Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 11, No. 1, pp. 79–99, January 2015

BRANCH-REDUCTION-BOUND ALGORITHM FOR LINEAR SUM-OF-RATIOS FRACTIONAL PROGRAMS*

Pei-Ping Shen†, Wei-Min Li and Yan-Chao Liang

Abstract: This article is to present a branch-reduction-bound algorithm for globally solving the linear sum-of-ratios fractional programs. Based on the algorithm provided by Kuno (2005, JGO), we develop the two new procedures: bound tightening (BT) and cone reduction (CR), and add them to the proposed algorithm. The convergence proof of the algorithm is provided, and the numerical results are reported to show that the computational efficiency can be improved obviously by using the procedures (BT and CR).

Key words: *global optimization, linear sum-of-ratios, cone reduction, bound tightening, branch-and-bound*

Mathematics Subject Classification: *90C26, 90C30, 90C32, 65K05*

1 Introduction

Consider the following maximization problem:

$$(\mathrm{P}) \quad \begin{cases} v = & \max \ h(x) = \sum\limits_{i=1}^{p} \dfrac{d_i^\top x + \delta_i}{c_i^\top x + \gamma_i} \\ & \text{s.t.} \quad X = \{x \in R^n | Ax \le b, \ x \ge 0\}, \end{cases}$$

where $A \in R^{m\times n}, b \in R^m, c_i, d_i \in R^n$, and $\gamma_i, \delta_i \in R$ for $i = 1, 2, \ldots, p$. Assume that the feasible region X is nonempty and bounded, and $c_i^\top x + \gamma_i \neq 0$, for all $x \in X$.

The problem (P) is a special class of optimization problem among fractional programs, which has attracted the interests of a growing number of researchers. This is at least in part because from a practical point of view, this problem has a variety of applications. Included among these, for example, are multistage shipping problems [1], certain government contracting problems [5], problems in cluster analysis [18], certain queueing location problems [6], bond portfolio optimization problems [14] and a number of problems in geometric optimization [4]. From a research point of view, the problem (P) poses significant theoretical and computational challenges since the objective function is neither quasi-convex nor quasi-concave. It is well-known that problem (P) is NP-hard [17] and that is a multi-extremal problem, i.e., generally possess multiple local optima that are not globally optimal [19].

Many algorithms have been proposed for solving special cases of problem (P), which are limited to the case where all of the numerators $d_i^\top x + \delta_i$ and the denominators $c_i^\top x +$

*Research supported by the National Natural Science Foundation of China (11171094) and by Program for Innovative Research Team (in Science and Technology) in University of Henan Province (14IRTSTHN023).

†Corresponding author.

γ_i are positive over the feasible region X. For instance, when $p = 2$, Konno et al. [15] use a parametric simplex method to find globally optimal solution to such problem; when $p \geq 2$, several algorithms [7, 11–13, 16] have been developed. The algorithm of Konno and Yamashita [16] solves subsequently an equivalent concave minimization problem by outer approximation; Konno and Fukaish [13] present a solution algorithm that involves subproblems with both linear and quadratic constraints; Falk and Palocsay [7] propose iteratively searching the image space of the problem to find a global solution; the other algorithms use the branch-and-bound search [3, 8, 10–12, 23], for instance, Kuno [11, 12] utilizes trapezoidal partition elements, where linear programming subproblems over these trapezoids are respectively solved to derive tight upper bounds during the branch-and-bound search. In addition, under the assumptions that $d_i^\top x + \delta_i \geq 0$ and $c_i^\top x + \gamma_i \neq 0$, Ji et al. [10] present a branch and bound algorithm. For a more general case, i.e., the problem (P) does not impose any sign restrictions on $d_i^\top x + \delta_i$ and $c_i^\top x + \gamma_i$, the corresponding solution algorithms have been still rare in the literature as far though there exists several methods (see [3, 21, 22]). For an excellent review of the applications, theory, and algorithms for the sum-of ratios fractional program, the reader can refer to [2, 9, 20].

The main purpose of this article is to present an improvement of the algorithm in Ref. [12] for globally solving problem (P) by using the proposed bound tightening (BT) and cone reduction (CR) procedures. In contrast to the usual branch and bound methods reviewed above (e.g. [3,8,10–12,21–23]), in the proposed branch-reduction-bound algorithm, an upper bound of the subproblem in each node is obtained by utilizing the proposed tight bound procedure, specially, the bound obtained is tighter than the one in [12]. Also, the cone reduction that doesn't appear in other branch and bound methods (e.g. [3,8,10–12,21–23]), can cut away a large part of the region in which the optimal solution does not exist, so that the rapid growth of the branching tree can be suppressed during the algorithm search. Additionally, the problem investigated in this paper generalizes those in [7, 11–13, 15, 16], and the numerical results show that the computational efficiency can be improved obviously by using BT and CR.

The remainder of this paper is organized as follows. The next section converts problem (P) into an equivalent problem P(Θ^0), further, the linear programming problem LP(Θ) is obtained to derive the upper bound of the optimal value. In Section 3, the algorithm is presented and its convergence is shown. In Section 4, we report some numerical results obtained by solving some examples. Finally, some concluding remarks are given in Section 5.

2 Global Optimization of the Problem

For solving problem (P), we first notice that

$$h(x) = \sum_{i\in I_+} \frac{d^i x + \delta_i}{c^i x + \gamma_i} + \sum_{i\in I_-} \frac{d^i x + \delta_i}{c^i x + \gamma_i} = \sum_{i\in I_+} \frac{d^i x + \delta_i}{c^i x + \gamma_i} + \sum_{i\in I_-} \frac{-(d^i x + \delta_i)}{-(c^i x + \gamma_i)}$$

where $I_+ = \{i \in \{1, 2, \ldots, p\} | c^i x + \gamma_i > 0, \forall x \in X\}$, $I_- = \{i \in \{1, 2, \ldots, p\} | c^i x + \gamma_i < 0, \forall x \in X\}$. Obviously, denominators in $h(x)$ are all positive. Hence, we can always assume that $c^i x + \gamma_i > 0$ always holds. In addition, under the above assumption $c^i x + \gamma_i > 0$, let us denote $J_+ = \{i \in \{1, 2, \ldots, p\} | d^i x + \delta_i > 0, \forall x \in X\}$, and $J_- = \{i \in \{1, 2, \ldots, p\} | d^i x + \delta_i <$

$0, \forall x \in X\}$. Thus, we have

$$\begin{aligned} h(x) &= \sum_{i\in J_+} \frac{d^i x+\delta_i}{c^i x+\gamma_i} + \sum_{i\in J_-} \frac{d^i x+\delta_i}{c^i x+\gamma_i} \\ &= \sum_{i\in J_+} \frac{d^i x+\delta_i}{c^i x+\gamma_i} + \sum_{i\in J_-} \frac{d^i x+\delta_i + m_i(c^i x+\gamma_i)}{c^i x+\gamma_i} - \sum_{i\in J_-} m_i \end{aligned}$$

by choosing a sufficiently large value $m_i > 0$ $(i \in J_-)$ such that $d^i x+\delta_i+m_i(c^i x+\gamma_i) > 0$ for any $x \in X$, $i \in J_-$. Therefore, throughout this paper, we can assume without loss generality that

$$d^i x + \delta_i > 0, \quad c^i x + \gamma_i > 0, \quad \forall x \in X, \quad i = 1, 2, \ldots, p.$$

The principal structure in the development of a solution procedure for solving problem (P) is the construction of upper bounds for this problem, as well as for its partitioned subproblems. An upper bound on the solution of problem (P) and its partitioned subproblems can be obtained by solving a relaxation linear programming of problem (P). Toward this end, l_i, u_i, I_i^0 and U_i^0 for each i will be introduced as follows:

$$\begin{aligned} &0 < l_i = \min\{d_i^\top x + c_i^\top x | x \in X\} + \delta_i + \gamma_i, \\ &\infty > u_i = \max\{d_i^\top x + c_i^\top x | x \in X\} + \delta_i + \gamma_i > 0, \\ &0 \le L_i^0 \le \frac{d_i^\top x + \delta_i}{c_i^\top x + \gamma_i} \le U_i^0 < \infty, \quad \forall x \in X. \end{aligned}$$

