Journal of Inequalities and Applications
a SpringerOpen Journal

RESEARCH Open Access

Regional division and reduction algorithm for minimizing the sum of linear fractional functions

Pei-Ping Shen* and Ting Lu

*Correspondence:
shenpeiping@163.com
College of Mathematics and Information Science, Henan Normal University, Xinxiang, P.R. China

Abstract

This paper presents a practicable regional division and cut algorithm for minimizing the sum of linear fractional functions over a polyhedron. In the algorithm, by using an equivalent problem (P) of the original problem, the proposed division operation generalizes the usual standard bisection, and the deleting and reduction operations can cut away a large part of the current investigated region in which the global optimal solution of (P) does not exist. The main computation involves solving a sequence of univariate equations with strict monotonicity. The proposed algorithm is convergent to the global minimum through the successive refinement of the solutions of a series of univariate equations. Numerical results are given to show the feasibility and effectiveness of the proposed algorithm.

Keywords: Global optimization; Sum of linear ratios; Regional division and reduction; Fractional programming

1 Introduction

Consider the following class of fractional programming:

$$(\mathrm{FP}):\begin{cases}\min & \sum_{i=1}^{N}\frac{c_i^\top y+c_{0i}}{d_i^\top y+d_{0i}}\\ \text{s.t.} & Ay\le b, y\ge 0,\end{cases}$$

where $A=(a_{rj})_{m\times n}$ is a real matrix, $c_i=(c_{ij})_{1\times n}$ and $d_i=(d_{ij})_{1\times n}$ are real vectors for each $i=1,\dots,N$, $b=(b_r)_{m\times 1}$, $c_{0i},d_{0i}\in R$. In addition, we assume that

$$\bar{D}=\{y\in R^n \mid Ay\le b, y\ge 0\}$$

is a nonempty compact set.

Since each $d_i^\top y+d_{0i}$ is a continuous function on $\bar{D}$, by the intermediate value theorem we can obtain that $d_i^\top y+d_{0i}>0$ or $d_i^\top y+d_{0i}<0$ for any $y\in\bar{D}$. Consequently,

$$\sum_{i=1}^{N}\frac{c_i^\top y+c_{0i}}{d_i^\top y+d_{0i}}=\sum_{i\in I^+}\frac{c_i^\top y+c_{0i}}{d_i^\top y+d_{0i}}+\sum_{i\in I^-}\frac{c_i^\top y+c_{0i}}{d_i^\top y+d_{0i}}=\sum_{i\in I^+}\frac{c_i^\top y+c_{0i}}{d_i^\top y+d_{0i}}+\sum_{i\in I^-}\frac{-(c_i^\top y+c_{0i})}{-(d_i^\top y+d_{0i})},$$

where $I^+ = \{i \mid d_i^\top y + d_{0i} > 0, i = 1, \ldots, N, \forall y \in \bar{D}\}$ and $I^- = \{i \mid d_i^\top y + d_{0i} < 0, i = 1, \ldots, N, \forall y \in \bar{D}\}$. Additionally, notice that

$$\min \sum_{i=1}^{N} \frac{c_i^\top y + c_{0i}}{d_i^\top y + d_{0i}} \Longleftrightarrow \min \sum_{i=1}^{N} \left(\frac{c_i^\top y + c_{0i}}{d_i^\top y + d_{0i}} + \bar{M}_i \right)$$
$$\Longleftrightarrow \min \sum_{i=1}^{N} \frac{c_i^\top y + c_{0i} + \bar{M}_i (d_i^\top y + d_{0i})}{d_i^\top y + d_{0i}}$$

by choosing a sufficiently large value $\bar{M}_i$ $(i = 1, \ldots, N)$ with $c_i^\top y + c_{0i} + \bar{M}_i (d_i^\top y + d_{0i}) > 0$ for any $y \in \bar{D}$. Therefore, in the following, without loss of generality, we suppose that $c_i^\top y + c_{0i} > 0$ and $d_i^\top y + d_{0i} > 0$, $\forall y \in \bar{D}$, for each $i = 1, \ldots, N$.

Problem (FP) is a well-known class among fractional programming problems. Theoretically, it is NP-hard [1, 2]. The primary challenges in solving problem (FP) arise from a lack of useful properties (convexity or otherwise) and from the number of ratios. In general, problem (FP) possesses more local optimal solutions that are not globally optimal [3], and so problem (FP) owns major theoretical and computational difficulties. From an application point view, this problem has a large deal of applications; for instance, traffic and economic domain [4, 5], multistage stochastic shipping problems [6], data envelopment analysis [7], and queueing-location problems [8]. The reader is referred to a survey to find many other applications [4, 5, 9–11].

Many algorithms have been proposed to solve problem (FP) with a limited number of ratios [4, 12–19]. For instance, Wang and Shen [4] give an efficient branch and bound algorithm by using a transformation technique and the linear relaxation programming of the objective function. By applying Lagrangian duality theory, Benson [9] presents a simplicial branch and bound duality-bounds algorithm. Carlsson and Shi [10] propose a linear relaxation algorithm with lower dimension which is performed on a $2N$-dimensional domain instead of the original n-dimensional one, the computational time is long with larger N. Ji and Zhang [16] consider a deterministic global optimization algorithm by utilizing a transformation technique and a linearizing method. Jiao et al. [17, 18] present the branch and bound algorithms for globally solving sum of linear and generalized polynomial ratios problems, by solving a sequence of linear relaxation programming problems. In short, most of them (see [4, 9, 16–18] for example) are branch and bound algorithms. The key idea behind such algorithms mentioned above is that the branch and bound operator is performed on an N-dimensional region (or the correspondence relating to N and n) rather than the native n-dimensional feasible set, that is, they all work on a space whose dimension increases with the number N of ratios.

In this article, a new division and reduction algorithm is proposed for globally solving problem (FP). To solve problem (FP), an equivalent optimization problem (P), whose objective function is just a simple univariate, is first presented by exploiting the feature of this problem. Then, in order to design a more efficient algorithm, several basic operations: division, deleting, and reduction, are incorporated into a similar branch and bound framework by utilizing the particular structure of problem (P). Compared with the usual branch and bound (BB) methods (e.g., [4, 5, 9, 10, 16]) mentioned above, the goal of this research is three fold. First, the proposed bounding operation is simple since the lower bound of the subproblem of each node can be achieved easily only by arithmetic computations, distinguishing it from the ones obtained by solving convex/linear programs in the usual BB

methods. Second, the reduction operation that does not appear in other BB methods is used to tighten the range of each variable, such that the growth of the branch tree can be suppressed. Moreover, the main computational cost of the algorithm is to implement the reduction operation which solves univariate equations with strict monotonicity. Third, the problem in this paper is more general than the others considered in [10, 11], since we only request $d_i^\top y + d_{0i} \neq 0$ for each i. Further, the proposed adaptive division operation both generalizes and is superior to the usual standard bisection in BB methods according to the numerical computational result in Sect. 5. Also, the computational results of the problem with the large number of ratio terms can be obtained to illustrate the feasibility and validity of the proposed algorithm.

This paper is summarized as follows. In Sect. 2, by using a conversion strategies, the original problem (FP) is transformed into an equivalent optimization problem (P). The adaptive division, deleting, and reduction operations are shown in Sect. 3. In Sect. 4 we give the proposed algorithm and its convergence. Some numerical results demonstrate the feasibility and availability of the algorithm in Sect. 5.

