exp(Z_j(x))] + \sum_{j=m+1}^p \alpha_j [\exp(Z_j^l(x)) - \exp(Z_j(x))] \\ &= \sum_{j=1}^m \alpha_j \Delta_{1j}^2 - \sum_{j=m+1}^p \alpha_j \Delta_{2j}^2. \end{aligned}$$

Since

$$\Delta_{1j}^2 = \exp(Z_j^u(x)) - \exp(Z_j(x)) \leq \|Z_j^u(x) - Z_j(x)\| \sup_{\theta_j \in L(Z_j^u(x), Z_j(x))} \exp(\theta_j), \quad (10)$$

where

$$L(Z_j^u(x), Z_j(x)) = \beta Z_j^u(x) + (1 - \beta) Z_j(x), \quad \beta \in [0, 1].$$

For each $j \in \{1, \dots, T\}$, let

$$\begin{aligned}\Delta_{1jt.1}^2 &= [k_{jt}^1(c_{jt}^\top x + d_{jt}) - 1 - \ln k_{jt}^1] - \ln(c_{jt}^\top x + d_{jt}), \\ \Delta_{1jt.2}^2 &= \ln(e_{jt}^\top x + f_{jt}) - [k_{jt}^2(e_{jt}^\top x + f_{jt} - \eta_{jt}^l) + \ln \eta_{jt}^l], \\ \Delta_{1jt.3}^2 &= \ln(c_{jt}^\top x + d_{jt}) - [k_{jt}^1(c_{jt}^\top x + d_{jt} - \xi_{jt}^l) + \ln \xi_{jt}^l], \\ \Delta_{1jt.4}^2 &= [k_{jt}^2(e_{jt}^\top x + f_{jt}) - 1 - \ln k_{jt}^2] - \ln(e_{jt}^\top x + f_{jt}),\end{aligned}$$

then it follows that

$$\begin{aligned}Z_j^u(x) - Z_j(x) &= \sum_{t=1, \gamma_{jt} > 0}^{T_j} \gamma_{jt} \phi_{jt}^u(x) + \sum_{t=1, \gamma_{jt} < 0}^{T_j} \gamma_{jt} \phi_{jt}^l(x) - \sum_{t=1}^{T_j} \gamma_{jt} \phi_{jt}(x) \\ &= \sum_{t=1, \gamma_{jt} > 0}^{T_j} \gamma_{jt} [\phi_{jt}^u(x) - \phi_{jt}(x)] + \sum_{t=1, \gamma_{jt} < 0}^{T_j} \gamma_{jt} [\phi_{jt}^l(x) - \phi_{jt}(x)] \\ &= \sum_{t=1, \gamma_{jt} > 0}^{T_j} \gamma_{jt} \left\{ [k_{jt}^1(c_{jt}^\top x + d_{jt}) - 1 - \ln k_{jt}^1] - \ln(c_{jt}^\top x + d_{jt}) \right\} \\ &\quad + \sum_{t=1, \gamma_{jt} > 0}^{T_j} \gamma_{jt} \left\{ \ln(e_{jt}^\top x + f_{jt}) - [k_{jt}^2(e_{jt}^\top x + f_{jt} - \eta_{jt}^l) + \ln \eta_{jt}^l] \right\} \\ &\quad - \sum_{t=1, \gamma_{jt} < 0}^{T_j} \gamma_{jt} \left\{ \ln(c_{jt}^\top x + d_{jt}) - [k_{jt}^1(c_{jt}^\top x + d_{jt} - \xi_{jt}^l) + \ln \xi_{jt}^l] \right\} \\ &\quad - \sum_{t=1, \gamma_{jt} < 0}^{T_j} \gamma_{jt} \left\{ [k_{jt}^2(e_{jt}^\top x + f_{jt}) - 1 - \ln k_{jt}^2] - \ln(e_{jt}^\top x + f_{jt}) \right\} \\ &= \sum_{t=1, \gamma_{jt} > 0}^{T_j} \gamma_{jt} (\Delta_{1jt.1}^2 + \Delta_{1jt.2}^2) - \sum_{t=1, \gamma_{jt} < 0}^{T_j} \gamma_{jt} (\Delta_{1jt.3}^2 + \Delta_{1jt.4}^2).\end{aligned}$$

Since the function $\Delta_{1jt.1}^2$ is a convex function about $(c_{jt}^\top x + d_{jt})$, it can obtain the maximum value $\Delta_{1jt.1}^{2, \max}$ at the point ξ_{jt}^l or ξ_{jt}^u . Then through computing we get

$$\Delta_{1jt.1}^{2, \max} = \frac{\ln(\frac{\xi_{jt}^u}{\xi_{jt}^l})}{\frac{\xi_{jt}^u}{\xi_{jt}^l} - 1} - 1 - \ln \frac{\ln(\frac{\xi_{jt}^u}{\xi_{jt}^l})}{\frac{\xi_{jt}^u}{\xi_{jt}^l} - 1}.$$

Since $\frac{\xi_{jt}^u}{\xi_{jt}^l} \rightarrow 1$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$, then we have $\Delta_{1jt.1}^{2, \max} \rightarrow 0$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$. Therefore, it follows that

$$\Delta_{1jt.1}^2 \rightarrow 0 \text{ as } \|\bar{x} - \underline{x}\| \rightarrow 0. \quad (11)$$

Similarly, we can prove that

$$\Delta_{1jt,2}^2 \rightarrow 0, \Delta_{1jt,3}^2 \rightarrow 0 \text{ and } \Delta_{1jt,4}^2 \rightarrow 0 \text{ as } \|\bar{x} - \underline{x}\| \rightarrow 0. \quad (12)$$

By (11) and (12), it follows that, as $\|\bar{x} - \underline{x}\| \rightarrow 0$,

$$\begin{aligned} Z_j^u(x) - Z_j(x) &= \sum_{t=1, \gamma_{jt} > 0}^{T_j} \gamma_{jt} (\Delta_{1jt,1}^2 + \Delta_{1jt,2}^2) \\ &\quad - \sum_{t=1, \gamma_{jt} < 0}^{T_j} \gamma_{jt} (\Delta_{1jt,3}^2 + \Delta_{1jt,4}^2) \rightarrow 0. \end{aligned} \quad (13)$$