Note that L_i^0, U_i^0 can be known by solving linear programming problems. Additionally, several sets are defined by

$$\begin{aligned} &\Omega_i = \{(t_i, s_i) \in R^2 | t_i = d_i^\top x + \delta_i,\ s_i = c_i^\top x + \gamma_i,\ x \in X\}, \\ &\Sigma_i(l_i, u_i) = \{(t_i, s_i) \in R^2 | 0 < l_i \le t_i + s_i \le u_i\}, \\ &\Theta_i^0(L_i^0, U_i^0) = \{(t_i, s_i) \in R^2 | 0 \le L_i^0 s_i \le t_i \le U_i^0 s_i\}, \\ &F_i = \Omega_i \cap \Sigma_i(l_i, u_i)_i \cap \Theta_i^0(L_i^0, U_i^0). \end{aligned}$$

By using the above notation, based on Ref. [12] problem (P) can be rewritten as:

$$\mathrm{P}(\Theta^0) \quad \left\{ \begin{array}{lll} v(\Theta^0) = & \max & \sum\limits_{i=1}^{p} \dfrac{t_i}{s_i} \\ & \text{s.t.} & (t_i, s_i) \in F_i, \quad i = 1, 2, \ldots, p, \end{array} \right.$$

where $\Theta^0 \triangleq \Theta^0(L^0, U^0) = \prod\limits_{i=1}^{p} \Theta_i^0(L_i^0, U_i^0)$, $L^0 = (L_i^0)_{p\times 1} \in R^p$, $U^0 = (U_i^0)_{p\times 1} \in R^p$.

2.1 Upper bound

Throughout this article, let Θ be any $2p$-dimensional cone with

$$\Theta(L, U) = \prod_{i=1}^{p} \Theta_i(L_i, U_i)$$

satisfying $L_i^0 \le L_i \le U_i \le U_i^0$, i.e. Θ denotes Θ^0 or a sub-cone of Θ^0, where $L = (L_i)_{p\times 1} \in R^p$, $U = (U_i)_{p\times 1} \in R^p$.

For solving problem $\mathrm{P}(\Theta^0)$, the branch and bound algorithm to be proposed will require an upper bound $UB1(\Theta)$ of the optimal value $v(\Theta)$ to a subproblem $\mathrm{P}(\Theta)$ of $\mathrm{P}(\Theta^0)$ restricted to Θ. To help explain how $UB1(\Theta)$ is formed, we need to solve a linear relaxation

programming problem of $\mathrm{P}(\Theta)$, based on the concave envelope $\Phi_i(t_i, s_i)$ of each ratio $\frac{t_i}{s_i}$ of the objective function to $\mathrm{P}(\Theta)$. As is given in Ref. [12], we have

$$\Phi_i(t_i, s_i) = \min\{f_i(t_i, s_i), g_i(t_i, s_i)\} \geq \frac{t_i}{s_i}, \quad \text{if } (t_i, s_i) \in \Sigma_i(l_i, u_i) \cap \Theta_i(l_i, u_i) \tag{2.1}$$

where

$$f_i(t_i, s_i) = \frac{U_i + 1}{l_i}(t_i - L_i s_i) + L_i,$$
$$g_i(t_i, s_i) = \frac{L_i + 1}{u_i}(t_i - U_i s_i) + U_i.$$

Clearly, $\Phi_i(t_i, s_i)$ is an over-estimator of $\dfrac{t_i}{s_i}$ over the trapezoid $\Sigma_i(l_i, u_i) \cap \Theta_i(L_i, U_i)$. Instead of maximizing $\sum\limits_{i=1}^{p} \dfrac{t_i}{s_i}$ in problem $\mathrm{P}(\Theta)$, we can obtain an upper bound $UB1(\Theta)$ of the optimal value $v(\Theta)$ to problem $\mathrm{P}(\Theta)$ by solving the following problem:

$$\mathrm{P}_1(\Theta) \quad \begin{cases} UB1(\Theta) = & \max \sum\limits_{i=1}^{p} \Phi_i(t_i, s_i) \\ & \text{s.t.} \quad (t_i, s_i) \in F_i, \quad i = 1, 2, \ldots, p. \end{cases}$$

It is easily seen that problem $\mathrm{P}_1(\Theta)$ is equivalent to

$$\mathrm{P}_2(\Theta) \quad \begin{cases} UB1(\Theta) = & \max \sum\limits_{i=1}^{p} r_i \\ & \text{s.t.} \quad (t_i, s_i) \in F_i, \quad i = 1, 2, \ldots, p, \\ & \qquad r_i \leq f_i(t_i, s_i), \ i = 1, \ldots, p, \\ & \qquad r_i \leq g_i(t_i, s_i), \ i = 1, \ldots, p. \end{cases}$$

Consequently, $\mathrm{P}_2(\Theta)$ can be rewritten as the following linear programming problem:

$$\mathrm{LP}(\Theta) \quad \begin{cases} UB1(\Theta) = & \max \sum\limits_{i=1}^{p} r_i \\ & \text{s.t.} \quad Ax \leq b, x \geq 0, \\ & \qquad (U_i + 1)(L_i c_i^\top - d_i^\top)x + l_i r_i \leq \alpha_i, i = 1, 2, \ldots, p, \\ & \qquad (L_i + 1)(U_i c_i^\top - d_i^\top)x + u_i r_i \leq \beta_i, i = 1, 2, \ldots, p, \\ & \qquad L_i \leq \dfrac{d_i^\top x + \delta_i}{c_i^\top x + \gamma_i} \leq U_i, i = 1, 2, \ldots, p, \\ & \qquad l_i \leq (d_i^\top + c_i^\top)x + \delta_i + \gamma_i \leq u_i, i = 1, 2, \ldots, p, \end{cases}$$

where

$$\alpha_i = (U_i + 1)(\delta_i - L_i \gamma_i) + l_i L_i,$$
$$\beta_i = (L_i + 1)(\delta_i - U_i \gamma_i) + u_i U_i.$$

Thus, we can obtain the following conclusion.

Theorem 2.1. *Problem* $\mathrm{LP}(\Theta)$ *is equivalent to problem* $\mathrm{P}_1(\Theta)$ *in the following sense: if* $\mathrm{LP}(\Theta)$ *is infeasible, then let* $UB1(\Theta) = -\infty$*; otherwise, for any optimal solution* $(\hat{x}, \hat{r})$ *to* $\mathrm{LP}(\Theta)$*, we can obtain the optimal solution* $(\hat{x}, \hat{t}, \hat{s})$ *to problem* $\mathrm{P}_1(\Theta)$ *with* $\hat{t}_i = d_i^\top \hat{x} + \delta_i$*,* $\hat{s}_i = c_i^\top \hat{x} + \gamma_i, i = 1, \ldots, p$*, and* $UB1(\Theta) = \sum\limits_{i=1}^{p} \hat{r}_i$ *provides an upper bound of the optimal value* $v(\Theta)$ *to problem* $\mathrm{P}(\Theta)$*.*

Proof. The proof of this result is straightforward and therefore is omitted. □

From the above discussion, although $UB1(\Theta)$ is an upper bound of $v(\Theta)$ to problem $\mathrm{P}(\Theta)$, to improve the computation efficiency of the branch and bound algorithm for solving problem $\mathrm{P}(\Theta^0)$, we can get a tighter upper bound $UB(\Theta)$ than $UB1(\Theta)$ by a new bounding tightening (BT) given as follows.

Next, we will describe the proposed BT (see Theorem 2.2 below). For this purpose, let $(\hat{x}, \hat{r})$ be an optimal solution to the linear program $\mathrm{LP}(\Theta)$, and let $\hat{t}_i = d_i^\top \hat{x} + \delta_i, \hat{s}_i = c_i^\top \hat{x} + \gamma_i, i = 1, 2, \ldots, p$. We see that the value of $\hat{r}_i$ depends on neither L_i nor U_i directly, but on $\Phi_i(\hat{t}_i, \hat{s}_i)$. Let us try improving this upper bound $\Phi_i(\hat{t}_i, \hat{s}_i)$ on $\dfrac{\hat{t}_i}{\hat{s}_i}$ by exploiting a relationship between $\dfrac{t_i}{s_i}$ and its over-estimator Φ_i.