2 Equivalent problem

For solving problem (FP), we first convert the primary problem (FP) into an equivalent optimization problem (P), in which the objective function is a single variable and the constraint functions are the difference of two increasing functions. To see that such a reformulation is possible, let us denote, for each $i = 1, \dots, N$,

$$L_i = \frac{1}{\max_{y\in\bar{D}} d_i^\top y + d_{0i}}, \qquad U_i = \frac{1}{\min_{y\in\bar{D}} d_i^\top y + d_{0i}}.$$

Clearly, $L_i, U_i > 0$ for each i. Additionally, by introducing some additional variables $w = (w_1, w_2, \dots, w_N) \in R^N$, from (FP) we then obtain the following equivalent problem:

$$(\mathrm{FP1}): \begin{cases} \min & \sum_{i=1}^N w_i(c_i^\top y + c_{0i}) \\ \text{s.t.} & 1 - w_i(d_i^\top y + d_{0i}) \le 0, \quad i = 1, \dots, N, \\ & L_i \le w_i \le U_i, \quad i = 1, \dots, N, \\ & y \in \bar{D}. \end{cases}$$

For simplicity, we denote

$$I_0^+ = \{i \mid c_{0i} > 0, i = 1, \dots, N\}, \qquad I_0^- = \{i \mid c_{0i} < 0, i = 1, \dots, N\},$$
$$J_i^+ = \{j \mid c_{ij} > 0, j = 1, \dots, n\}, \qquad J_i^- = \{j \mid c_{ij} < 0, j = 1, \dots, n\},$$
$$T_i^+ = \{j \mid d_{ij} > 0, j = 1, \dots, n\}, \qquad T_i^- = \{j \mid d_{ij} < 0, j = 1, \dots, n\},$$
$$M_r^+ = \{j \mid a_{rj} > 0, j = 1, \dots, n\}, \qquad M_r^- = \{j \mid a_{rj} < 0, j = 1, \dots, n\}.$$

Define a box as follows:

$$[\underline{y}, \bar{y}] = \left\{ y \in R^n \mid y_j^l \triangleq \min_{y\in\bar{D}} y_j \le y_j \le y_j^u \triangleq \max_{y\in\bar{D}} y_j, j = 1, \dots, n \right\}.$$

Based on the above notations, by introducing an extra variable $y_0 \in R^+$, we can convert the above problem (FP1) into the form:

$$(\mathrm{FP2}): \begin{cases} \min & y_0 \\ \text{s.t.} & \sum_{i=1}^{N} \sum_{j\in J_i^+} c_{ij} w_i y_j \\ & \quad + \sum_{i\in I_0^+} c_{0i} w_i - \sum_{i=1}^{N} \sum_{j\in J_i^-} (-c_{ij}) w_i y_j - \sum_{i\in I_0^-} (-c_{0i}) w_i - y_0 \le 0, \\ & - \sum_{j\in T_i^+} d_{ij} w_i y_j + \sum_{j\in T_i^-} (-d_{ij}) w_i y_j - d_{0i} w_i + 1 \le 0, \quad i = 1,\dots,N, \\ & \sum_{j\in M_r^+} a_{rj} y_j - b_r - \sum_{j\in M_r^-} (-a_{rj}) y_j \le 0, \quad r = 1,\dots,m, \\ & L_i \le w_i \le U_i, \quad i = 1,\dots,N, \\ & y \in [\underline{y}, \bar{y}], \qquad y_0 \in [y_0^l, y_0^u], \end{cases}$$

where

$$y_0^l = \sum_{i=1}^{N} \sum_{j=1}^{n} \min\left\{c_{ij} L_i y_j^l, c_{ij} U_i y_j^u\right\} + \sum_{i=1}^{N} \min\{c_{0i} L_i, c_{0i} U_i\},$$

$$y_0^u = \sum_{i=1}^{N} \sum_{j=1}^{n} \max\left\{c_{ij} L_i y_j^l, c_{ij} U_i y_j^u\right\} + \sum_{i=1}^{N} \max\{c_{0i} L_i, c_{0i} U_i\}.$$

Then problems (FP2) and (FP) are equivalent in the sense of the following result.

Proposition 2.1 *$y^* \in R^n$ is a globally optimal solution for problem* (FP) *if and only if $(y_0^*, y^*, w^*) \in R^{n+N+1}$ with $y_0^* \in R$ and $w^* \in R^N$ is a globally optimal solution for problem* (FP2), *where $y_0^* = \sum_{i=1}^{N} w_i^* (c_i^\top y^* + c_{0i})$ and $w_i^* = \frac{1}{d_i(y^*)+d_{0i}}$ for each $i = 1,\dots,N$.*

Proof The proof of this result is obvious. □

From Proposition 2.1, notice that, in order to globally solve problem (FP), we may globally solve problem (FP2) instead. Moreover, it is easy to see that the global optimal values of problems (FP) and (FP2) are equal.

In addition, for convenience of the following discussion, let us denote $x = (y_0, y, w) \in R^{n+N+1}$ with $y \in R^n$, $w \in R^N$. Then problem (FP2) can be rewritten as problem (P):

$$(\mathrm{P}): \begin{cases} \min & F(x) = x_0 \\ \text{s.t.} & G_k(x) \le 0, \quad k = 0,\dots,m+N, \\ & x \in D^0, \end{cases}$$

where

$$\begin{aligned} D^0 &= \{x \in R^{n+N+1} \mid x_j^l \le x_j \le x_j^u, j = 0,1,\dots,n+N\} \\ &= \left\{ x \in R^{n+N+1} \,\middle|\, \begin{array}{l} x_j^l \triangleq y_j^l \le x_j = y_j \le x_j^u \triangleq y_j^u, j = 0,1,\dots,n, \\ L_{j-n} \le x_j = w_{j-n} \le U_{j-n}, \quad j = n+1,\dots,n+N \end{array} \right\} \end{aligned}$$

and $G_k(x) = G_k^+(x) - G_k^-(x)$ with $G_k^+(x)$, $G_k^-(x)$ being increasing functions given by

$$G_k^+(x) = \begin{cases} \sum_{i=1}^{N} \sum_{j\in J_i^+} c_{ij}x_j x_{i+n} + \sum_{i\in I_0^+} c_{0i}x_{i+n}, & k = 0, \\ \sum_{j\in T_k^-} (-d_{kj})x_j x_{k+n} + 1, & k = 1,\dots,N', \\ \sum_{j\in T_k^-} (-d_{kj})x_j x_{k+n} + (-d_{0k})x_{k+n} + 1, & k = N'+1,\dots,N, \\ \sum_{j\in M_{k-N}^+} a_{k-N,j}x_j - b_{k-N}, & k = N+1,\dots,N+m, \end{cases}$$

and

$$G_k^-(x) = \begin{cases} \sum_{i=1}^{N} \sum_{j\in J_i^-} (-c_{ij})x_j x_{i+n} + \sum_{i\in I_0^-} (-c_{0i})x_{i+n}, & k = 0, \\ \sum_{j\in T_k^+} d_{kj}x_j x_{k+n} + d_{0k}x_{k+n}, & k = 1,\dots,N', \\ \sum_{j\in T_k^+} d_{kj}x_j x_{k+n}, & k = N'+1,\dots,N, \\ \sum_{j\in M_{k-N}^-} (-a_{k-N,j})x_j, & k = N+1,\dots,N+m. \end{cases}$$

Note that both problem (FP) and problem (P) are equivalent according to Proposition 2.1, hence for globally solving problem (FP), the algorithm to be presented concentrates on how to solve problem (P).

3 Essential operations

For solving problem (P), we first give the following concept about an approximate optimal solution.

Definition 3.1 For given $\varepsilon, \eta \geq 0$, let

$$D_\varepsilon = \left\{x \in D^0 \mid G_k(x) < \varepsilon, k = 0, 1, \dots, m+N\right\},$$

a point $x \in D_\varepsilon$ is said to be an ε-feasible solution to problem (P). If an ε-feasible solution $\bar{x} = (\bar{x}_0, \bar{x}_1, \dots, \bar{x}_{n+N})$ to problem (P) satisfies

$$\bar{x}_0 \leq \min\{x_0 \mid x \in D_\varepsilon\} + \eta,$$

$\bar{x}$ is called an (ε, η)-optimal solution to problem (P).

Remark 3.1 All feasible solutions to problem (P) are ε-feasible. When $\eta = 0$, an (ε, η)-optimal solution is optimal over all ε-feasible solutions of problem (P). When $\eta > 0$, an (ε, η)-optimal solution is η-optimal for all ε-feasible solutions to problem (P).

For seeking an (ε, η)-optimal solution of problem (P), a division and cut algorithm to be developed includes three essential operations: division operation, deleting operation, and reduction operation.