Since $\exp(\theta_j)$ is a continuous and bounded function about θ_j , then there exists some $\bar{M} > 0$ such that $\exp(\theta_j) \leq \bar{M}$. Therefore, by (10) and (13), it follows that

$$\Delta_{1j}^2 \leq \bar{M} \|Z_j^u(x) - Z_j(x)\| \rightarrow 0 \text{ as } \|\bar{x} - \underline{x}\| \rightarrow 0.$$

Therefore, we have

$$\Delta_{1j}^2 \rightarrow 0 \text{ as } \|\bar{x} - \underline{x}\| \rightarrow 0. \quad (14)$$

Using the similar method as the former, we can also prove that

$$\Delta_{2j}^2 \rightarrow 0 \text{ as } \|\bar{x} - \underline{x}\| \rightarrow 0. \quad (15)$$

By (14) and (15), it follows that

$$\Delta^2 = \sum_{j=1}^m \alpha_j \Delta_{1j}^2 - \sum_{j=m+1}^p \alpha_j \Delta_{2j}^2 \rightarrow 0 \text{ as } \|\bar{x} - \underline{x}\| \rightarrow 0. \quad (16)$$

By (6), (9) and (16), it follows that

$$\Delta = g^u(x) - g(x) = \Delta^1 + \Delta^2 \rightarrow 0 \text{ as } \|\bar{x} - \underline{x}\| \rightarrow 0.$$

By the above discussions it is obvious that the conclusion is followed.

From Theorem 2, it follows that $g^u(x)$ will infinitely approximate $g(x)$ as $\|\bar{x} - \underline{x}\| \rightarrow 0$. Next, we have the following convergent conclusions.

Theorem 3 *If Algorithm 1 is finite, then it terminates after k iterations, and x^k is an ε -global optimal solution of GLFP problem. Otherwise, Algorithm 1 will generate an infinite sequence $\{x^k\}$ such that any of its accumulation point will be the global solution of GLFP problem.*

Proof If Algorithm 1 is finite, then it terminates after k iterations; by the termination condition of Algorithm 1, it follows that

$$\text{UB}_k - \text{LB}_k \leq \varepsilon.$$

By Step 4, it follows that

$$\text{LB}_k = g(x^k).$$

Denote by v the global optimal value of GLFP problem, by the structure of Algorithm 1 we have

$$v \leq \text{UB}_k \text{ and } \text{LB}_k \leq v.$$

Combining the above inequalities and equivalence, it follows that

$$v - \varepsilon \leq \text{UB}_k - \varepsilon \leq \text{LB}_k = g(x^k) = \text{LB}_k \leq v.$$

Therefore, x^k is an ε -global optimal solution of GLFP problem.

If Algorithm 1 does not terminate after k iterations, then by the structure of Algorithm 1, we can obtain a nondecreasing sequence $\{\text{UB}_k\}$, which has the lower bound $\max_{x \in \Pi} g(x)$, where Π is the known feasible point set. Therefore, we have $\text{UB} = \lim_{k \rightarrow \infty} \text{UB}_k \geq \max_{x \in \Pi} g(x)$.

Since $\{x^k\} \in X^0$, where X^0 is a bounded close set, we can get that it must exist a convergent subsequence $\{x^\rho\} \subseteq \{x^k\}$ such that $\lim_{\rho \rightarrow \infty} x^\rho = x^*$. Obviously, we have $x^* \in X^0$. By the structure of Algorithm 1, there must exist a subsequence $\{X^\rho\} \subseteq \{X^k\}$, where $X^k \in \Omega_k$ with $x^\rho \in X^\rho$, $\text{UB}_\rho = \text{UB}(X^\rho) = g^u(x^\rho)$, $\lim_{\rho \rightarrow \infty} x^\rho = x^*$ and $\lim_{\rho \rightarrow \infty} X^\rho = x^*$. By Theorem 2 and the continuity of $g(x)$, it follows that

$$\lim_{\rho \rightarrow \infty} \text{UB}_\rho = \lim_{\rho \rightarrow \infty} \text{UB}(X^\rho) = \lim_{\rho \rightarrow \infty} g^u(x^\rho) = \lim_{\rho \rightarrow \infty} g(x^\rho) = g(x^*).$$

Since x^ρ is always feasible to GLFP problem, it is obvious that x^* is also a feasible solution for GLFP problem. Therefore, x^* is the global optimal solution for GLFP problem. The proof is completed.

3.3 Computational complexity of Algorithm 1

In this subsection, to get the computational complexity of Algorithm 1, first of all, we give the gap between $g^u(x)$ and $g(x)$ with relationship of $(\bar{x} - \underline{x})$ as follows.

3.3.1 The gap between $g^u(x)$ and $g(x)$ with relationship of $(\bar{x} - \underline{x})$

From above Theorem 2, we have that

$$g^u(x) - g(x) = g^u(x) - U(x) + U(x) - g(x).$$

By the proving process of Theorem 2, we have

$$g^u(x) - U(x) = \sum_{j=1}^m \alpha_j \left[K_j^2(Z_j^u(x) - Z_j^{ul}) + \exp(Z_j^{ul}) - \exp(Z_j^u(x)) \right] \\ - \sum_{j=m+1}^p \alpha_j \left[\exp(Z_j^l(x)) - K_j^1[1 + Z_j^l(x) - \ln K_j^1] \right].$$

For any $x \in X^0$, for each $j = 1, \dots, p$, $t = 1, \dots, T_j$, we define

$$\hat{\xi}_{jt}^l = \min_{x \in X^0} c_{jt}^\top x + d_{jt}, \quad \hat{\eta}_{jt}^l = \min_{x \in X^0} e_{jt}^\top x + f_{jt}, \quad \hat{Z}_j^{uu} = \max_{x \in X^0} Z_j^u(x).$$