Theorem 2.2. *For any cone $\Theta \subseteq \Theta^0$, let $(\hat{x}, \hat{r})$ and $UB(\Theta)$ be the optimal solution and the optimal value to problem $\mathrm{LP}(\Theta)$, and let $\hat{t}_i = d_i^\top \hat{x} + \delta_i, \hat{s}_i = c_i^\top \hat{x} + \gamma_i$ for each i. Then, by choosing $\sigma \in (0, 1]$, an upper bound $UB(\Theta)$ of $v(\Theta)$ to problem $\mathrm{P}(\Theta)$ can be given by*

$$UB(\Theta) = \sum_{i=1}^{p} \frac{(\sigma L_i + \sigma U_i + 1)\hat{t}_i \hat{s}_i + (1 - \sigma)\hat{t}_i^2 - \sigma L_i U_i \hat{s}_i^2}{\hat{s}_i(\hat{t}_i + \hat{s}_i)}$$

satisfying

$$UB1(\Theta) \geq UB(\Theta) \geq v(\Theta),$$

especially, the inequalities hold strictly if $(\hat{t}_i, \hat{s}_i) \in int(\Sigma_i \cap \Theta_i)$ for each i.

Proof. Denote the intersection points of $t_i = (\hat{t}_i/\hat{s}_i)s_i$ with $t_i + s_i = l_i$ and $t_i + s_i = u_i$, respectively, by

$$\xi_i = \frac{l_i}{\hat{t}_i + \hat{s}_i}(\hat{t}_i, \hat{s}_i), \qquad \eta_i = \frac{u_i}{\hat{t}_i + \hat{s}_i}(\hat{t}_i, \hat{s}_i).$$

Since the value of $\dfrac{t_i}{s_i}$ is constant along the half line defined by $t_i = (\hat{t}_i/\hat{s}_i)s_i$, it follows that both the values of $\dfrac{t_i}{s_i}$ at ξ_i and η_i are $\dfrac{\hat{t}_i}{\hat{s}_i}$, i.e.

$$\frac{t_i}{s_i}\Big|_{\xi_i} = \frac{t_i}{s_i}\Big|_{\eta_i} = \frac{\hat{t}_i}{\hat{s}_i}.$$

Further, from (2.1) it follows that

$$\begin{aligned} \xi_i &\in \Sigma_i(l_i, u_i) \cap \{(t_i, s_i) | f_i(t_i, s_i) \leq g_i(t_i, s_i)\}, \\ \eta_i &\in \Sigma_i(l_i, u_i) \cap \{(t_i, s_i) | f_i(t_i, s_i) \geq g_i(t_i, s_i)\}, \end{aligned}$$

and so we have

$$\Phi_i(\xi_i) = f_i(\xi_i) = \frac{(U_i + L_i + 1)\hat{t}_i - U_i L_i \hat{s}_i}{\hat{t}_i + \hat{s}_i} = g_i(\eta_i) = \Phi_i(\eta_i). \tag{2.2}$$

Since $\Phi_i(t_i, s_i)$ is the concave envelope of $\dfrac{t_i}{s_i}$ on $\Theta_i \cap \Sigma_i$, we can obtain

$$\left.\begin{aligned} \Phi_i(\xi_i) &\geq \frac{\hat{t}_i}{\hat{s}_i}, \\ \Phi_i(\eta_i) &\geq \frac{\hat{t}_i}{\hat{s}_i}. \end{aligned}\right\} \tag{2.3}$$

On the other hand, since $l_i \leq \hat{t}_i + \hat{s}_i \leq u_i$, there exists

$$\mu = \frac{\hat{t}_i + \hat{s}_i - u_i}{l_i - u_i} \quad \text{with } \mu \in [0,1]$$

such that $(\hat{t}_i, \hat{s}_i) = \mu\xi_i + (1-\mu)\eta_i$. Thus, by (2.2) and the concavity of $\Phi_i(\hat{t}_i, \hat{s}_i)$ it follows that

$$\Phi_i(\hat{t}_i, \hat{s}_i) = \Phi_i(\mu\xi_i + (1-\mu)\eta_i) \geq \mu\Phi_i(\xi_i) + (1-\mu)\Phi_i(\eta_i) = \Phi_i(\xi_i). \tag{2.4}$$

Hence, from (2.2)-(2.4), it follows that

$$\Phi_i(\hat{t}_i, \hat{s}_i) \geq \Phi_i(\xi_i) = \frac{(U_i + L_i + 1)\hat{t}_i - U_i L_i \hat{s}_i}{\hat{t}_i + \hat{s}_i} \geq \frac{\hat{t}_i}{\hat{s}_i}. \tag{2.5}$$

Furthermore, by using the above inequality (2.5) and choosing a scalar $\sigma \in (0,1]$ we see that there exists a nonnegative constant $\pi_i(\sigma)$, given by

$$\pi_i(\sigma) = (1-\sigma)\frac{(U_i\hat{s}_i - \hat{t}_i)(\hat{t}_i - L_i\hat{s}_i)}{\hat{s}_i(\hat{t}_i + \hat{s}_i)} \geq 0$$

such that

$$\Phi_i(\hat{t}_i, \hat{s}_i) \geq \Phi_i(\xi_i) \geq \Phi_i(\xi_i) - \pi_i(\sigma) \geq \frac{\hat{t}_i}{\hat{s}_i}. \tag{2.6}$$

Notice that the following relation holds

$$\Phi_i(\xi_i) - \pi_i(\sigma) - \frac{\hat{t}_i}{\hat{s}_i} = \sigma\frac{(U_i\hat{s}_i - \hat{t}_i)(\hat{t}_i - L_i\hat{s}_i)}{\hat{s}_i(\hat{t}_i + \hat{s}_i)} \geq 0. \tag{2.7}$$

Hence, we can see easily from (2.3), (2.6) and (2.7) that an improving upper bound $\Psi_i(\hat{t}_i, \hat{s}_i; \sigma)$ on $\dfrac{\hat{t}_i}{\hat{s}_i}$ is given by

$$\Psi_i(\hat{t}_i, \hat{s}_i; \sigma) = \Phi_i(\xi_i) - \pi_i(\sigma) = \frac{(\sigma L_i + \sigma U_i + 1)\hat{t}_i\hat{s}_i + (1-\sigma)\hat{t}_i^2 - \sigma L_i U_i \hat{s}_i^2}{\hat{s}_i(\hat{t}_i + \hat{s}_i)} \tag{2.8}$$

with $\sigma \in (0,1]$ satisfying

$$\Phi_i(\hat{t}_i, \hat{s}_i) \geq \Psi_i(\hat{t}_i, \hat{s}_i; \sigma) \geq \frac{\hat{t}_i}{\hat{s}_i}. \tag{2.9}$$

Thus, by problems P(Θ) and LP(Θ), the inequalities

$$UB1(\Theta) \geq UB(\Theta) \geq v(\Theta)$$

can be derived from (2.8) and (2.9). Especially, when $(\hat{t}_i, \hat{s}_i) \in \text{int}(\Sigma_i \cap \Theta_i)$, these inequalities throughout in the above proof hold strictly, and so the result is obvious. □