First, the division operation consists in a sequential box division of the original box D^0 following in an exhaustive subsection principle, such that any infinite nested sequence of division sets generated through the algorithm reduces to a singleton. This paper takes an adaptive division operation, which extends the standard bisection in the normally used exhaustive subsection principle. Second, by using overestimation of the constraints, the

deleting operation consists in eliminating each subbox D generated by the division operation, in which there is no feasible solution. In addition, the reduction operation is used to reduce the size of the current partition set (referred to a node), aiming at tightening each subbox which contains the feasible portion currently still of interest.

For any box $D = [p, q] = \prod_{i=0}^{n+N} [p_i, q_i] \subseteq D^0$, for convenience, we will use the following functions throughout this paper:

$$f_k^i(\alpha) = G_k^+(p) - G_k^-\big(q - \alpha(q_i - p_i)e^i\big), \quad k = 0, 1, \ldots, m + N,$$
$$g_k^i(\beta) = G_k^+\big(p' + \beta\big(q_i - p_i'\big)e^i\big) - G_k^-(q), \quad k = 0, 1, \ldots, m + N,$$

where $\alpha, \beta \in (0, 1), p_i \le p_i' \le q_i$, and e^i denotes the ith unit vector, i.e., a vector with 1 at the ith position and 0 everywhere else, $i = 0, 1, \ldots, n + N$.

At a given stage of the proposed algorithm for problem (P), let V represent the best current objective function value to problem (P). Next, we will show these detailed operations.

3.1 Deleting operation

In this subsection, we will give a suitable deleting operation, which offers a possibility to remove a subbox D of D^0 without feasibility. Toward this end, we take into account a subproblem of problem (P) over a given box $D = [p, q] = \prod_{i=0}^{n+N} [p_i, q_i] \subseteq D^0$ as follows:

$$\mathrm{P}(D): \begin{cases} \min & x_0, \\ \text{s.t.} & \bar{F}(x) \triangleq \max\{G_k^+(x) - G_k^-(x) \mid k = 0, 1, \ldots, m + N\} \le 0, \\ & x = (x_0, x_1, \ldots, x_{n+N}) \in D. \end{cases}$$

Given $\eta > 0$, for solving P(D), we need to seek out a feasible solution $\bar{x} = (\bar{x}_0, \bar{x}_1, \ldots, \bar{x}_{n+N}) \in D$ of P(D) such that $\bar{x}_0 \le V - \eta$, or to draw a conclusion that there exists no such $\bar{x}$, where V is the best objective value to problem P(D) known so far. Obviously, if $p_0 > V - \eta$, there exists no $\bar{x} \in D$ with $\bar{x}_0 \le V - \eta$. If $p_0 \le V - \eta$ and $G_k(p) \le 0$ for each $k = 0, \ldots, m + N$, then update $\bar{x} = p$. Therefore, without loss of generality, in the following discussion we shall suppose that

$$\bar{F}(p) > \varepsilon, \qquad p_0 \le V - \eta, \qquad \big\{x \in D^0 \mid \bar{F}(x) < \varepsilon\big\} \ne \emptyset. \tag{3.1}$$

Clearly, if $\bar{F}(x) > \varepsilon$ for any $x \in D$, there exists no feasible solution on D, and so D is deleted for further discussion. However, since the judgement satisfying $\bar{F}(x) > \varepsilon$ is not easy, we introduce an auxiliary problem of P(D) as follows:

$$\mathrm{Q}(D): \begin{cases} \min & \bar{F}(x) \\ \text{s.t.} & x_0 \le V - \eta, \\ & x = (x_0, x_1, \ldots, x_{n+N}) \in D \subseteq D^0. \end{cases}$$

Observe that the objective and constraint functions are interchanged in P(D) and Q(D). Let $V(\mathrm{P}(D))$ and $V(\mathrm{Q}(D))$ be the optimal values of problems P(D) and Q(D), respectively. Then we give the following results about problems P(D) and Q(D).

Theorem 3.1 *Let $\varepsilon > 0$, $\eta > 0$ be given, and let D be a box with $D \subseteq D^0$. We have the following result*:

(i) *A feasible solution to* $\mathrm{Q}(D)$ *satisfying* $\bar{F}(\hat{x}) < \varepsilon$ *is an* ε*-feasible solution of* $\mathrm{P}(D)$ *with* $\hat{x}_0 \leq V - \eta$.

(ii) *If* $V(\mathrm{Q}(D^0)) > 0$, *consider the following two cases*: (a) *problem* $P(D^0)$ *has no feasible solution if* $V = x_0^u + \eta$, *and* (b) *an* ε*-feasible solution* $\tilde{x} = (\tilde{x}_0, \tilde{x}_1, \ldots, \tilde{x}_{n+N})$ *of* $\mathrm{Q}(D^0)$ *is an* (ε, η)*-optimal solution of* $\mathrm{P}(D^0)$ *if* $V = \tilde{x}_0$.

Proof (i) This result is obvious, and here it is omitted.

(ii) By utilizing the assumption $V(\mathrm{Q}(D^0)) > 0$, i.e.,

$$\min\left\{\bar{F}(x) \mid x_0 \leq V - \eta, x \in D^0\right\} > 0, \tag{3.2}$$

we have the following conclusions:

(a) If $V = x_0^u + \eta$, by (3.2) we get $\{x \mid \bar{F}(x) \leq 0, x \in D^0\} = \emptyset$, which implies that problem $\mathrm{P}(D)$ has no feasible solution.

(b) If $V = \tilde{x}_0$, from (3.2) it is easy to see that

$$x_0 > V - \eta = \tilde{x}_0 - \eta,$$

for any $x = (x_0, x_1, \ldots, x_{n+N}) \in \{x \mid \bar{F}(x) \leq 0 < \varepsilon, x \in D^0\}$, which means that

$$\min\left\{x_0 \mid G_k(x) < \varepsilon, k = 0, 1, \ldots, m+N, x \in D^0\right\} \geq \tilde{x}_0 - \eta.$$

Consequently, $\tilde{x}$ is an (ε, η)-optimal solution of $\mathrm{P}(D)$, and this accomplishes the proof. □

Theorem 3.1 illustrates that by utilizing problem $\mathrm{Q}(D)$ one can know whether or not there exists a feasible solution $\hat{x} = (\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_{n+N})$ of $\mathrm{P}(D)$ with improving the current objective function value V, i.e., $\hat{x}_0 \leq V - \eta$. Further, if $V(\mathrm{Q}(D^0)) \geq \varepsilon > 0$, then an (ε, η)-optimal solution of $\mathrm{P}(D^0)$ can be obtained, or it can be confirmed that problem $\mathrm{P}(D^0)$ has no feasible solution.

Additionally, if $V(\mathrm{Q}(D)) > \varepsilon$, it is easy to see that a fathomed box D cannot contain the feasible solutions for problem (P). Thus, D is excluded from further consideration by the search. Unfortunately, solving problem $\mathrm{Q}(D)$ may be as difficult as solving the original problem $\mathrm{P}(D)$. Hence, a lower bound $\mathrm{LB}(D)$ of $V(\mathrm{Q}(D))$ is required for eliminating the part of D^0 which does not contain the solution to problem (P). Clearly, if $\mathrm{LB}(D) \geq \varepsilon$, D is excluded from further consideration by the search. Since $G_k^+(x)$ and $G_k^-(x)$ are all increasing, an apparent lower bound is

$$\underline{\mathrm{LB}}(D) = \max_{k=0,1,\ldots,m+N} \left\{G_k^+(p) - G_k^-(q)\right\}.$$

Although quite straightforward, the bound is sufficient to confirm convergence of the algorithm, as we will see immediately. For strengthening the computational validity in solving problem $\mathrm{P}(D)$, some tighter lower bounds can be obtained by utilizing the following Theorem 3.2. Especially, what is more important is that such tighter lower bounds can be gained only by simple arithmetic computations, which is different from the ones in the usual branch and bound methods for solving convex or linear programs [5, 6, 11, 12, 14].

Theorem 3.2 *For any* $D = [p,q] = \prod_{i=0}^{n+N}[p_i,q_i] \subseteq D^0$, *under assumption* (3.1), *let*

$$\alpha = \begin{cases} \frac{q_0 - V + \eta}{q_0 - p_0}, & \text{if } q_0 > V - \eta, \\ 0, & \text{otherwise.} \end{cases}$$

Then a lower bound $\mathrm{LB}(D)$ *satisfying* $\mathrm{LB}(D) \le V(Q(D))$ *for problem* $\mathrm{Q}(D)$ *is given by*

$$\mathrm{LB}(D) = \min_{i=0,1,\ldots,n+N} \max_{k=0,1,\ldots,m+N} \{f_k^i(\alpha)\}.$$

Proof (i) If $\alpha = 0$, this result is obvious.