Obviously, we have that

$$k_{jt}^1 \leq \frac{1}{\hat{\xi}_{jt}^l} \quad \text{and} \quad k_{jt}^2 \leq \frac{1}{\hat{\eta}_{jt}^l}, \quad j = 1, \dots, p, t = 1, \dots, T_j;$$

and for any $x \in X \subseteq X^0$, for any $j = 1, \dots, p$, we have that

$$\hat{Z}_j^{uu} = \max_{x \in X^0} Z_j^u(x) \geq \max_{x \in X} Z_j^u(x) = Z_j^{uu}.$$

From the proving process of Theorem 2, for any $x \in X \subseteq X^0$, we have that

$$\begin{aligned} & K_j^2(Z_j^u(x) - Z_j^{ul}) + \exp(Z_j^{ul}) - \exp(Z_j^u(x)) \\ & \leq K_j^2(Z_j^{uu} - Z_j^{ul}) + \exp(Z_j^{ul}) - \exp(Z_j^u(x)) \\ & = \exp(Z_j^{uu}) - \exp(Z_j^{ul}) + \exp(Z_j^{ul}) - \exp(Z_j^u(x)) \\ & = \exp(Z_j^{uu}) - \exp(Z_j^u(x)) \\ & \leq \exp(Z_j^{uu})(Z_j^{uu} - Z_j^{ul}) \\ & \leq \exp(\max_{x \in X^0} Z_j^u(x))(Z_j^{uu} - Z_j^{ul}) \\ & = \exp(\hat{Z}_j^{uu}) \times \left[\sum_{t=1}^{T_j} |\gamma_{jt}| k_{jt}^1 \sum_{i=1}^N |c_{jti}| (\bar{x}_i - \underline{x}_i) + \sum_{t=1}^{T_j} |\gamma_{jt}| k_{jt}^2 \sum_{i=1}^N |e_{jti}| (\bar{x}_i - \underline{x}_i) \right] \\ & = \exp(\hat{Z}_j^{uu}) \times \left[\sum_{i=1}^N \left(\sum_{t=1}^{T_j} k_{jt}^1 |\gamma_{jt} c_{jti}| + \sum_{t=1}^{T_j} k_{jt}^2 |\gamma_{jt} e_{jti}| \right) (\bar{x}_i - \underline{x}_i) \right] \\ & = \exp(\hat{Z}_j^{uu}) \times \left[\sum_{i=1}^N \left(\sum_{t=1}^{T_j} (k_{jt}^1 |\gamma_{jt} c_{jti}| + k_{jt}^2 |\gamma_{jt} e_{jti}|) \right) (\bar{x}_i - \underline{x}_i) \right] \\ & \leq \exp(\hat{Z}_j^{uu}) \times \left[\sum_{i=1}^N \left(\sum_{t=1}^{T_j} \left(\frac{1}{\hat{\xi}_{jt}^l} |\gamma_{jt} c_{jti}| + \frac{1}{\hat{\eta}_{jt}^l} |\gamma_{jt} e_{jti}| \right) \right) (\bar{x}_i - \underline{x}_i) \right] \end{aligned}$$

and

$$\begin{aligned}
& \exp(Z_j^l(x)) - K_j^1[1 + Z_j^l(x) - \ln K_j^1] \\
& \leq \max_{x \in X} \{\exp(Z_j^l(x)) - K_j^1[1 + Z_j^l(x) - \ln K_j^1]\} \\
& = \max_{x \in X} \{K_j^1(Z_j^l(x) - Z_j^{ll}) + \exp(Z_j^{ll}) - \exp(Z_j^l(x))\} \\
& \leq \max_{x \in X} \{K_j^1(Z_j^{lu} - Z_j^{ll}) + \exp(Z_j^{ll}) - \exp(Z_j^l(x))\} \\
& = \max_{x \in X} \{\exp(Z_j^{lu}) - \exp(Z_j^{ll}) + \exp(Z_j^{ll}) - \exp(Z_j^l(x))\} \\
& = \max_{x \in X} \{\exp(Z_j^{lu}) - \exp(Z_j^l(x))\} \\
& = \exp(Z_j^{lu}) - \exp(Z_j^l) \\
& \leq \exp(Z_j^{lu})(Z_j^{lu} - Z_j^{ll}) \\
& \leq \exp(\max_{x \in X^0} Z_j^l(x))(Z_j^{lu} - Z_j^{ll}) \\
& \leq \exp(\max_{x \in X^0} Z_j^u(x))(Z_j^{lu} - Z_j^{ll}) \\
& = \exp(\hat{Z}_j^{uu}) \times \left[\sum_{t=1}^{T_j} |\gamma_{jt}| k_{jt}^1 \sum_{i=1}^N |c_{jti}| (\bar{x}_i - \underline{x}_i) + \sum_{t=1}^{T_j} |\gamma_{jt}| k_{jt}^2 \sum_{i=1}^N |e_{jti}| (\bar{x}_i - \underline{x}_i) \right] \\
& = \exp(\hat{Z}_j^{uu}) \times \left[\sum_{i=1}^N \left(\sum_{t=1}^{T_j} k_{jt}^1 |\gamma_{jt} c_{jti}| + \sum_{t=1}^{T_j} k_{jt}^2 |\gamma_{jt} e_{jti}| \right) (\bar{x}_i - \underline{x}_i) \right] \\
& = \exp(\hat{Z}_j^{uu}) \times \left[\sum_{i=1}^N \left(\sum_{t=1}^{T_j} (k_{jt}^1 |\gamma_{jt} c_{jti}| + k_{jt}^2 |\gamma_{jt} e_{jti}|) \right) (\bar{x}_i - \underline{x}_i) \right] \\
& \leq \exp(\hat{Z}_j^{uu}) \times \left[\sum_{i=1}^N \left(\sum_{t=1}^{T_j} \left(\frac{1}{\xi_{jt}^l} |\gamma_{jt} c_{jti}| + \frac{1}{\eta_{jt}^l} |\gamma_{jt} e_{jti}| \right) \right) (\bar{x}_i - \underline{x}_i) \right].
\end{aligned}$$