From the proof of the above theorem, we know that $\Phi_i(\xi_i)$ or $\Phi_i(\eta_i)$ can be also served as an improving upper bound on $\dfrac{\hat{t}_i}{\hat{s}_i}$ from (2.5), as Kuno given in Ref. [12]. However, according to (2.6) and (2.8), $\Psi_i(\hat{t}_i, \hat{s}_i; \sigma)$ is more tightening than $\Phi_i(\xi_i)$ as an upper bound

of $\frac{\hat{t}_i}{\hat{s}_i}$. In fact, if we choose $\sigma = 1$ then $\Psi_i(\hat{t}_i, \hat{s}_i; 1) = \Phi_i(\xi_i)$, specially, it is easily seen from (7) that $\Psi_i(\hat{t}_i, \hat{s}_i; \sigma)$ is more tightening about $\frac{t_i}{s_i}$ when σ is closer to zero. Hence, if we replace $\hat{r}_i = \Phi_i(\hat{t}_i, \hat{s}_i)$ by $\bar{r}_i = \Psi_i(\hat{t}_i, \hat{s}_i; \sigma)$, a new upper bound $UB(\Theta) = \sum_{i=1}^{p} \bar{r}_i$ will improve $\sum_{i=1}^{p} \Phi_i(\hat{t}_i, \hat{s}_i) = \sum_{i=1}^{p} \hat{r}_i$ and this will suppress the rapid growth of branching trees. The computational results also confirm this conclusion in Section 4 below. □

Based on Theorem 2.2 we can give the proposed (BT) as follows. Let $\hat{t}_i = d^i\hat{x}+\delta_i$ and $\hat{s}_i = c^i\hat{x}+\gamma_i$ for all $i = 1, \ldots, p$, and we replace $\hat{r}_i$ by $\bar{r}_i = \Psi_i(\hat{t}_i, \hat{s}_i; \sigma)$ for each $i = 1, \ldots, p$. Then a new tightening upper bound $UB(\Theta)$ of $v(\Theta)$ is given by

$$UB(\Theta) = \sum_{i=1}^{p} \bar{r}_i = \sum_{i=1}^{p} \Psi_i(\hat{t}_i, \hat{s}_i; \sigma)$$

satisfying $v(\Theta) \leq UB(\Theta) \leq UB1(\Theta)$.

2.2 Cone reduction

In this subsection, we pay our attention on how to form a new cone reduction procedure for eliminating a region in which the global minimum of $\mathrm{P}(\Theta^0)$ does not exist. The main purpose of the cone reduction is to decrease the size of the cone Θ^0 or its partitioned sub-cone Θ without losing any current feasible solution still of interest. It can be regarded as an accelerating device of branch-and-bound algorithm for solving problem $\mathrm{P}(\Theta^0)$ from the numerical results in Section 4 below. To see how to form this procedure, let LB_{best} be the best lower bound of the optimal value to $\mathrm{P}(\Theta^0)$ so far, and for a $2p$-dimensional sub-cone Θ, denote

$$RU = \sum_{i=1}^{p} U_i,$$

$$\rho_j = LB_{best} - \sum_{i=1, i\neq j}^{p} U_i, \quad j = 1, 2, \ldots, p.$$

Next, we give the following Theorems 2.3 and 2.4, and the cone reduction will be generated via these theorems.

Theorem 2.3. *Assume that $\Theta(L, U) = \prod_{i=1}^{p} \Theta_i(L_i, U_i)$ is any sub-cone of Θ^0. If $RU < LB_{best}$, then there exists no the optimal solution to problem $P(\Theta^0)$ over $\Theta(L, U)$. Otherwise, if there exists some index $h \in \{1, \ldots, p\}$ satisfying $\rho_h > L_h$, then there is no the optimal solution to problem $P(\Theta^0)$ over $\Theta_a(L, U) = \prod_{i=1}^{p} \Theta_{ai}(L_i, U_i)$, where*

$$\Theta_{ai}(L_i, U_i) = \begin{cases} \Theta_i(L_i, U_i), & \text{if } i \neq h \\ \Delta_{ah}(L_h, \rho_h) \cap \Theta_h(L_h, U_h), & \text{if } i = h \end{cases}$$

with $\Delta_{ah}(L_h, \rho_h) = \{(t_h, s_h) \in R^2 | L_h s_h \leq t_h < \rho_h s_h\}$.

Proof. If $RU < LB_{best}$, then, for any $(t_i, s_i) \in \Theta_i(L_i, U_i), i = 1, 2, \ldots, p$, from problem $\mathrm{P}(\Theta)$, it follows that

$$v(\Theta) = \max_{\forall i,\ (t_i,s_i)\in F_i} \sum_{i=1}^{p} \frac{t_i}{s_i} \le \sum_{i=1}^{p} U_i = RU < LB_{best},$$

and so, there exists no the optimal solution to problem $\mathrm{P}(\Theta^0)$ over $\Theta(L, U)$.

Next, we will prove the remains of this theorem. For any $(t, s) \in \Theta_a(L, U)$, notice that $t_h < \rho_h s_h$, we can obtain that

$$\begin{aligned}
\sum_{i=1}^{p} \frac{t_i}{s_i} &= \sum_{i=1, i\neq h}^{p} \frac{t_i}{s_i} + \frac{t_h}{s_h} \\
&< \sum_{i=1, i\neq h}^{p} \frac{t_i}{s_i} + LB_{best} - \sum_{i=1, i\neq h}^{p} U_i \\
&= \sum_{i=1, i\neq h}^{p} (\frac{t_i}{s_i} - U_i) + LB_{best} \\
&\le LB_{best}
\end{aligned}$$

since $t_i \le s_i U_i$ for each $i = 1, 2, \ldots, p$. This implies that

$$v(\Theta_a) = \max_{\forall i,(t_i,s_i)\in F_i} \sum_{i=1}^{p} \frac{t_i}{s_i} < LB_{best}.$$

Therefore, there is no the optimal solution to problem $\mathrm{P}(\Theta^0)$ over $\Theta_a(L, U)$. □

To give Theorem 2.4 below, let us consider the following two linear programs for each $j = 1, 2, \ldots, p$,

$$Q1^{(j)}(\Theta) \quad \begin{cases} U^1(j) = & \max(\alpha_j - (U_j + 1)(L_j c^j - d^j)x)/l_j \\ & \text{s.t.} \quad Ax \le b, x \ge 0, \\ & \qquad (2.6) - (2.7) \end{cases}$$

and

$$Q2^{(j)}(\Theta) \quad \begin{cases} U^2(j) = & \max(\beta_j - (L_j + 1)(U_j c^j - d^j)x)/u_j \\ & \text{s.t.} \quad Ax \le b, x \ge 0, \\ & \qquad (2.6) - (2.7). \end{cases}$$

Denote

$$\tau_j = \min\{U^1(j), U^2(j)\},\ j = 1, 2, \ldots, p, \tag{2.10}$$

then we can get the following conclusion.

Theorem 2.4. *For any* $\Theta(L, U) = \prod_{i=1}^{p} \Theta_i(L_i, U_i) \subseteq \Theta(L^0, U^0)$, *if there exists some index* $h \in \{1, \ldots, p\}$ *satisfying* $\tau_h < U_h$, *then there is no the feasible solution to problem* $P(\Theta^0)$ *over* $\Theta_b(L, U)$, *where*

$$\Theta_b(L, U) = \prod_{i=1}^{p} \Theta_{bi}(L_i, U_i) \quad \textit{with} \quad \Theta_{bi}(L_i, U_i) = \begin{cases} \Theta_i(L_i, U_i), & \textit{if } i \neq h, \\ \Delta_{bh}(\tau_h, U_h) \cap \Theta_h(L_h, U_h), & \textit{if } i = h \end{cases}$$

satisfying $\Delta_{bh}(\tau_h, U_h) = \{(t_h, s_h) \in R^2 | \tau_h s_h < t_h \le U_h s_h\}$.

Proof. If the inclusion of Theorem 2.4 is not true, there exists a feasible solution $(x^\star, t^\star, s^\star)$ to problem $\mathrm{P}(\Theta^0)$ over $\Theta_b(L,U)$ satisfying

$$\tau_h s_h^\star < t_h^\star \le U_h s_h^\star, \qquad L_i s_i^\star \le t_i^\star \le U_i s_i^\star, \quad i = 1,2,\ldots,p,\ i \ne h, \tag{2.11}$$

where $t_i^\star = d_i^\top x^\star + \delta_i$, $\quad s_i^\star = c_i^\top x^\star + \gamma_i$, $\quad i = 1,2,\ldots,p$.

By the definition of τ_h consider the following two cases.