(ii) By the assumption it holds that $p_0 \le V - \eta < q_0$. Then we get $0 < \alpha < 1$, and there exists $\tilde{x} = q - \alpha(q-p)$ so that $\tilde{x}_0 = V - \eta$ with $\tilde{x}_0 = q_0 - \alpha(q_0 - p_0)$. Thus, for each $\hat{x} = q - \beta(q-p)$ with $\beta < \alpha$, it is easy to find that $\hat{x}_0 > \tilde{x}_0 = V - \eta$, that is, for each $x > \tilde{x}$, there exists $\hat{x} = q - \beta(q-p)$ with $\beta < \alpha$ such that $x \ge \hat{x}$, and so it holds that $x_0 \ge \hat{x}_0 > V - \eta$. Now, let us denote

$$\Omega^i = \{x \in [p,q] \mid p_i \le x_i \le \tilde{x}_i\},$$

then we can acquire that

$$\begin{aligned} \{x \in [p,q] \mid x_0 \le V - \eta\} &\subset [p,q] \setminus \{x \mid \tilde{x} < x \le q\} \\ &= [p,q] \setminus \bigcap_{i=0}^{n+N} \{x \in [p,q] \mid \tilde{x}_i < x_i\} \\ &= \bigcup_{i=0}^{n+N} \{x \in [p,q] \mid p_i \le x_i \le \tilde{x}_i\} \\ &= \bigcup_{i=0}^{n+N} \Omega^i. \end{aligned}$$

Let $\mathrm{LB}(\Omega^i) = \max_{k=0,1,\ldots,m+N}\{f_k^i(\alpha)\}$ and $\mathrm{LB}(D) = \min\{\mathrm{LB}(\Omega^i) \mid i = 0,1,\ldots,n+N\}$. Obviously, we get $\mathrm{LB}(\Omega^i) \le \min\{\bar{F}(x) \mid x \in \bigcup_{i=0}^{n+N} \Omega^i\}$,

$$\mathrm{LB}(D) \le \min\left\{\bar{F}(x) \,\middle|\, x \in \bigcup_{i=0}^{n+N} \Omega^i\right\} \le \min\{\bar{F}(x) \mid x_0 \le V - \eta, x \in [p,q]\}.$$

Thus, the proof is complete. □

Notice that $\mathrm{LB}(D)$ in Theorem 3.2 satisfies

$$\min\{\bar{F}(x) \mid x_0 \le V - \eta, x \in D^0\} \ge \mathrm{LB}(D) \ge \max_{k=0,1,\ldots,m+N} \{G_k^+(p) - G_k^-(q)\}. \tag{3.3}$$

3.2 Adaptive division

The division operation repeatedly subdivides an $(n+N+1)$-dimensional box D^0 into $(n+N+1)$-dimensional subboxes. This operation helps the algorithm confirm the position of a global optimal solution in D^0 for problem (P). Throughout this algorithm, we take a new adaptive subdivision principle as follows.

Adaptive subdivision For given $\eta > 0$, consider any box $D = [p, q] = \{x \in R^{n+N+1} | p_i \le x_i \le q_i, i = 0, 1, \dots, n+N\} \subseteq D^0$.

(i) If $q_0 > V - \eta$, then let $\alpha = \frac{q_0 - V + \eta}{q_0 - p_0}$; otherwise let $\alpha = 0$.

(ii) Denote $t = \operatorname{argmax}\{q_i - p_i \mid i = 0, 1, \dots, n+N\}$. Let $u_t = p_t$ and $v_t = q_t - \alpha(q_t - p_t)$. Set $\bar{x}_t = (u_t + v_t)/2$.

(iii) By using $\bar{x}_t$, let us denote

$$D_1 = \left\{x \in R^{n+N+1} \mid p_i \le x_i \le q_i, i \ne t, p_t \le x_t \le \bar{x}_t, i = 0, 1, \dots, n+N\right\},$$

and

$$D_2 = \left\{x \in R^{n+N+1} \mid p_i \le x_i \le q_i, i \ne t, \bar{x}_t \le x_t \le q_t, i = 0, 1, \dots, n+N\right\}.$$

Based on the above division operation, D is divided into two new boxes D_1 and D_2. Especially, when $\alpha = 0$, the adaptive subdivision simply reduces to the standard bisection. As we will see from numerical experiments in Sect. 5, the adaptive subdivision is superior to the standard bisection. Moreover, the subdivision can confirm the convergence of the algorithm, and we have the following results.

Theorem 3.3 *Suppose that the above adaptive division operation is infinite, then it generates a nested sequence* $\{D^{s_t}\}$ *of partition sets* $\{D^s\}$ *generated by the adaptive division operation, so that* $\mathrm{LB}(D^{s_t}) \to V(\mathrm{Q}(D^0))$ *as* $t \to +\infty$.

Proof By the adaptive division operation, for each box $D^s = [p^s, q^s] \subseteq D^0$, we can acquire the points $u^s, v^s(i) \in D^s$ $(i = 0, 1, \dots, n+N)$ satisfying

$$u^s = p^s, \qquad v^s(i) = q^s - \alpha^s\left(q_i^s - p_i^s\right)e^i \quad \text{with } \alpha^s = 0 \text{ or } \alpha^s = \frac{q_0^s - V + \eta}{q_0^s - p_0^s}.$$

From Theorem 3.2, we can obtain

$$\mathrm{LB}\left(D^s\right) = \min_{i=0,1,\dots,n+N} \max_{k=0,1,\dots,m+N} \left\{G_k^+\left(u^s\right) - G_k^-\left(v^s(i)\right)\right\}.$$

According to Ref. [18], this adaptive division ensures the existence of an infinite subsequence $\{s_t\}$ with $D^{s_t+1} \subseteq D^{s_t}$ and $\mathrm{LB}(D^{s_t}) \le V(Q(D^0))$ for each t, so that for each $i = 0, 1, \dots, n+N$,

$$v^{s_t}(i) - u^{s_t} \to 0, \quad \text{as } t \to +\infty,$$

$$\lim_{t \to +\infty} v^{s_t}(i) = \lim_{t \to +\infty} u^{s_t} = \hat{u} \in D^0.$$

Thus we can obtain that

$$\lim_{t \to +\infty} \mathrm{LB}\left(D^{s_t}\right) = \min_{i=0,1,\dots,n+N} \max_{k=0,1,\dots,m+N} \left\{G_k^+(\hat{u}) - G_k^-(\hat{u})\right\} = \bar{F}(\hat{u}).$$

In addition, by assumption (3.1), since $u_0^{s_t} = p_0^{s_t} \le V - \eta$, it holds that

$$\lim_{t \to +\infty} u_0^{s_t} = \hat{u}_0 \le V - \eta,$$

which implies that $\hat{u}$ is feasible to $\mathrm{Q}(D^0)$, then we get

$$\mathrm{LB}\big(D^{s_t}\big) \le \min\big\{\bar{F}(x) \mid u_0 \le V-\eta, u \in D^0\big\} \le \bar{F}(\hat{u}).$$

Consequently, the limitation $\mathrm{LB}(D^{s_t}) \to V(Q(D^0))$ $(t \to +\infty)$ holds and then we accomplish the proof. □

3.3 Reduction operation

At a given stage of the proposed algorithm for problem (P), let $D = [p,q] \subseteq D^0$ be a box generated by the division operation and still of interest. Clearly, the smaller this box D is, the closer the feasible solution will be to the (ε,η)-optimal solution to problem (P). Hence, to effectively tighten the variable bounds in a particular node, a valid range reduction strategy is introduced by overestimation of the constraints, and by applying the monotonic decompositions to problem (P). Based on the above discussion, for any box $D = [p,q] \subseteq D^0$ generated by the division operation and still of interest, we intend to identify whether or not the box D contains a feasible solution $\hat{x}$ of $\mathrm{P}(D)$ such that $\hat{x}_0 \le V-\eta$. Consequently, seeking such a point $\hat{x}$ can be confined to the set $\hat{D} \cap D$, where

$$\hat{D} = \big\{x \mid x_0 \le V-\eta, G_k^+(x) - G_k^-(x) < \varepsilon, k = 0,1,\dots,m+N\big\}.$$

The reduction operation is based on special cuts that exploit the monotonic structure of problem (P), and it aims at substituting the box $D = [p,q]$ with a smaller box D' without losing any valid point $x \in \hat{D} \cap D$, i.e.,

$$\hat{D} \cap D' = \hat{D} \cap D. \tag{3.4}$$

This will suppress the fast growth of the branching tree in the division operation for seeking the (ε,η)-optimal solution of problem (P).