Therefore, combining the above inequalities together, we have

$$\begin{aligned}
& g^u(x) - U(x) \\
& = \sum_{j=1}^m \alpha_j \left[K_j^2(Z_j^u(x) - Z_j^{ul}) + \exp(Z_j^{ul}) - \exp(Z_j^u(x)) \right] \\
& \quad - \sum_{j=m+1}^p \alpha_j \left[\exp(Z_j^l(x)) - K_j^1[1 + Z_j^l(x) - \ln K_j^1] \right] \\
& \leq \sum_{j=1}^m \alpha_j \exp(\hat{Z}_j^{uu}) \times \left[\sum_{i=1}^N \left(\sum_{t=1}^{T_j} \left(\frac{1}{\xi_{jt}^l} |\gamma_{jt} c_{jti}| + \frac{1}{\eta_{jt}^l} |\gamma_{jt} e_{jti}| \right) \right) (\bar{x}_i - \underline{x}_i) \right] \\
& \quad - \sum_{j=m+1}^p \alpha_j \exp(\hat{Z}_j^{uu}) \times \left[\sum_{i=1}^N \left(\sum_{t=1}^{T_j} \left(\frac{1}{\xi_{jt}^l} |\gamma_{jt} c_{jti}| + \frac{1}{\eta_{jt}^l} |\gamma_{jt} e_{jti}| \right) \right) (\bar{x}_i - \underline{x}_i) \right] \\
& = \sum_{i=1}^N \left\{ \sum_{j=1}^p |\alpha_j| \exp(\hat{Z}_j^{uu}) \left(\sum_{t=1}^{T_j} \left(\frac{1}{\xi_{jt}^l} |\gamma_{jt} c_{jti}| + \frac{1}{\eta_{jt}^l} |\gamma_{jt} e_{jti}| \right) \right) \right\} (\bar{x}_i - \underline{x}_i).
\end{aligned}$$

By the proving process of Theorem 2, for any $x \in X \subseteq X^0$, we also have

$$\begin{aligned}
 & U(x) - g(x) \\
 &= \sum_{j=1}^m \alpha_j \left[\exp(Z_j^u(x)) - \exp(Z_j(x)) \right] - \sum_{j=m+1}^p \alpha_j \left[\exp(Z_j(x)) - \exp(Z_j^l(x)) \right] \\
 &\leq \sum_{j=1}^m \alpha_j \exp(Z_j^u(x)) \left[Z_j^u(x) - Z_j(x) \right] - \sum_{j=m+1}^p \alpha_j \exp(Z_j(x)) \left[Z_j(x) - Z_j^l(x) \right] \\
 &\leq \sum_{j=1}^m \alpha_j \exp(Z_j^u(x)) \left[Z_j^u(x) - Z_j(x) \right] - \sum_{j=m+1}^p \alpha_j \exp(Z_j^u(x)) \left[Z_j(x) - Z_j^l(x) \right] \\
 &\leq \sum_{j=1}^m \alpha_j \exp(\max_{x \in X^0} Z_j^u(x)) \left[Z_j^u(x) - Z_j(x) \right] \\
 &\quad - \sum_{j=m+1}^p \alpha_j \exp(\max_{x \in X^0} Z_j^u(x)) \left[Z_j(x) - Z_j^l(x) \right] \\
 &\leq \sum_{j=1}^m \alpha_j \exp(\hat{Z}_j^{uu}) \times \max_{x \in X} \left[Z_j^u(x) - Z_j(x) \right] \\
 &\quad - \sum_{j=m+1}^p \alpha_j \exp(\hat{Z}_j^{uu}) \times \max_{x \in X} \left[Z_j(x) - Z_j^l(x) \right] \\
 &\leq \sum_{j=1}^m \alpha_j \exp(\hat{Z}_j^{uu}) \times 2 \left[\sum_{i=1}^N \left(\sum_{t=1}^{T_j} \left(\frac{1}{\xi_{jt}^l} |\gamma_{jt} c_{jti}| + \frac{1}{\eta_{jt}^l} |\gamma_{jt} e_{jti}| \right) \right) (\bar{x}_i - \underline{x}_i) \right] \\
 &\quad - \sum_{j=m+1}^p \alpha_j \exp(\hat{Z}_j^{uu}) \times 2 \left[\sum_{i=1}^N \left(\sum_{t=1}^{T_j} \left(\frac{1}{\xi_{jt}^l} |\gamma_{jt} c_{jti}| + \frac{1}{\eta_{jt}^l} |\gamma_{jt} e_{jti}| \right) \right) (\bar{x}_i - \underline{x}_i) \right] \\
 &= \sum_{i=1}^N 2 \left\{ \sum_{j=1}^p |\alpha_j| \exp(\hat{Z}_j^{uu}) \left(\sum_{t=1}^{T_j} \left(\frac{1}{\xi_{jt}^l} |\gamma_{jt} c_{jti}| + \frac{1}{\eta_{jt}^l} |\gamma_{jt} e_{jti}| \right) \right) \right\} (\bar{x}_i - \underline{x}_i).
 \end{aligned}$$

Therefore, by the above conclusions, we can get that

$$\begin{aligned}
 & g^u(x) - g(x) \\
 &= [g^u(x) - U(x)] + [U(x) - g(x)] \\
 &\leq \sum_{i=1}^N \left\{ 3 \sum_{j=1}^p |\alpha_j| \exp(\hat{Z}_j^{uu}) \left(\sum_{t=1}^{T_j} \left(\frac{1}{\xi_{jt}^l} |\gamma_{jt} c_{jti}| + \frac{1}{\eta_{jt}^l} |\gamma_{jt} e_{jti}| \right) \right) \right\} (\bar{x}_i - \underline{x}_i).
 \end{aligned}$$

Theorem 4 From the above derivation, for any $x \in X \subseteq X^0$, the gap between $g^u(x)$ and $g(x)$ with relationship of $(\bar{x} - \underline{x})$ satisfies the following inequality:

$$g^u(x) - g(x) \leq \sum_{i=1}^N \left\{ 3 \sum_{j=1}^p |\alpha_j| \exp(\hat{Z}_j^{uu}) \left(\sum_{t=1}^{T_j} \left(\frac{1}{\xi_{jt}^l} |\gamma_{jt} c_{jti}| + \frac{1}{\eta_{jt}^l} |\gamma_{jt} e_{jti}| \right) \right) \right\} (\bar{x}_i - \underline{x}_i).$$

Proof From the above derivation process, the conclusion is obvious.