Case (i). If $\tau_h = U^1(h)$, it follows that

$$\tau_h \quad \ge (\alpha_h - (U_h+1)(L_h c^h - d^h)x^\star)/l_h = \frac{U_h+1}{l_h}(t_h^\star - L_h s_h^\star) + L_h$$

since $x^\star$ is also a feasible solution to problem $Q1^{(h)}(\Theta)$. Further, from (2.2) we have

$$\tau_h \ge f_h(t_h^\star, s_h^\star). \tag{2.12}$$

Case (ii). If $\tau_h = U^2(h)$, similar to the above proof, we can obtain that

$$\tau_h \ge g_h(t_h^\star, s_h^\star). \tag{2.13}$$

Based on the above results (2.12) and (2.13), in either case from (2.2) and (2.3) we have

$$\tau_h \ge \min\{f_h(t_h^\star, s_h^\star), g_h(t_h^\star, s_h^\star)\} = \Phi_h(t_h^\star, s_h^\star) \ge \frac{t_h^\star}{s_h^\star}. \tag{2.14}$$

This will imply that (2.14) is contradictive to (2.11), and the proof is complete. □

By using Theorems 2.3 and 2.4, we can give a new cone reduction procedure that includes the two basis steps (Optimality and Feasibility) to reject some regions in which the globally optimal solution of $\mathrm{P}(\Theta^0)$ does not exist.

Cone reduction (CR):

(1) Optimality. Compute RU in (2.8). If $RU < LB_{best}$, let $\Theta(L,U) = \emptyset$; otherwise, compute ρ_i $(i = 1,\ldots,p)$ in (2.9). If $\rho_h > L_h$ for some $h \in \{1,\ldots,p\}$, then let $L_h = \rho_h$ and $\Theta(L,U) = \prod_{i=1}^{p} \Theta_i(L_i,U_i)$ if $\rho_h \le U_h$, and let $\Theta(L,U) = \emptyset$ if $\rho_h > U_h$.

(2) Feasibility. Compute τ_i $(i = 1,\ldots,p)$ in (2.10). If there exists some $h \in \{1,\ldots,p\}$ satisfying $\tau_h < U_h$, then let $U_h = \tau_h$ and $\Theta(L,U) = \prod_{i=1}^{p} \Theta_i(L_i,U_i)$ if $\tau_i \ge L_h$, and let $\Theta(L,U) = \emptyset$ if $\tau_i < L_h$.

This cone reduction procedure provides a possibility to cut away all or a large part of the $2p$-dimensional sub-cone $\Theta(L,U)$ created by the branch process.

3 Algorithm and Its Convergence

In this section, based on the former the new cone reduction and bound tightening procedures, a new branch-reduction-bound algorithm is proposed to find the globally optimal solution of $\mathrm{P}(\Theta^0)$. There are three fundamental processes in the algorithm search: reduction, branching and updating upper and lower bounds.

Firstly, under the appropriate condition the reduction process can cut away all or a large part of the currently investigated feasible region in which the globally optimal solution does not exist.

The second fundamental process iteratively subdivides the cone Θ^k into two sub-cones. During the iteration of the algorithm, the branching process creates a more refined partition that cannot yet be excluded from further consideration in searching for a globally optimal solution for $\mathrm{P}(\Theta^0)$. In this paper, we choose a simple bisection rule which is called bisection of ratio. This branching rule is given as follows.

Consider any node subproblem identified by $\Theta(L,U) = \prod_{i=1}^{p} \Theta_i(L_i,U_i)$. Let $j \in \operatorname{argmax}\{U_i - L_i | i = 1,\ldots,p\}$, and denote $w_j = \lambda L_j + (1-\lambda)U_j$ with $\lambda \in (0, 0.5]$. Then $\{\bar{\Theta},\ \bar{\bar{\Theta}}\}$ is called a conical bisection of Θ, where

$$\bar{\Theta} = \Theta_1 \times \cdots \times \Theta_{j-1} \times \bar{\Theta}_j \times \Theta_{j+1} \times \cdots \times \Theta_p,$$
$$\bar{\bar{\Theta}} = \Theta_1 \times \cdots \times \Theta_{j-1} \times \bar{\bar{\Theta}}_j \times \Theta_{j+1} \times \cdots \times \Theta_p$$

with

$$\bar{\Theta}_j = \{(t_j, s_j) \in R^2 | L_j s_j \le t_j \le w_j s_j\},$$
$$\bar{\bar{\Theta}}_j = \{(t_j, s_j) \in R^2 | w_j s_j \le t_j \le U_j s_j\}.$$

The third process is to update the upper and lower bounds of the optimal objective function value of $\mathrm{P}(\Theta^0)$. This process needs to solve a sequence of linear programming problems and to compute the corresponding objective function value of $\mathrm{P}(\Theta^0)$. In addition, the current bound tightening procedure is applied to the proposed algorithm.

The basic steps of the proposed algorithm are summarized as follows.

3.1 Algorithm statement

Step 0 (Initialization)

(0.1) Given a convergence tolerance $\varepsilon > 0$, choose the parameters $\sigma \in (0,1)$ and $\lambda \in (0, 0.5]$. Set the iteration counter $k = 0$, the active node index set $Q_0 = \{0\}$. Let $\Theta^k = \prod_{i=1}^{p} \Theta_i^k = \Theta^0$.

(0.2) Solve problem $\mathrm{LP}(\Theta^k)$ to get the solution $(\hat{x}(\Theta^k), \hat{r}(\Theta^k))$. Let $\hat{t}_i = d_i^\top \hat{x}(\Theta^k) + \delta_i,\ \hat{s}_i = c_i^\top \hat{x}(\Theta^k) + \gamma_i$, and let $\bar{r}_i(\Theta^k) = \Psi_i(\hat{t}_i, \hat{s}_i; \sigma)$ for each $i = 1, 2, \ldots, p$. Set the upper bound $UB(k) = \sum_{i=1}^{p} \bar{r}_i(\Theta^k)$.

(0.3) Set the current best feasible point $x^* = \hat{x}(\Theta^k)$, and the lower bound $LB_{best} = h(x^*)$.

(0.4) If $UB(k) - LB_{best} \le \varepsilon$, stop, and x^* is the globally optimal solution and LB_{best} is the optimal value to (P). Otherwise, proceed to Step 1.

Step 1 (Reduction) For the sub-cone Θ^k that is currently investigated, CR described in Section 3 is applied to Θ^k and still denote the remaining as Θ^k.

Step 2 (Branching) According to the above conical partition rule, partition Θ^k to get $\Theta^{k.1}$ and $\Theta^{k.2}$. Replace k by node indices $k.1,\ k.2$ in Q_k.

Step 3 (Bounding) For each $k.\iota$, where $\iota = 1, 2$, compute $RU^{k.\iota}$ as given in (2.8). If $RU^{k.\iota} < LB_{best}$, then the corresponding node indices $q(k).\iota$ will be eliminated. If $\Theta^{k.\iota}$ $(\iota = 1, 2)$ are all eliminated, then go to Step 4. Otherwise, for each not eliminated $\Theta^{k.\iota}$, where $\iota = 1$ or 2 or $\iota = 1, 2$, solve $(\mathrm{LP}(\Theta^{k.\iota}))$ to get the solution $(\hat{x}(\Theta^{k.\iota}), \hat{r}(\Theta^{k.\iota}))$. Let $\hat{t}_i^{k.\iota} = d_i^\top \hat{x}(\Theta^{k.\iota}) + \delta_i,\ \hat{s}_i^{k.\iota} = c_i^\top \hat{x}(\Theta^{k.\iota}) + \gamma_i$, for each $i = 1, \ldots, p$. Then by utilizing BT the corresponding tighten upper bound is set $UB_{k.\iota} = \sum_{i=1}^{p} \Psi_i(\hat{t}_i^{k.\iota}, \hat{s}_i^{k.\iota}; \sigma)$. Update the feasible point x^* and the lower bound LB_{best} such that $h(x^*) = \max\{h(\hat{x}(\Theta^{k.\iota})), h(x^*)\}$ and $LB_{best} = h(x^*)$.