For any $D = [p,q] = \prod_{i=0}^{n+N}[p_i,q_i] \subseteq D^0$ generated by the division operation, the box D' satisfying condition (3.4) is denoted by $R[p,q]$. To recognize how $R[p,q]$ is acquired, we first demand to know the parameters γ, α_k^i, and β_k^i computed for each $k = 0,1,\dots,m+N$, $i = 0,1,\dots,n+N$ by utilizing the following rules.

Rule (i): Given the box $[p,q] \subseteq D^0$, if $f_k^i(1) > \varepsilon$, let α_k^i be the solution to the equation $f_k^i(\alpha_k^i) = \varepsilon$ about the univariate α_k^i; otherwise let $\alpha_k^i = 1$.

Rule (ii): For given boxes $D = [p,q]$ and $D' = [\bar{p},q]$ with $D' \subseteq D \subseteq D^0$, if $g_k^i(1) > \varepsilon$, one can solve the univariate equation $g_k^i(\beta_k^i) = \varepsilon$ to obtain β_k^i; otherwise let $\beta_k^i = 1$. If $\bar{p}_0 < V-\eta < q_0$, then set $\gamma = \frac{V-\eta-\bar{p}_0}{q_0-\bar{p}_0}$; otherwise let $\gamma = 1$.

Notice that it is easy to get α_k^i and β_k^i, since the univariate functions $f_k^i(\lambda)$ and $g_k^i(\mu)$ are strictly monotonic in Rules (i) and (ii).

According to Rules (i) and (ii), let us denote

$$\begin{aligned} &\hat{\alpha}^i = \min_{k=0,1,\dots,m+N}\big\{\alpha_k^i\big\}, \\ &\hat{\beta}^i = \min_{k=0,1,\dots,m+N}\big\{\beta_k^i, \gamma\big\}, \quad i = 0,1,\dots,n+N, \end{aligned} \tag{3.5}$$

then $R[p,q]$ can be obtained by Theorems 3.4 and 3.5.

Theorem 3.4 *Given the box* $D = [p, q] = \prod_{i=0}^{n+N} [p_i, q_i] \subseteq D^0$, *it holds that*

(i) *If* $p_0 \leq V - \eta$ *and* $\bar{F}(p) < \varepsilon$, *then* $R[p, q] = [p, p]$, *and*

(ii) *If* $p_0 > V - \eta$ *or* $\max\{f_k^i(0) | k = 0, 1, \ldots, m + N\} > \varepsilon$ *holds for some* $i \in \{0, 1, \ldots, n + N\}$, *then* $R[p, q] = \emptyset$.

Proof (i) The proof of this result is easy, here it is omitted.

(ii) The former of the conclusion is apparent, we only need to give the proof of the latter.

If there exists some $i \in \{0, 1, \ldots, n + N\}$ so that $\max\{f_k^i(0) | k = 0, 1, \ldots, m + N\} > \varepsilon$, we can obtain

$$\bar{F}(x) > \max_{k=0,1,\ldots,m+N} \left\{ f_k^i(0) \right\} > \varepsilon \quad \text{for each } x \in [p, q].$$

Therefore, we can acquire $R[p, q] = \emptyset$, and this accomplishes the proof. □

Theorem 3.5 *For any* $D = [p, q] \subseteq D^0$, *under Rule* (i) *and* (3.5) *and the assumption*

$$p_0 \leq V - \eta \quad \text{and} \quad \max_{k=0,1,\ldots,m+N} \left\{ f_k^i(0) \right\} < \varepsilon \quad \text{for each } i = 0, 1, \ldots, n + N,$$

let $\hat{p} = q - \sum_{i=0}^{n+N} \hat{\alpha}^i (q_i - p_i) e^i$. *If the box* $[\hat{p}, q]$ *satisfies the assumption of Theorem* 3.4, *then* $R[p, q] = [\hat{p}, \hat{p}]$ *or* $R[p, q] = \emptyset$. *Otherwise,* $R[p, q] = [\hat{p}, \hat{q}]$, *where* $\hat{q} = \hat{p} - \sum_{i=0}^{n+N} \hat{\beta}^i (q_i - p_i) e^i$ *with respect to Rule* (ii) *and* (3.5).

Proof For any given $x = (x_0, \ldots, x_{n+N})^T \in [p, q]$, we first confirm that $x \geq \hat{p}$.

Assume that $x \ngeq \hat{p}$, that is to say, there exists some $i \in \{0, 1, \ldots, n + N\}$ such that

$$x_i < \hat{p}_i = q_i - \hat{\alpha}^i (q_i - p_i) \quad \text{i.e. } x_i = q_i - \alpha (q_i - p_i) \text{ with } \alpha > \hat{\alpha}^i. \tag{3.6}$$

Then we discuss as follows:

If $\hat{\alpha}^i = 1$, we can obtain $x_i < \hat{p}_i = q_i - \hat{\alpha}^i (q_i - p_i) = p_i$ from (3.6), contradicting $x \in [p, q]$, then $x \geq \hat{p}$.

If $\hat{\alpha}^i \in (0, 1)$, from Rule (i) and the definition of $\hat{\alpha}^i$, there must exist some k such that $f_k^i(\hat{\alpha}^i) = \varepsilon$, i.e.,

$$G_k^+(p) - G_k^- \left(q - \hat{\alpha}^i (q_i - p_i) e^i \right) = \varepsilon. \tag{3.7}$$

In addition, through Rule (i) and the definition of $G_k^-(x)$, it holds from (3.6) and (3.7) that

$$G_k^- \left(q - (q_i - x_i) e^i \right) = G_k^- \left(q - \alpha (q_i - p_i) e^i \right) < G_k^- \left(q - \hat{\alpha}^i (q_i - p_i) e^i \right) = G_k^+(p) - \varepsilon.$$

Consequently,

$$G_k^-(x) \leq G_k^- \left(q - (q_i - x_i) e^i \right) < G_k^+(p) - \varepsilon \leq G_k^+(x) - \varepsilon,$$

contradicting $G_k^+(x) - G_k^-(x) \leq \varepsilon$. According to the above discussions, we have demonstrated that $x \geq \hat{p}$, i.e., $x \in [\hat{p}, q]$. Next we will show that $x \leq \hat{q}$.

Assume that $x \not\leq \hat{q}$, then there exists some i such that

$$x_i > \hat{q}_i = \hat{p}_i + \hat{\beta}^i(q_i - \hat{p}_i), \quad \text{i.e. } x_i = \hat{p}_i + \beta(q_i - \hat{p}_i) \text{ with } \beta > \hat{\beta}^i. \tag{3.8}$$

We discuss as follows:

If $\hat{\beta}^i = 1$, from (3.8) we can acquire $x_i > \hat{q}_i = \hat{p}_i + \hat{\alpha}^i(q_i - \hat{p}_i) = q_i$, contradicting $x \in [p, q]$.