3.3.2 Computational complexity of Algorithm 1

Without losing generality, we define the size $\Delta(X)$ of a rectangle $X = \{x \in \mathbb{R}^N \mid \underline{x}_i \leq x_i \leq \bar{x}_i, i = 1, 2, \dots, N\} \subseteq X^0$ by

$$\Delta(X) := \max\{\bar{x}_i - \underline{x}_i \mid i = 1, 2, \dots, N\}.$$

Besides, for convenience, we let

$$\lambda = \max \left\{ 3 \sum_{j=1}^p |\alpha_j| \exp(\hat{Z}_j^{uu}) \left(\sum_{t=1}^{T_j} \left(\frac{1}{\hat{\xi}_{jt}} |\gamma_{jt} c_{jti}| + \frac{1}{\hat{\eta}_{jt}} |\gamma_{jt} e_{jti}| \right) \right) \mid i = 1, \dots, N \right\}. \quad (17)$$

Theorem 5 For any given termination error $\varepsilon > 0$, if there exists a rectangle X^k , which is produced by Algorithm 1 at k_{th} iteration, and which satisfies that $\Delta(X^k) \leq \frac{\varepsilon}{N\lambda}$, then we can follow that

$$UB(X^k) - v^k \leq \varepsilon,$$

where $UB(X^k)$ is the optimum value of LPRP over the rectangle X^k , and v^k is the best currently known lower bound of the global optimum value for GLFP problem.

Proof Suppose that x^k is the optimal solution of the linear programming relaxation LPRP over the rectangle X^k , then x^k is a feasible solution of GLFP problem over X^k . By the definition of the lower bound v^k and $UB(X^k)$, we have that

$$g(x^k) \leq v^k \leq UB(X^k) = g^u(x^k).$$

Thus, according to Steps 4 and 5 of Algorithm 1 and Theorem 4, it follows that

$$\begin{aligned} UB(X^k) - v^k &\leq g^u(x^k) - g(x^k) \\ &\leq \sum_{i=1}^N \left\{ 3 \sum_{j=1}^p |\alpha_j| \exp(\hat{Z}_j^{uu}) \left(\sum_{t=1}^{T_j} \left(\frac{1}{\hat{\xi}_{jt}} |\gamma_{jt} c_{jti}| + \frac{1}{\hat{\eta}_{jt}} |\gamma_{jt} e_{jti}| \right) \right) \right\} (\bar{x}_i^k - \underline{x}_i^k) \\ &\leq \sum_{i=1}^N (\Delta(X^k) \lambda) \\ &\leq N \lambda \Delta(X^k). \end{aligned}$$

Further, by the above inequalities and $\Delta(X^k) \leq \frac{\varepsilon}{N\lambda}$, we can get that

$$UB(X^k) - v^k \leq \sum_{i=1}^N (\lambda \Delta(X^k)) \leq \varepsilon.$$

The proof of the Theorem 5 is completed.

From Step 5 of Algorithm 1 and Theorem 5, it is obvious, if $\Delta(X^k) \leq \frac{\varepsilon}{N\lambda}$, then the rectangle X^k will be deleted. Hence, if the sizes of all rectangles X produced by Algorithm 1 satisfy $\Delta(X) \leq \frac{\varepsilon}{N\lambda}$, then Algorithm 1 will be stopped. By Theorem 5, we can derive the maximum iterations of the algorithm, which is given by following Theorem 6.

Theorem 6 For any given termination error $\varepsilon > 0$, Algorithm 1 can obtain a global ε -optimal solution of the GLFP problem in at most

$$\hat{N} = 2^{\sum_{i=1}^N \lceil \log_2 \frac{N\lambda(\bar{x}_i^0 - x_i^0)}{\varepsilon} \rceil} - 1$$

iterations, where λ is given in (17), and $X^0 = \prod_{i=1}^N X_i^0$ with $X_i^0 = [x_i^0, \bar{x}_i^0]$.

Proof Suppose that a rectangle $X = \prod_{i=1}^N X_i$ with $X_i = [x_i, \bar{x}_i] \subseteq X_i^0$ is selected from X^0 for partitioning in Step 2 of the algorithm for each iteration. Without losing generality, assume that after N_i iterations there exists a subinterval $X_i^{N_i} = [x_i^{N_i}, \bar{x}_i^{N_i}]$ of the interval $X_i^0 = [x_i^0, \bar{x}_i^0]$ such that, for each $i = 1, 2, \dots, N$, we have that

$$\bar{x}_i^{N_i} - x_i^{N_i} \leq \frac{\varepsilon}{N\lambda}. \quad (18)$$

By the branching operation of the algorithm, it can obtain that, for each $i = 1, 2, \dots, N$,

$$\bar{x}_i^{N_i} - x_i^{N_i} = \frac{1}{2^{N_i}} (\bar{x}_i^0 - x_i^0). \quad (19)$$

By (18) and (19), for each $i = 1, 2, \dots, N$, we can follow that

$$\frac{1}{2^{N_i}} (\bar{x}_i^0 - x_i^0) \leq \frac{\varepsilon}{N\lambda}.$$

Thus, for each $i = 1, 2, \dots, N$, we can obtain that

$$N_i \geq \log_2 \frac{N\lambda(\bar{x}_i^0 - x_i^0)}{\varepsilon}.$$

For each $i = 1, 2, \dots, N$, let

$$\bar{N}_i = \lceil \log_2 \frac{N\lambda(\bar{x}_i^0 - x_i^0)}{\varepsilon} \rceil.$$