Step 4 (Fathoming) Fathoming any improving nodes by setting

$$Q_{k+1} = Q_k - \{q \in Q_k | UB_q \leq LB_{best} + \varepsilon\}.$$

If $Q_{k+1} = \emptyset$, then stop, and LB_{best} is the optimal value and x^* is the global solution to (P). Otherwise, set $k = k + 1$.

Step 5 (Node Selection) Set $UB(k) = \max\{UB_q | q \in Q_k\}$, then select an active node $q \in Q_k$ satisfying $UB(k) = UB_q$ for further considering, and go to Step 1.

Remark. In the above algorithm, notice that CR in Step 1 and BT in Step 3 can enhance the solution procedure of the algorithm. If we do not use CR (i.e. Step 1 is discarded) and choose $\sigma = 1$ in Step 3, then the corresponding simplified algorithm from the above is the same as one given by Kuno in [12].

3.2 Algorithm convergence

By the construction of the algorithm, when the algorithm is finite, it can find a global solution to problem (P). It is also possible for the algorithm to be infinite. The following definition and lemmas help to analyze these results.

Definition 3.1. The algorithm of problem (P) is said to be convergent if it is infinite and $\lim\limits_{l\to\infty} h(\hat{x}^l) = v$, or if it is finite.

When the algorithm is infinite, since $\{1, 2, \ldots, p\}$ is finite, there exists an infinite sequence $\{\Theta^l\}_{l=1}^{\infty}$ generated by the algorithm, such that for each $l = 1, 2, \ldots$, $\Theta^{l+1} \subset \Theta^l$ and Θ^{l+1} is formed from Θ^l by the conical partition process. Additionally, in Step 2 of the algorithm, for some fixed $j_0 \in \{1, 2, \ldots, p\}$, we have

$$U_{j_0}^l - L_{j_0}^l = \max_{i=1,2,\ldots,p} \{U_i^l - L_i^l\},$$

and $\Theta^l = \prod\limits_{i=1}^{p} \Theta_i^l$ with $\Theta_i^l = \{(t_i, s_i) \in R^2 | L_i^l s_i \leq t_i \leq U_i^l s_i\}$ for each l and each $i = 1, 2, \ldots, p$. Assume in the next result that $\{\Theta^l\}_{l=1}^{\infty}$ is a sequence of cones of this type, and for each l and each $i = 1, 2, \ldots, p$, let

$$\Theta_i^l = \{(t_i, s_i) \in R^2 | L_i^l s_i \leq t_i \leq U_i^l s_i\}.$$

Lemma 3.2. *For some subsequence Q of $\{1, 2, \ldots\}$, the limit cone*

$$\Theta_{j_0}^* = \bigcap_l \Theta_{j_0}^l$$

is a half-line

$$\Theta_{j_0}^* = \{(t_{j_0}, s_{j_0}) \in R^2 | t_{j_0} = w_{j_0}^* s_{j_0}\},$$

where $w_{j_0}^*$ is some point in $[L_{j_0}^0, U_{j_0}^0]$.

Proof. By the conical partition rule, there exist a subsequence Q of $\{1, 2 \ldots\}$

$$L_{j_0}^0 \leq L_{j_0}^l \leq L_{j_0}^{l+1} \leq U_{j_0}^{l+1} \leq U_{j_0}^l \leq U_{j_0}^0, \ \forall\, l \in Q.$$

Hence, for some $L_{j_0}^*$ and $U_{j_0}^*$ with $L_{j_0}^0 \leq L_{j_0}^* \leq U_{j_0}^* \leq U_{j_0}^0$ we have

$$\lim_{l\to\infty} U_{j_0}^l = \lim_{l\to\infty} U_{j_0}^{l+1} = U_{j_0}^*, \ \lim_{l\to\infty} L_{j_0}^l = \lim_{l\to\infty} L_{j_0}^{l+1} = L_{j_0}^*.$$

These also imply that $\lim_{l\to\infty} w_{j_0}^l = w_{j_0}^* \in \{L_{j_0}^*, U_{j_0}^*\}$ because $w_{j_0}^l = \lambda L_{j_0}^l + (1-\lambda)U_{j_0}^l$ with $\lambda \in (0, \frac{1}{2}]$ coincides with either $L_{j_0}^{l+1}$ or $U_{j_0}^{l+1}$ for each $l \in Q$. In either case, $w_i^* \in (L_i^0, U_i^0)$ is a single point. This means that if $l \to \infty$ the cone Θ^l shrinks to a half-line:

$$\Theta^* = \{(t,s) \in R^{2p} | t_i = w_i^* s_i,\ i = 1, 2, \ldots, p\},$$

where $t = (t_1, \ldots, t_p), s = (s_1, \ldots, s_p)$, and w_i^* is some point in $[L_i^0, U_i^0]$. □

Lemma 3.3. *Suppose that the algorithm is infinite, and let $\{\Theta^l\}_{l=1}^\infty$ be a sequence of cones in R^{2p} generated by the algorithm such that for each $l = 1, 2, \ldots$, $\Theta^{l+1} \subset \Theta^l$. Then, for some subsequence Q of $\{1, 2, \ldots\}$,*

$$\lim_{l\in Q}(\sum_{i=1}^p \frac{\hat{t}_i^l}{\hat{s}_i^l} - UB(\Theta^l)) = 0,$$

where $(\hat{t}_i^l, \hat{s}_i^l)$ is an optimal solution to problem $\mathrm{P}_1(\Theta^l)$ and $UB(\Theta^l)$ is the tight upper bound based on the optimal value of $\mathrm{P}_1(\Theta^l)$.

Proof. Since $\{\Theta^l\}_{l=1}^\infty$ is infinite, by Steps 0.4 and 4 of the algorithm, we may choose a subsequence Q of $\{1, 2, \ldots\}$ such that for each $l \in Q$ we may assume without loss of generality that $\{\Theta^l\}_{l\in Q}$ has the properties of Lemma 3.2. In addition, for each i, the sequence $\{(\hat{t}_i^l, \hat{s}_i^l) | l \in Q\}$ is generated in the compact set $\Sigma_i \cap \Theta_i^0$, thus has at least one limit point $(\hat{t}_i^*, \hat{s}_i^*)$ satisfying $\hat{t}_i^* = w_i^* \hat{s}_i^*$. Therefore,

$$\lim_{l\to\infty} \frac{\hat{t}_i^l}{\hat{s}_i^l} = w_i^*. \tag{3.1}$$

On the other hand, by the proof of Theorem 2.2, $\Psi_i(\hat{t}_i^*, \hat{s}_i^*; \sigma)$ is the over-estimator of $\frac{\hat{t}_i^l}{\hat{s}_i^l}$ on $\Theta_i^l \cap \Sigma_i$, by the continuity of Ψ_i on $\Theta_i^l \cap \Sigma_i$ we have

$$\lim_{l\to\infty} \hat{r}_i^l = w_i^*, \text{ for each } i \in \{1, 2, \ldots, p\}. \tag{3.2}$$

Hence, by Step 3, it follows from (3.1) and (3.2) that

$$\lim_{l\in Q}[UB(\Theta^l) - \sum_{i=1}^p \frac{\hat{t}_i^l}{\hat{s}_i^l}] = 0.$$

This completes the proof. □

Lemma 3.4. *The proposed algorithm is convergent.*

Proof. Suppose that the algorithm is infinite. Then, as noted previously, we may choose a sequence of cones, which we denote without loss of generality by $\{\Theta^l\}_{l=1}^\infty$, such that for each $l = 1, 2, \ldots$, $\Theta^{l+1} \subset \Theta^l$ and Θ^{l+1} is formed from Θ^l by the conical partition process. By the validity of the upper bounding process and Steps 2 and 3 of the algorithm, for each $l = 1, 2, \ldots$

$$UB(\Theta^l) \geq v(\Theta^0) = v \geq \sum_{i=1}^p \frac{\hat{t}_i^l}{\hat{s}_i^l} = h(\hat{x}^l) \triangleq v^l, \tag{3.3}$$

and $\{v^l\}_{l=1}^{\infty}$ is a nondecreasing sequence of real numbers. By Lemma 2, for some $Q \subset \{1, 2, \dots\}$, we have

$$\lim_{l \in Q} UB(\Theta^l) = \lim_{l \in Q} \sum_{i=1}^{p} \frac{\hat{t}_i^l}{\hat{s}_i^l}.$$

From (3.3), this implies that

$$\lim_{l \in Q} UB(\Theta^l) = \lim_{l \in Q} \hat{v}(\Theta^0) = v = \lim_{l \in Q} \sum_{i=1}^{p} \frac{\hat{t}_i^l}{\hat{s}_i^l} = \lim_{l \in Q} h(\hat{x}^l) = \lim_{l \in Q} v^l.$$

Since $\{v^l\}_{l=1}^{\infty}$ is a nondecreasing sequence, this implies that $\lim_{l \to \infty} h(\hat{x}^l) = v$. □

Theorem 3.5. (a) *If the algorithm is finite, then upon termination, x^* is a global ε-optimal solution to problem* (P).