If $\hat{\beta}^i \in (0, 1)$, from Rules (i) and (ii) and the definition of $\hat{\beta}^i$, we can obtain $g_k^i(\hat{\beta}^i) = \varepsilon$, i.e.,

$$G_k^+\big(\hat{p}_i + \hat{\beta}^i(q_i - \hat{p}_i)e^i\big) - G_k^-(q) = \varepsilon, \tag{3.9}$$

or

$$\hat{p}_0 + \hat{\beta}^0(q_0 - \hat{p}_0) = V - \eta. \tag{3.10}$$

Suppose that (3.9) holds, due to Rule (ii) and the definition of $G_k^+(x)$, by (3.8) and (3.9) it follows that

$$G_k^+\big(\hat{p} + (x_i - \hat{p}_i)e^i\big) = G_k^+\big(\hat{p} + \beta(q_i - \hat{p}_i)e^i\big) > G_k^+\big(\hat{p} + \hat{\beta}^i(q_i - \hat{p}_i)e^i\big) = G_k^-(q) + \varepsilon;$$

therefore,

$$G_k^+(x) \geq G_k^+\big(\hat{p} + (x_i - \hat{p}_i)e^i\big) > G_k^-(q) + \varepsilon \geq G_k^-(x) + \varepsilon \quad \text{with } x_i = \hat{p}_i + \beta(q_i - \hat{p}_i),$$

which contradicts $G_k^+(x) - G_k^-(x) \leq \varepsilon$.

Suppose that (3.10) holds, from (3.8), (3.10), and Rule (ii), we can draw a conclusion that

$$x_0 = \hat{p}_0 + \beta(q_0 - \hat{p}_0) > \hat{p}_0 + \hat{\beta}^0(q_0 - \hat{p}_0) = V - \eta,$$

which contradicts $x_0 \leq V - \eta$. According to the above discussions, we can acquire $x \leq \hat{q}$, i.e., $x \in [\hat{p}, \hat{q}]$, and the proof is complete. □

From Theorem 3.5, by Rules (i) and (ii) the main computational effort for deriving $R[p, q]$ is to solve some univariate equations about the variables $\hat{\alpha}_k^i$ and $\hat{\beta}_k^i$, which is easy to solve, for example, by using the bisection approach. What is more, as seen below, the cost of main computation in the proposed algorithm is also to form $R[p, q]$.

4 Algorithm and its convergence

According to the above discussions, the proposed algorithm is shown as follows.

Algorithm statement

Step (0) Given tolerances $\varepsilon, \eta > 0$, if there is no known feasible solution at present, set $V = x_0^u$ with $D^0 = [x^l, x^u]$; otherwise let x^* be the best feasible solution of problem (P), and set $V = x_0^*$. Let $M_0 = \{D^0\}$, $N_0 = \emptyset, t = 0$.

Step (1) For each $D = [p, q] = \prod_{i=0}^{n+N}[p_i, q_i] \in M_t$, compute $R[p, q]$ according to Theorems 3.4 and 3.5. Then:

(a) If $R[p, q] = \emptyset$, eliminate D.

(b) If $R[p,q] = [p,p]$, discard D and update $x^* = p$, $V = p_0$.

(c) If $R[p,q] \neq \emptyset$, set $D = R[p,q]$ and compute the lower bound $\mathrm{LB}(D)$ by Theorem 3.5. If $\mathrm{LB}(D) > \varepsilon$, delete D.

Step (2) Denote $\bar{M}_t$ to be the collection of boxes that results from M_t after accomplishment of Step (1), and set $\bar{N}_t = N_t \cup \bar{M}_t$.

(a) If $\bar{N}_t = \emptyset$, stop: (i) if $V = x_0^u$, problem (P) is infeasible; (ii) if $V = x_0^*$, x^* is an (ε, η)-optimal solution of problem (P).

(b) If $\bar{N}_t \neq \emptyset$, Theorem 3.4 is applied to each $D \in \bar{N}_t$, then eliminate D and update $x^* = p$, $V = p_0$ if necessary.

Step (3) Denote $\bar{N}_t$ to be the collection of boxes after accomplishment of Step (2). Choose $D^t = [p^t, q^t] \in \operatorname{argmin}\{\mathrm{LB}(D) \mid D \in \bar{N}_t\}$, and denote $\mathrm{LB}_t = \mathrm{LB}(\bar{N}_t)$. If $\mathrm{LB}_t > 0$, then terminate: the conclusion is the same as situation (a) of Step (2); otherwise go to Step (4).

Step (4) If $q_0^t > V - \eta$, let $s^t = p^t + \alpha_t(q^t - p^t)$ with $\alpha_t = \frac{q_0 - V + \eta}{q_0 - p_0}$; otherwise set $s^t = p^t$ or $s^t = (p^t + q^t)/2$. If $\bar{F}(s^t) < \varepsilon$, update $x^* = s^t$, $V = s_0^t$.

Step (5) Divide D^t into two subboxes by the adaptive division, and set M_{t+1} to be the collection of these two subboxes of D^t. Let $N_{t+1} = \bar{N}_t \backslash \{D^t\}$. Set $t = t + 1$, and return to Step (1).

Remark By utilizing a local solver in Step (4) of the proposed algorithm, instead of evaluating one point, we may acquire a point with the better objective function value V, and the iteration count of the algorithm may be reduced. However, because the computational cost increases rapidly with the quality of the objective value V, the running time is not always decreasing. So a trade-off must be made, practically just one evaluating point as in the above algorithm is used. Moreover, to implement the algorithm, all that is required is the ability to solve univariate equations with monotonicity and to execute simple algebraic steps.

The convergence of the algorithm is shown by the following theorem.

Theorem 4.1 *For given tolerances $\varepsilon, \eta > 0$, the above algorithm always terminates after finitely many iterations, obtaining an (ε, η)-optimal solution of problem* (P), *or a demonstration that the problem is infeasible.*

Proof Since any feasible solution $x = (x_0, x_1, \ldots, x_{n+N})$ to problem (P) with $x_0 \leq V - \eta$ must exist in some box $D \in \bar{N}_t$, case (a) of Step (2) implies that there exists no such solution x. In addition, if $\mathrm{LB}_t > 0$ occurs in Step (3), then we can obtain $V(\mathrm{MQ}(D)) > 0$. Consequently, by Theorem 3.5 the conclusion is true if one of the following cases occurs:

$$\bar{N}_t = \emptyset \quad \text{or} \quad \mathrm{LB}_t > 0,$$

that is to say, for extremely large t Steps (4) and (5) cannot appear. It remains to illustrate that either $\bar{N}_t = \emptyset$ or $\mathrm{LB}_t > 0$ must take place for extremely large t.

By contradiction, assume that the algorithm is infinite. Because each appearance of Step (4) reduces the current best objective function value $V = x_0^*$ at least by $\eta > 0$ if $\bar{F}(s^t) < \varepsilon$, this conflicts with the fact that x_0 is bounded below if Step (4) takes place infinitely. In other words, $\bar{F}(s^t) < \varepsilon$ in Step (4) cannot occur infinitely often. Therefore, for all t extremely

large, x^* is unaltered, and $\bar{F}(s^t) \geq \varepsilon$ while $\mathrm{LB}_t \leq 0$, and then Step (5) must be performed infinitely. By the division operation of the algorithm, as $t \to +\infty$, $q^t - p^t \to 0$, we have

$$\lim_{t\to+\infty} p^t = \lim_{t\to+\infty} q^t = \lim_{t\to+\infty} s^t = \hat{x} \in D^0.$$

According to (3.5), it holds that

$$0 \geq \mathrm{LB}_t \geq \max_{k=0,1,\ldots,m+N} \left\{ G_k^+\left(p^t\right) - G_k^-\left(q^t\right) \right\},$$

thus, it follows that

$$0 \geq \lim_{t\to+\infty} \mathrm{LB}_t \geq \max_{k=0,1,\ldots,m+N} \left\{ G_k^+(\hat{x}) - G_k^-(\hat{x}) \right\} = \bar{F}(\hat{x}),$$

which conflicts with $\lim_{t\to+\infty} \bar{F}(s^t) = \bar{F}(\hat{x}) \geq \varepsilon > 0$. Consequently, the algorithm must be finite, and the proof is finished. □

5 Numerical experiments

In this section, to verify the performance of the proposed algorithm, we give the computational results of six test examples and randomly generated problems. All tests are carried out on an Intel 5 CPU 2.33 GHz with 2 GB memory microcomputer and the algorithm is encoded in MATLAB. The numerical results are shown in Tables 1–2 and Figs. 1–4 to illustrate the feasibility and validity of the proposed algorithm. We first describe several simple examples in order to compare them with Refs. [4, 16], and the corresponding computational results are illustrated in Table 1. Additionally, we choose other problems randomly generated to test further the proposed algorithm, and the numerical results are shown in Figs. 1–4 and Table 2. As seen in the proposed algorithm, the main computational effort is to solve some univariate and monotonic equations to obtain $\hat{\alpha}_k^i$ and $\hat{\beta}_k^i$ required in the numerical experiments.