Then, after $\hat{N}_1 = \sum_{i=1}^N \bar{N}_i$ iterations, the algorithm will generate at most $\hat{N}_1 + 1$ sub-rectangles, and denote these sub-rectangles as $X^1, X^2, \dots, X^{\hat{N}_1+1}$, which satisfy that

$$\Delta(X^\tau) = 2^{\hat{N}_1 - \tau} \Delta(X^{\hat{N}_1}) = 2^{\hat{N}_1 - \tau} \Delta(X^{\hat{N}_1+1}), \quad \tau = \hat{N}_1, \hat{N}_1 - 1, \dots, 2, 1, \quad (20)$$

where $\Delta(X^{\hat{N}_1}) = \Delta(X^{\hat{N}_1+1}) = \max\{\bar{x}_i^{\hat{N}_i} - \underline{x}_i^{\hat{N}_i} \mid i = 1, 2, \dots, N\}$ and

$$X^0 = \bigcup_{\tau=1}^{\hat{N}_1+1} X^\tau. \quad (21)$$

At the same time, put this $\hat{N}_1 + 1$ sub-rectangles into the set $\Omega^{\hat{N}_1+1}$, i.e.,

$$\Omega^{\hat{N}_1+1} = \{X^\tau \mid \tau = 1, 2, \dots, \hat{N}_1 + 1\}.$$

From (18), we can follow that

$$\Delta(X^{\hat{N}_1}) = \Delta(X^{\hat{N}_1+1}) \leq \frac{\varepsilon}{N\lambda}. \quad (22)$$

For the convenience of narration, we denote $\bar{\Delta} = \Delta(X^{\hat{N}_1}) = \Delta(X^{\hat{N}_1+1})$. Thus, from (22), it follows that

$$\bar{\Delta} \leq \frac{\varepsilon}{N\lambda}. \quad (23)$$

Thus, from Theorem 5 and Step 5 of the algorithm, there doesn't exist the global optimal solution of GLFP problem over $X^{\hat{N}_1}$ and $X^{\hat{N}_1+1}$, so that the rectangles $X^{\hat{N}_1}$ and $X^{\hat{N}_1+1}$ will be deleted from the set $\Omega^{\hat{N}_1+1}$. Next, the residual sub-rectangles will be put into the set $\Omega^{\hat{N}_1}$, where

$$\Omega^{\hat{N}_1} = \Omega^{\hat{N}_1+1} \setminus \{X^{\hat{N}_1}, X^{\hat{N}_1+1}\} = \{X^\tau \mid \tau = 1, 2, \dots, \hat{N}_1 - 1\}.$$

and the residual sub-rectangles X^t , $t = 1, 2, \dots, \hat{N}_1 - 1$, will be considered further.

Next, we investigate the rectangle $X^{\hat{N}_1-1}$. By (20), using the branching operation to subdivide $X^{\hat{N}_1-1}$ once, we can get two new sub-rectangles $X^{\hat{N}_1-1,1}$ and $X^{\hat{N}_1-1,2}$, such that

$$X^{\hat{N}_1-1} = X^{\hat{N}_1-1,1} \cup X^{\hat{N}_1-1,2} \quad (24)$$

and

$$\Delta(X^{\hat{N}_1-1}) = 2\Delta(X^{\hat{N}_1-1,1}) = 2\Delta(X^{\hat{N}_1-1,2}) = 2\bar{\Delta}. \quad (25)$$

Thus, after $\hat{N}_1 + (2^1 - 1)$ iterations, by (23)–(25), the rectangle $X^{\hat{N}_1-1}$ will be deleted from the set $\Omega^{\hat{N}_1}$, and the residual rectangles will be put into the set $\Omega^{\hat{N}_1-1}$, where

$$\Omega^{\hat{N}_1-1} = \Omega^{\hat{N}_1+1} \setminus \{X^{\hat{N}_1-1}, X^{\hat{N}_1}, X^{\hat{N}_1+1}\} = \{X^\tau \mid \tau = 1, 2, \dots, \hat{N}_1 - 2\}.$$

Similarly, when Algorithm 1 carries out $\hat{N}_1 + (2^1 - 1) + (2^2 - 1)$ iterations, the rectangle $X^{\hat{N}_1-2}$ will be deleted from the set $\Omega^{\hat{N}_1-1}$ and put the residual sub-rectangles into the set $\Omega^{\hat{N}_1-2}$, where

$$\Omega^{\hat{N}_1-2} = \Omega^{\hat{N}_1+1} \setminus \{X^{\hat{N}_1-2}, X^{\hat{N}_1-1}, X^{\hat{N}_1}, X^{\hat{N}_1+1}\} = \{X^\tau | \tau = 1, 2, \dots, \hat{N}_1 - 3\}.$$

Repeat the foregoing operations, until all sub-rectangles X^τ ($\tau = 1, 2, \dots, \hat{N}_1 + 1$) are deleted from X^0 . Thus, by (21), the algorithm can be stopped in at most

$$\begin{aligned} \hat{N} &= \hat{N}_1 + (2^1 - 1) + (2^2 - 1) + (2^3 - 1) + \dots + (2^{\hat{N}_1-1} - 1) = 2^{\hat{N}_1} - 1 \\ &= 2^{\sum_{i=1}^N \lceil \log_2 \frac{N\lambda(x_i^0 - \underline{x}_i^0)}{\varepsilon} \rceil} - 1 \end{aligned}$$

iterations, and the proof of the theorem is complete.

Remark 1 By Theorem 6 and steps of the algorithm, the running time of the proposed algorithm for getting a global ε -optimal solution of the problem (GLFP) is bounded by

$$2\hat{N}\hat{T}(M, N),$$

where $\hat{T}(M, N)$ denotes the time spent to solve a linear programming relaxation problem with M linear inequality constraints and N variables.

4 Numerical Experiments

To verify the feasibility of Algorithm 1, some test examples are solved on Intel(R) Core(TM)2 i5-4590s CPU @3.0GHz microcomputer, the proposed algorithm is coded in C++ software, given the convergent error $\varepsilon = 10^{-6}$, the simplex method is used to solve the LPRP, and numerical results are illustrated in Tables 1 and 2. In Tables 1 and 2, we denote “Iter.” as the number of iterations of the algorithm. Although these test examples have a relatively small number of variables, they are still quite challenging.

Example 1 (Refs. [24,25]).