(b) If the algorithm for problem P(Θ^0) is infinite and convergent, then there exists an accumulation point $(\hat{x}^*, \hat{t}^*, \hat{s}^*)$ of the infinite sequence $\{(\hat{x}^l, \hat{t}^l, \hat{s}^l)\}_{l=1}^{\infty}$ such that $(\hat{x}^*, \hat{t}^*, \hat{s}^*)$ is a globally optimal solution to problem P(Θ^0), and $\hat{x}^*$ is a globally optimal solution to problem (P), where $\hat{t}_i^l = d_i^\top \hat{x}^l + \delta_i, \hat{s}_i^l = c_i^\top \hat{x}^l + \gamma_i, i = 1, 2, \dots, p$.

Proof. (a) If the algorithm is finite, then it terminates in Step k, $k \geq 1$. Upon termination, since x^* is found by solving problem LP($\Theta^{k.\iota}$) for some $\Theta^{k.\iota} \subseteq \Theta^0$, we get that x^* is a feasible solution to problem (P). Upon termination of the algorithm,

$$UB(k) - \sum_{i=1}^{p} \frac{d_i^\top x^* + \delta_i}{c_i^\top x^* + \gamma_i} \leq \varepsilon$$

is satisfied. From the algorithm, we have $UB(k) \geq v$. Since x^* is a feasible solution for problem (P), we have

$$\sum_{i=1}^{p} \frac{d_i^\top x^* + \delta_i}{c_i^\top x^* + \gamma_i} \leq v.$$

Taken together, the three previous statements imply that

$$v \leq UB(k) \leq \sum_{i=1}^{p} \frac{d_i^\top x^* + \delta_i}{c_i^\top x^* + \gamma_i} + \varepsilon \leq v + \varepsilon.$$

Therefore,

$$v - \varepsilon \leq \sum_{i=1}^{p} \frac{d_i^\top x^* + \delta_i}{c_i^\top x^* + \gamma_i} \leq v,$$

and the proof of part (a) is complete.

(b) Since the algorithm for problem P(Θ^0) is infinite and convergent, then by Lemma 3.4, we have

$$\lim_{l \to \infty} \sum_{i=1}^{p} \frac{\hat{t}_i^l}{\hat{s}_i^l} = v. \tag{3.4}$$

Since $(\hat{x}^*, \hat{t}^*, \hat{s}^*)$ is an accumulation point of $\{(\hat{x}^l, \hat{t}^l, \hat{s}^l)\}_{l=1}^{\infty}$, then for some $Q \subseteq \{1, 2, \dots\}$, such that

$$\lim_{l \in Q} (\hat{x}^l, \hat{t}^l, \hat{s}^l) = (\hat{x}^*, \hat{t}^*, \hat{s}^*). \tag{3.5}$$

By (3.4)-(3.5) and the continuity of the objective function to problem $P(\Theta^0)$, we have

$$\lim_{l\in Q}\sum_{i=1}^{p}\frac{\hat{t}_i^l}{\hat{s}_i^l}=\sum_{i=1}^{p}\frac{\hat{t}_i^*}{\hat{s}_i^*}. \tag{3.6}$$

Hence

$$\sum_{i=1}^{p}\frac{\hat{t}_i^*}{\hat{s}_i^*}=v.$$

Additionally, since the feasible region of problem $P(\Theta^0)$ is a closed set, we can obtain that $(\hat{x}^*,\hat{t}^*,\hat{s}^*)$ is feasible to $P(\Theta^0)$. Together with (20), this implies that $(\hat{x}^*,\hat{t}^*,\hat{s}^*)$ is a globally optimal solution to problem $P(\Theta^0)$, i.e. $\hat{x}^*$ is a globally optimal solution to problem (P). □

4 Numerical Experiments

To demonstrate the potentiality and feasibility of the proposed algorithm, our numerical experiment is reported in this section. We test the performance of the above algorithm and compare it with the global optimization algorithms in Ref. [3, 7, 10–12, 21–23].

The algorithm is coded in Visual Fortran 95. The simplex method is applied to solve the concerning linear programming problems $LP(\Theta)$. For all test examples, they are implemented on an Intel(R)Core(TM)Duo CPU T5870 @ 2.00Ghz with 2 GB memory microcomputer. In our experiments, it is observed that the change of the parameter $\lambda\in(0,0.5]$ in the branching operation has less influence on the computational results. Also, the main aim of the experiment in this section is to demonstrate the performance of the proposed CR and BT given in Section 2. Thus, without loss of generality, for simplicity, set $\lambda=0.3$ in our experiments, and the corresponding computational results are summarized in Tables 1-4 below.

In Tables 1-4, the notation has been used for column headers: Bra: the number of the branching operations; Node: the maximal number of the active nodes necessary; Value: the optimal value; Solution: the optimal solution; Time: the execution time of CPU in seconds, where we record with 0 second in short if the execution time is very short (Time$<10^{-3}$ second, for example).

In order to illustrate the effectiveness of CR and BT, Steps 1 and 3 of the algorithm execute the different operation procedure in our experiments depending on the distinction adopting CR and BT. The notation in Tables 1-4 is given as follows:

BT1: the results by alone using BT with $\sigma=1.0$;

BT2: the results by utilizing separately BT with $\sigma=0.01$;

CRBT: the results by adopting both CR and BT with $\sigma=0.01$.

We emphasize here that the objective values in Tables 2-4, which are obtained separately from BT1, DBT and CRBT, are the same for a given example. Additionally, it should be noted that the results BT1 computed by us, actually, also stand for ones achieved by implementing the algorithm TRAPEZOID given by Kuno in [12].

Next, we first describe some simple examples in order to compare with [3,7,10,11,21,22], and then some problems generated randomly are implemented to test further the algorithm. The computational results of simple examples are given in Table 1, and it is seen that CR and BT can improve the computational efficiency, that is, the number of branching operations and the maximal number of the active node necessary can be reduced significantly. Therefore, CR and BT are all necessary and effective to improve the solution procedure. Especially, the smaller the value of σ in BT, the better the performance of the algorithm will

be. Based on these observations, we further test the algorithm for some random problems by choosing $\sigma = 0.01$ in the end of this section.

The test examples are defined as follows.