The notation in Table 1 is as follows:

- Ref.: reference;

Table 1 Computational results for Examples 1–6

Ex.	Ref.	Solution	ε	η	Optimum	Iter.	$L_{\max}$	Time(s)
1	[ours]	(0.0000, 1.66666667, 0.0000)	10^{-3}	10^{-6}	3.710919	8	4	0.1830
	[4]	(0.0000, 0.625, 1.875)	10^{-4}		4.0000	58	18	2.968694
	[17]	(1.1111, 1.36577e–05, 1.35168e–05)	10^{-9}					
2	[ours]	(5.0000, 0.0000,0.0000)	10^{-3}	10^{-6}	2.861905	16	8	0.1250
	[4]	(0, 3.3333, 0)	10^{-4}		3.0029	80	64	8.566259
3	[ours]	(1.5000, 1.5000)	10^{-3}	10^{-6}	4.9125874	56	14	1.0870
	[4]	(3.0000, 4.0000)	10^{-4}		5	32	32	1.089285
	[17]	(3.0000, 4.0000)	10^{-6}					
4	[ours]	(1.1111, 0.0000, 0.0000)	10^{-2}	10^{-5}	–4.090703	185	55	3.2510
	[16]	(1.0715, 0, 0)	10^{-6}		–4.087412	17		
5	[ours]	(0.0000, 0.3333, 0.0000)	10^{-3}	10^{-6}	–3.0029	17	3	0.1290
	[16]	(0, 0.33329, 0)	10^{-6}		–3.000042	30		
6	[ours]	(1.0000, 0.0000)	10^{-3}	10^{-6}	1.428571	6	2	0.0470
	[16]	(1.0000, 0.0000)	10^{-6}		1.428571	10		

Table 2 Average performances of adaptive division and bisection when $n = 3$ and $\varepsilon = 0.05$

	N	600	800	1000	1200	1500
Adaptive division	CPU time (s)	305.709	479.106	563.286	754.851	1009.027
Bisection	CPU time (s)	386.527	573.421	769.683	971.858	1227.389

Figure 1 Average computational time in seconds when $n = 3$, $N = 50$ and $\varepsilon = 0.05$

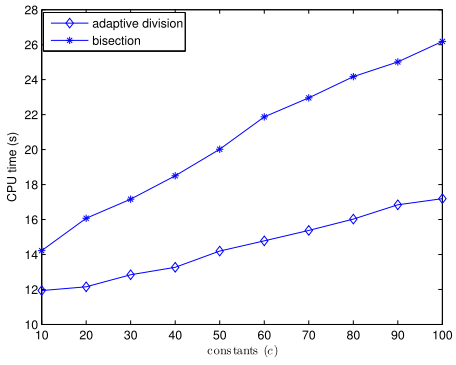


Figure 2 Average computational time in seconds when $n = 3$ and $\varepsilon = 0.05$

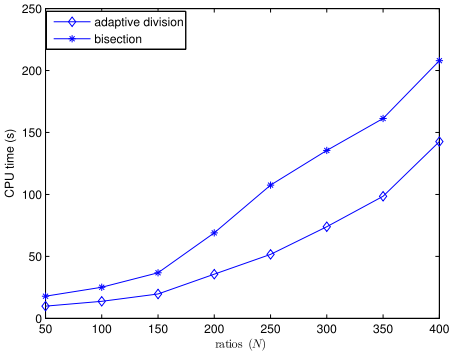
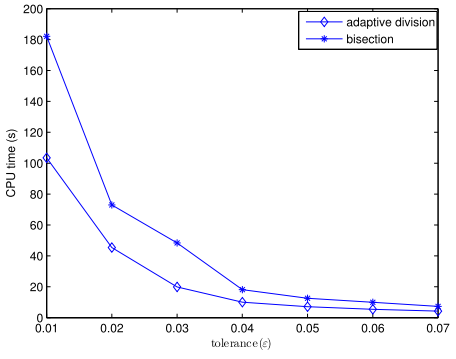


Figure 3 Average computational time in seconds when $n = 3$ and $N = 50$



- Iter.: the number of the algorithm iterations;
- Time (s): CPU seconds required for solving a problem;
- $L_{\max}$: the maximal number of the active node necessary.

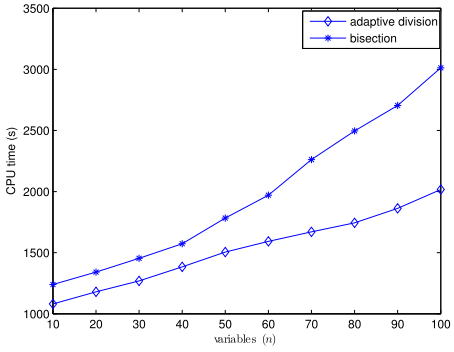


Figure 4 Average computational time in seconds when $N = 50$ and $\varepsilon = 0.05$

Example 1 (see [4, 17])

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

Example 2 (see [4])

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

Example 3 (see [4, 17])

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

Example 4 (see [16])

$$\begin{aligned}
\min\quad & -\left(\frac{4x_1+3x_2+3x_3+50}{3x_2+3x_3+50}+\frac{3x_1+4x_3+50}{4x_1+4x_2+5x_3+50}\right.\\
&\left.+\frac{x_1+2x_2+5x_3+50}{x_1+5x_2+5x_3+50}+\frac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50}\right)
\end{aligned}$$

$$\begin{aligned}
\text{s.t.}\quad & 2x_1 + x_2 + 5x_3 \le 10,\\
& x_1 + 6x_2 + 3x_3 \le 10,\\
& 5x_1 + 9x_2 + 2x_3 \le 10,\\
& 9x_1 + 7x_2 + 3x_3 \le 10,\\
& x_1, x_2, x_3 \ge 0.
\end{aligned}$$

Example 5 (see [16])

$$\begin{aligned}
\min\quad & -\left(\frac{3x_1 + 5x_2 + 3x_3 + 50}{3x_1 + 4x_2 + 5x_3 + 50} + \frac{3x_1 + 4x_2 + 50}{4x_1 + 3x_2 + 2x_3 + 50} + \frac{4x_1 + 2x_2 + 4x_3 + 50}{5x_1 + 4x_2 + 3x_3 + 50}\right)\\
\text{s.t.}\quad & 6x_1 + 3x_2 + 3x_3 \le 10,\\
& 10x_1 + 3x_2 + 8x_3 \le 10,\\
& x_1, x_2, x_3 \ge 0.
\end{aligned}$$

Example 6 (see [16])

$$\begin{aligned}
\min\quad & \frac{x_1 + 3x_2 + 2}{4x_1 + x_2 + 3} + \frac{4x_1 + 3x_2 + 1}{x_1 + x_2 + 4}\\
\text{s.t.}\quad & -(x_1 + x_2) \le -1,\\
& x_1, x_2 \ge 0.
\end{aligned}$$

From Table 1, we can obtain that solving all of the examples by the proposed algorithm yields the (ε, η)-optimal solutions with much better objective function values and being feasible. In addition, for Example 2, it is observed that the computational solution $x^* = (0, 3.3333, 0)$ of Ref. [4] does not satisfy the constraint $x_1 + 6x_2 + 2x_3 \le 10$, i.e., x^* is infeasible.

Next, we are particularly interested in instances of problem (FP) with a small number of variables and a lager number of ratios. The reason is that this case has many applications, especially in layered manufacturing and material layout [20–22]. In the following, we will show computational results of experimenting on the proposed algorithm for randomly generated problems, which are of the following form:

$$(\mathrm{P}): \begin{cases} \min & \sum_{i=1}^{N} \frac{c_i^\top y + c}{d_i^\top y + c} \\ \text{s.t.} & Ay \le b, y \ge 0, \end{cases}$$

where the elements of the matrix $A \in R^{m\times n}$, c_i, $d_i \in R^n$ $(i = 1, \ldots, N)$ are randomly generated in the unit interval $[0, 1]$ and $c \in R$. As for the subdivision operation, except for the proposed operation, we also test the usual bisection operation in branch and bound methods (e.g., [4, 5, 16–18, 23]) for comparison, where their respective computer programs were referred to as adaptive division and bisection. During running procedures, η is fixed on 10^{-5}, then we obtain a few data up and down; ultimately we choose the average result for running 10 times.