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

Table 1 Numerical results for Examples 1–5

Example	Refs.	Optimal solution	Optimal value	Iter.	Time/s
1	Ours	(1.5, 1.5)	4.912 587 413	29	0.127 15
	[24]	(1.5, 1.5)	4.912 5	113	201.626
	[25]	(3.0, 4.0)	5.0	32	1.089 28
2	Ours	(0.0, 0.0, 0.0)	3.0	37	0.109 40
	[26]	(0.001 3, 1.672 5, 0.0)	3.0	1 033	99.357 0
3	Ours	(1.111 111 111, 0.0, 0.0)	4.090 702 948	21	0.083 11
	[27]	(0.001 3, 0.0, 0.0)	4.000 1	1 640	120.635
4	Ours	(0, 0)	0.533 333 333	14	0.062 44
	[27]	(0, 0)	0.533 333 333	34	0.205 70
5	Ours	(1.0, 1.0)	0.336 037 467	8	0.046 82
	[17]	(1.000 000 69, 1.000 000 46)	0.336 036 010	557	6.000 00
	[27]	(1.0, 1.0)	0.336 037 467	16	0.102 61
	[28]	(1.000 004 57, 1.000 000 76)	0.336 033 923	1 572	14.506 0

Table 2 Numerical results for Example 6

(p, n)	Iter.	Maximum node	Optimal value	Time/s
(6, 20)	176	2	−12.002 106 539	11.471 7
(9, 20)	182	31	−11.237 761 937	19.027 0
(11, 20)	191	31	−10.206 479 224	25.152 8
(13, 20)	195	31	−10.591 983 120	29.969 7
(15, 20)	188	28	−14.515 684 594	32.762 2
(17, 20)	191	27	−18.108 094 898	38.021 0
(20, 10)	652	145	−2.853 202 555 0	50.242 7
(40, 10)	2 182	760	−23.908 886 727	301.062
(50, 10)	1 079	341	−39.183 806 458	202.096
(60, 10)	2 305	773	−28.696 187 644	516.118
(70, 10)	9 388	3 690	−18.917 551 937	2 405.86
(80, 10)	16 869	6 734	−15.387 765 122	5 015.31
(90, 10)	21 904	8 781	−12.015 057 419	6 872.87

Example 2 (Ref. [26]).

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

Example 3 (Ref. [26]).

$$\left\{ \begin{array}{l} \max \frac{4x_1 + 3x_2 + 3x_3 + 50}{3x_2 + 3x_3 + 50} + \frac{3x_1 + 4x_2 + 50}{4x_1 + 4x_2 + 5x_3 + 50} + \frac{x_1 + 2x_2 + 5x_3 + 50}{x_1 + 5x_2 + 5x_3 + 50} \\ \quad + \frac{x_1 + 2x_2 + 4x_3 + 50}{5x_2 + 4x_3 + 50} \\ \text{s. t. } 2x_1 + x_2 + 5x_3 \leq 10, \\ \quad x_1 + 6x_2 + 3x_3 \leq 10, \\ \quad 5x_1 + 9x_2 + 2x_3 \leq 10, \\ \quad 9x_1 + 7x_2 + 3x_3 \leq 10, \\ \quad x_1, x_2, x_3 \geq 0. \end{array} \right.$$

Example 4 (Ref. [27]).

$$\left\{ \begin{array}{l} \min \frac{-x_1 + 2x_2 + 2}{3x_1 - 4x_2 + 5} \times \frac{4x_1 - 3x_2 + 4}{-2x_1 + x_2 + 3} \\ \text{s. t. } x_1 + x_2 \leq 1.5, \\ \quad x_1 \leq x_2, \\ \quad 0 \leq x_1 \leq 1, \quad 0 \leq x_2 \leq 1. \end{array} \right.$$

Example 5 (Refs. [17,27,28]).

$$\left\{ \begin{array}{l} \min \left(\frac{x_1 + x_2 + 1}{x_1 + x_2 + 2} \right)^{1.1} \times \left(\frac{x_1 + x_2 + 3}{x_1 + x_2 + 4} \right)^{1.2} \times \left(\frac{x_1 + x_2 + 5}{x_1 + x_2 + 6} \right)^{1.2} \\ \quad \times \left(\frac{x_1 + x_2 + 7}{x_1 + x_2 + 8} \right)^{1.1} \\ \text{s. t. } x_1 - x_2 = 0, \\ \quad 1.0 \leq x_1 \leq 2.0, \quad 1.0 \leq x_2 \leq 2.0. \end{array} \right.$$

To demonstrate the reliability and validity of the Algorithm 1, we have solved randomly generated Example 6, and numerical results are listed in Table 2.

Example 6

$$\left\{ \begin{array}{l} \max g(x) = \sum_{j=1}^p \alpha_j \left(\sum_{i=1}^N c_{ji} x_i + d_j \right)^{\gamma_{j1}} \left(\sum_{i=1}^N e_{ji} x_i + f_j \right)^{\gamma_{j2}} \\ \text{s. t. } Ax \leq b, \quad 0 \leq x \leq 1, \end{array} \right.$$

where p are all positive integers, $A \in \mathbb{R}^{M \times N}$, $b \in \mathbb{R}^M$; c_{ji} and e_{ji} , $j = 1, \dots, p$, $i = 1, \dots, N$, are all randomly generated in the unit interval $[0, 1]$; d_j and f_j , $j = 1, \dots, p$, are all randomly generated in the interval $(0, 1]$; α_j , γ_{j1} and γ_{j2} , $j = 1, \dots, p$, are all randomly generated in the interval $[-1, 1]$; all elements of A are randomly generated in the unit interval $[0, 1]$; all elements of b are randomly generated in the interval $[0, 16]$; $M = 5$.

Obviously, for any $A \in \mathbb{R}^{M \times N}$, and for any $b \in [0, 16]$, we have $A \times 0 = 0 \leq b$. Therefore, $0 \in \{x \in \mathbb{R}^N \mid Ax \leq b, x \geq 0\}$ always holds; it is to say the feasible region of Example 6 has always feasible point 0.

In summary, Table 1 shows that Algorithm 1 yields better or at least the same global optimal solutions and optimal values compared with the known algorithms; Table 2 shows the robustness and stability of the proposed algorithm.