The comparison results between the adaptive division and bisection are shown in Figs. 1–4. Figure 1 illustrates the variedness of average computational time (in seconds) acquired by each item with c changing from 10 to 100 in 10 increments when $n = 3, N = 50$, and $\varepsilon = 0.05$, by which one can see that the influence of the value of c on computational time is slight, hence we let c be constant 3 in other experiments. Figure 2 gives the alteration of the average computational time (in seconds) gained by each term with N ranging from 50 to 400 when $(n, \varepsilon) = (3, 0.05)$, by which one can observe that when $N > 150$ the running time increases rapidly, so the effect of variedness of the number of ratios on computational time is biggish. Figure 3 demonstrates the change of the average computational time acquired by each item with ε ranging from 0.01 to 0.07 in 0.01 increments. It is obvious that this program is sensitive to changes in ε. From numerical results, we can see that the adaptive division acquires less time required by bisection for each program.

For the adaptive division and bisection the computational time of solving examples with larger N ranging from 600 to 1500 are listed in Table 2. We can see that even if it takes about seventeen minutes to solve problem (P) of size $(n, N) = (3, 1500)$, the adaptive division takes less time required by bisection for each N, which confirms the feasibility and availability of the proposed algorithm. We can draw a conclusion that the algorithm has more than enough performance, at least for three dimensions.

How does the algorithm behave for instances with $n > 3$? Unfortunately, the performance of the adaptive division rapidly deteriorates with increasing n, as shown in Fig. 3. For the same set of instances as in Fig. 1, Fig. 4 shows the variation of average computational time obtained by each program with n varying from 10 to 100 when (N, ε) is fixed at (50, 0.05), from which one can know that even though the running time is large, but the impact of variedness of the number of variables is mild with increasing running time in a nearly linear mode with increasing N.

Based on the above results, it is easy to see that the adaptive division has many advantages over the usual bisection in the computation. Additionally, when n is not larger than 3, the algorithm can rapidly solve problems, even if N takes on values as high as 1500. On the other hand, when the number of variables is as high as 100, with increasing n, we need to solve more equations in reduction operation, therefore, it is reasonable that the computational time is larger. Moreover, the lower bound is acquired only by simple arithmetic computations, that is, by executing simple algebraic steps, which is different from the ones used in solving convex or linear programs in common branch and bound methods. Finally, the computation for obtaining $\hat{\alpha}_k^i$ and $\hat{\beta}_k^i$ is easy by solving the equations with univariate and monotonicity in reduction operation, which is also the main computational cost of the algorithm.

6 Results and discussion

In this article, a new division and reduction algorithm is proposed for globally solving problem (FP). First, the original problem (FP) is converted into an equivalent optimization problem (P), in which the objective function is a single variable and the constraint functions are the difference of two increasing functions. Second, several basic operations (i.e., division, deleting, and reduction) are presented for designing a more efficient algorithm to problem (P). Finally, the numerical computational results show the feasibility and efficiency of the proposed basic operations, compared with the usual branch and bound (BB) methods (e.g., [4, 5, 9, 10, 16, 17]). Additionally, as further work, we think the ideas in

this article can be extended to the sum of nonlinear ratios optimization problems; for example, the numerator and denominator of each ratio in the objective function to problem (FP) are replaced with a generalized polynomial function, respectively.

Acknowledgements
The authors are grateful to the responsible editor and the anonymous referees for their valuable comments and suggestions, which have greatly improved the earlier version of this paper. This paper is supported by the National Natural Science Foundation of China (11671122; 11171094), the Program for Innovative Research Team (in Science and Technology) in University of Henan Province (14IRTSTHN023).

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
PPS carried out the idea of this paper, the description of the division and reduction algorithm, and drafted the manuscript. TL carried out the numerical experiments of the algorithm. All authors read and approved the final manuscript.

Publisher's Note

Received: 30 January 2018 Accepted: 7 March 2018 Published online: 21 March 2018

References

1. Freund, R.W., Jarre, F.: Solving the sum-of-ratios problem by an interior-point method. J. Glob. Optim. **19**, 83–102 (2001)
2. Matsui, T.: NP-hardness of linear multiplicative programming and related problems. J. Glob. Optim. **9**, 113–119 (1996)
3. Jiao, H.W., Liu, S.Y.: A practicable branch and bound algorithm for sum of linear ratios problem. Eur. J. Oper. Res. **243**, 723–730 (2015)
4. Wang, C., Shen, P.: A global optimization algorithm for linear fractional programming. Appl. Math. Comput. **204**(1), 281–287 (2008)
5. Shen, P., Wang, C.: Global optimization for sum of linear ratios problem. Appl. Math. Comput. **176**, 219–229 (2006)
6. Lim, S., Zhu, J.: Integrated data envelopment analysis: global vs. local optimum. Eur. J. Oper. Res. **229**, 276–278 (2013)
7. Rao, M.R.: Cluster analysis and mathematical programming. J. Am. Stat. Assoc. **66**, 622–626 (1971)
8. Drezner, Z., Manes, R.P., Whinston, A.: Queueing-location problems on the plane. Nav. Res. Logist. **37**, 929–935 (1990)
9. Benson, H.: A simplicial branch and bound duality-bounds algorithm for the linear sum-of-ratios problem. Eur. J. Oper. Res. **182**, 597–611 (2007)
10. Carlsson, J.G., Shi, J.: A linear relaxation algorithm for solving the sum-of-linear-ratios problem with lower dimension. Oper. Res. Lett. **41**(4), 381–389 (2013)
11. Depetrini, D., Locatelli, M.: Approximation of linear fractional-multiplicative problems. Math. Program. **128**, 437–443 (2011)
12. Shen, P., Li, W., Liang, Y.: Branch-reduction-bound algorithm for linear sum-of-ratios fractional programs. Pac. J. Optim. **11**(1), 79–99 (2015)
13. Konno, H., Yajima, Y., Matsui, T.: Parametric simplex algorithms for solving a special class of nonconvex minimization problems. J. Glob. Optim. **1**, 65–81 (1991)
14. Konno, H., Yamashita, H.: Minimization of the sum and the product of several linear fractional functions. Nav. Res. Logist. **46**, 583–596 (1999)
15. Muu, L.D., Tam, B.T., Schaible, S.: Efficient algorithms for solving certain nonconvex programs dealing with the product of two affine fractional functions. J. Glob. Optim. **6**, 179–191 (1995)
16. Ji, Y., Zhang, K., Qu, S.: A deterministic global optimization algorithm. Appl. Math. Comput. **185**, 382–387 (2007)
17. Jiao, H., Liu, S.: A practicable branch and bound algorithm for sum of linear ratios problem. Eur. J. Oper. Res. **243**, 723–730 (2015)
18. Jiao, H., Wang, Z., Chen, Y.: Global optimization algorithm for sum of generalized polynomial ratios problem. Appl. Math. Model. **37**, 187–197 (2013)
19. Tuy, H.: Convex Analysis and Global Optimization. Kluwer, Dordrecht (1978)
20. Majhi, J., Janardan, R., Schwerdt, J., Smid, M., Gupta, P.: Minimizing support structures and trapped area in two-dimensional layered manufacturing. Comput. Geom. Theory Appl. **12**, 241–267 (1999)
21. Chen, D., Daescu, O., Hu, X., Wu, X., Xu, J.: Determining an optimal penetration among weighted regions in two and three dimensions. J. Comb. Optim. **5**, 59–79 (2001)
22. Chen, D., Daescu, O., Dai, Y., Katoh, N., Wu, X., Xu, J.: Efficient algorithms and implementations for optimizing the sum of linear fractional functions, with applications. J. Comb. Optim. **9**, 69–90 (2005)
23. 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**, 153–169 (2013)