5 Concluding Remarks

In this paper, we present a branch-and-bound algorithm for globally solving the generalized linear fractional programming problem. In this algorithm, we construct a two-level linear relaxation method, which is used to derive the linear programming relaxation problem of GLFP problem. The established linear programming relaxation problem can provide a reliable upper bound in the proposed branch-and-bound algorithm. By partitioning the initial rectangle region and solving a series of linear programming relaxation problems, the proposed algorithm is globally convergent to the global optimal solution for the initial GLFP problem. In addition, we derive the gap between $g''(x)$ and $g(x)$ with relationship of $(\bar{x} - \underline{x})$ and present the computational complexity of the proposed algorithm. Finally, numerical experimental results verify the feasibility of the proposed algorithm.

Acknowledgements The authors are very grateful to the responsible editors and the anonymous referees for their valuable comments and suggestions, which have greatly improved the earlier version of this paper.

Declarations

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

References

- [1] Stancu-Minasian, I.M.: Fractional Programming: Theory, Methods and Applications. Kluwer, Dordrecht (1997)
- [2] Nguyen, T.H.P., Tuy, H.: A unified monotonic approach to generalized linear fractional programming. *J. Glob. Optim.* **26**, 229–259 (2003)
- [3] Gao, Y.L., Xu, C.X., Yan, Y.L.: An outcome-space finite algorithm for solving linear multiplicative programming. *Appl. Math. Comput.* **179**(2), 494–505 (2006)
- [4] Wang, C.F., Liu, S.Y.: A new linearization method for generalized linear multiplicative programming. *Comput. Oper. Res.* **38**, 1008–1013 (2011)
- [5] Shen, P., Jiao, H.: Linearization method for a class of multiplicative programming with exponent. *Appl. Math. Comput.* **183**(1), 328–336 (2006)
- [6] Ryoo, H.S., Sahinidis, N.V.: Global optimization of multiplicative programs. *J. Glob. Optim.* **26**, 387–418 (2003)
- [7] Jiao, H., Liu, S.: Global optimization algorithm for a generalized linear multiplicative programming. *J. Appl. Math. Comput.* **40**(1–2), 551–568 (2012)
- [8] Jiao, H., Liu, S., Yin, J., Zhao, Y.: Outcome space range reduction method for global optimization of sum of linear ratios problems. *Open Math.* **14**, 736–746 (2016)
- [9] Jiao, H., Liu, S.: Range division and compression algorithm for quadratically constrained sum of quadratic ratios. *Comput. Appl. Math.* **36**(1), 225–247 (2017)

- [10] Jiao, H., Liu, S.: A practicable branch and bound algorithm for sum of linear ratios problem. *Eur. J. Oper. Res.* **243**(3), 723–730 (2015)
- [11] Shen, P., Li, X.: Branch-reduction-bound algorithm for generalized geometric programming. *J. Glob. Optim.* **56**(3), 1123–1142 (2013)
- [12] Wang, Y.J., Liang, Z.A.: A deterministic global optimization algorithm for generalized geometric programming. *Appl. Math. Comput.* **168**, 722–737 (2005)
- [13] Gao, Y., Xu, C., Wang, Y., Zhang, L.: A new two-level linear relaxed bound method for geometric programming problems. *Appl. Math. Comput.* **164**(1), 117–131 (2005)
- [14] Shen, P.: Linearization method of global optimization for generalized geometric programming. *Appl. Math. Comput.* **162**, 353–370 (2005)
- [15] Shen, P., Zhang, K.: Global optimization of signomial geometric programming using linear relaxation. *Appl. Math. Comput.* **150**, 99–114 (2004)
- [16] Shen, P., Li, X., Jiao, H.: Accelerating method of global optimization for signomial geometric programming. *J. Comput. Appl. Math.* **214**, 66–77 (2008)
- [17] Jiao, H., Guo, Y., Shen, P.: Global optimization of generalized linear fractional programming with nonlinear constraints. *Appl. Math. Comput.* **183**(2), 717–728 (2006)
- [18] Liu, X., Gao, Y., Zhang, B., Tian, F.: A new global optimization algorithm for a class of linear fractional programming. *Mathematics* **7**, 867 (2019)
- [19] Zhang, B., Gao, Y., Liu, X., Huang, X.: Output-space branch-and-bound reduction algorithm for a class of linear multiplicative programs. *Mathematics* **8**, 315 (2020)
- [20] Shen, P., Huang, B., Wang, L.: Range division and linearization algorithm for a class of linear ratios optimization problems. *J. Comput. Appl. Math.* **350**, 324–342 (2019)
- [21] Shen, P., Zhu, Z., Chen, X.: A practicable contraction approach for the sum of the generalized polynomial ratios problem. *Eur. J. Oper. Res.* **278**(1), 36–48 (2019)
- [22] Shen, P., Wang, C.: Global optimization for sum of generalized fractional functions. *J. Comput. Appl. Math.* **214**, 1–12 (2008)
- [23] Jiao, H., Liu, S.: An efficient algorithm for quadratic sum-of-ratios fractional programs problem. *Numer. Func. Anal. Opt.* **38**(11), 1426–1445 (2017)
- [24] Gao Y., Jin S.: A global optimization algorithm for sum of linear ratios problem, *J. Appl. Math.* (2013), Article ID 276245, 7 pages
- [25] Wang, C., Shen, P.: A global optimization algorithm for linear fractional programming. *Appl. Math. Comput.* **204**, 281–287 (2008)
- [26] Pei, Y., Zhu, D.: Global optimization method for maximizing the sum of difference of convex functions ratios over nonconvex region. *J. Appl. Math. Comput.* **41**(1–2), 153–169 (2013)
- [27] Jiao, H.: A branch and bound algorithm for globally solving a class of nonconvex programming problems. *Nonlinear Anal-Theor.* **70**, 1113–1123 (2009)
- [28] Jiao, H., Liu, S., Zhao, Y.: Effective algorithm for solving the generalized linear multiplicative problem with generalized polynomial constraints. *Appl. Math. Model.* **39**(23–24), 7568–7582 (2015)