Example 1 (Refs. [10, 11, 21])**.**

$$\begin{array}{ll} \max & \dfrac{3x_1+5x_2+3x_3+50}{3x_1+4x_2+5x_3+50}+\dfrac{3x_1+4x_2+50}{4x_1+3x_2+2x_3+50}+\dfrac{4x_1+2x_2+4x_3+50}{5x_1+4x_2+3x_3+50} \\ \text{s.t.} & 6x_1+3x_2+3x_3 \le 10 \\ & 10x_1+3x_2+8x_3 \le 10 \\ & x_1, x_2, x_3 \ge 0. \end{array}$$

Example 2 (Refs. [10, 11, 21])**.**

$$\begin{array}{ll} \max & \dfrac{4x_1+3x_2+3x_3+50}{3x_2+3x_3+50}+\dfrac{3x_1+4x_3+50}{4x_1+4x_2+5x_3+50}+\dfrac{x_1+2x_2+5x_3+50}{x_1+5x_2+5x_3+50} \\ & +\dfrac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50} \\ \text{s.t.} & 2x_1+x_2+5x_3 \le 10 \\ & x_1+6x_2+3x_3 \le 10 \\ & 5x_1+9x_2+2x_3 \le 10 \\ & 9x_1+7x_2+3x_3 \le 10 \\ & x_1, x_2, x_3 \ge 0. \end{array}$$

Example 3 (Refs. [3, 7])**.**

$$\begin{array}{ll} \max & \dfrac{x_1-2x_2-2}{3x_1-4x_2+5}-\dfrac{4x_1-3x_2+4}{-2x_1+x_2+3} \\ \text{s.t.} & x_1+x_2 \le 1.5 \\ & x_1-x_2 \le 0 \\ & 0 \le x_1, x_2 \le 1. \end{array}$$

Example 4 (Ref. [3])**.**

$$\begin{array}{ll} \max & \dfrac{3.3333x_1+3x_2+1}{1.6666x_1+x_2+1}+\dfrac{4x_1+3x_2+1}{x_1+x_2+1} \\ \text{s.t.} & 5x_1+4x_2 \le 10 \\ & -x_1 \le -0.1 \\ & -x_2 \le -0.1 \\ & -2x_1-x_2 \le -2 \\ & x_1, x_2 \ge 0. \end{array}$$

Example 5 (Ref. [22])**.**

$$\begin{array}{ll} \max & \dfrac{37x_1+73x_2+13}{13x_1+13x_2+13}+\dfrac{63x_1-18x_2+39}{13x_1+26x_2+13} \\ \text{s.t.} & 5x_1-3x_2 = 3 \\ & 1.5 \le x_1 \le 3. \end{array}$$

Example 6 (Ref. [22]).

$$\begin{aligned}\max\quad & \frac{4x_1+3x_2+3x_3+50}{3x_2+3x_3+50}+\frac{3x_1+4x_3+50}{4x_1+4x_2+5x_3+50}+\frac{x_1+2x_2+4x_3+50.0}{x_1+5x_2+5x_3+50}\\ & +\frac{x_1+2x_2+4x_3+50}{5x_2+4x_3+50.0}\\ \text{s.t.}\quad & 2x_1+x_2+5x_3\le 10\\ & x_1+6x_2+2x_3\le 10\\ & 9x_1+7x_2+3x_3\le 10\\ & x_1,x_2,x_3\ge 0.\end{aligned}$$

Example 7 (Ref. [21]).

$$\begin{aligned}\max\quad & \frac{37x_1+73x_2+13}{13x_1+13x_2+13}-\frac{63x_1-18x_2+39}{13x_1+26x_2+13}+\frac{13x_1+13x_2+13}{63x_2-18x_3+39}-\frac{13x_1+26x_2+13}{37x_1+73x_2+13}\\ \text{s.t.}\quad & 5x_1-3x_2=3\\ & 1.5\le x_1\le 3.\end{aligned}$$

Example 8 (Ref. [21]).

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

Further, we choose the following linear sum-of-ratios problem to test our algorithm, which are generated randomly.

Problem 1 (Refs. [11, 12, 23]

$$\begin{aligned}\max\quad & \sum_{i=1}^{p}\frac{\sum_{j=1}^{n}d_i^jx_j+c}{\sum_{j=1}^{n}c_i^jx_j+c}\\ \text{s.t.}\quad & \sum_{j=1}^{n}a_j^kx_j\le 1.0,\ k=1,2,\ldots,m,\\ & x_j\ge 0.0,\ j=1,\ldots,n,\end{aligned}$$

where $c_i^j, d_i^j \in [0.0, 0.5]$ and $a_j^k \in [0.0, 1.0]$ are uniformly random numbers. The constant terms of denominators and nominators were all set to the same number c, which ranged between 2.0 and 80.0. The convergence tolerance parameter is set as $\varepsilon = 10^{-5}$.

For solving the above test problem 1, we utilize CR and BT in the algorithm according to different computational procedures. The corresponding numerical results of CRBT and BT1 are listed in Tables 2-4, where average percentages are obtained by running the algorithm for 10 test problems.

Table 1 Computational results for Examples 1 – 8

Ex.	Ref.	Node	Bra	Time	Value	Solution	ε
1	CRBT	1	6	0	3.00292	(0,3.333333,0)	10^{-6}
	BT1	6	62	0	3.00292	(0,3.333333,0)	10^{-6}
	BT2	1	7	0	3.00292	(0,3.333333,0)	10^{-6}
	[10]		30		3.000042	(0,0.33329,0)	10^{-6}
	[21]	14	25	0	3.00292	(0,0.33333,0)	10^{-6}
	[11]	52	103		2.9312	(0,0,1.25)	10^{-3}
2	CRBT	1	2	0	4.0907	(1.11111,0,0)	10^{-6}
	BT1	2	9	0	4.0907	(1.11111,0,0)	10^{-6}
	BT2	1	7	0	4.0907	(1.11111,0,0)	10^{-6}
	[10]		17		4.087412	(1.0715,0,0)	10^{-6}
	[21]	2	3	0	4.0907	(1.11111,0,-3.333067e-016)	10^{-6}
	[11]	59	117		3.798367	(0,1.111,0)	10^{-3}
3	CRBT	2	10	0	-1.6232	(0,0.2873)	10^{-2}
	BT1	9	80	0	-1.6232	(0,0.2873)	10^{-2}
	BT2	5	24	0	-1.6232	(0,0.2873)	10^{-2}
	[7]				-1.6240	(0,0.2839)	
	[3]		18		-1.6236	(0,0.2679)	10^{-2}
4	CRBT	1	3	0	4.8415	(0.1,2.375)	10^{-2}
	BT1	2	3	0	4.8415	(0.1,2.375)	10^{-2}
	BT2	1	4	0	4.8415	(0.1,2.375)	10^{-2}
	[3]		4		4.8415	(0.1,2.375)	10^{-2}
5	CRBT	1	2	0	5	(3,4)	10^{-4}
	BT1	2	10	0	5	(3,4)	10^{-4}
	BT2	1	2	0	5	(3,4)	10^{-4}
	[22]	32	32	1.089285	5	(3,4)	10^{-4}
6	CRBT	1	3	0	4.42857	(5,0,0)	10^{-4}
	BT1	2	35	0	4.42857	(5,0,0)	10^{-4}
	BT2	2	22	0	4.42857	(5,0,0)	10^{-4}
	[22]	18	58	2.968684	4	(0,0.625,1.875)	10^{-4}
7	CRBT	1	3	0	3.29167	(3,4)	10^{-6}
	BT1	5	78	0	3.29167	(3,4)	10^{-6}
	BT2	1	6	0	3.29167	(3,4)	10^{-6}
	[21]	3	9	0	3.29167	(3,4)	10^{-6}
8	CRBT	1	1	0	-1.9	(0,3.33333,0)	10^{-6}
	BT1	1	8	0	-1.9	(0,3.33333,0)	10^{-6}
	BT2	1	2	0	-1.9	(0,3.33333,0)	10^{-6}
	[21]	3	8	0	-1.9	(-1.83881e-16,3.33333,0)	10^{-6}

First, given $(m, n) = (60, 40)$ and $c = 10.0$, Table 2 shows the variation in Time, Bra and Node as changing $p \in \{4, 6, \ldots, 10, 13, 15, 18, 20, 23, 25, 28, 30\}$. From Table 2 it is not difficult to find that Time, Bra and Node increase as an exponential function in p, when CR and BT with $\sigma = 1$ (i.e. TRAPEZOID) are executed. However, we should notice that CRBT shows fairly less Time, Bra and Node than BT1 for each p. This demonstrates that the proposed CR and BT are rather effective for improving the computational efficiency of the algorithm.

Second, when (m, n) is larger than $(40, 60)$, the corresponding computational results are listed in Tables 3 and 4 with c fixed at 10.0 for each $p \in \{4, 5, 6, 7, 10, 13, 15\}$. Observing Tables 3 and 4, we can obtain that Node, Bra and Time computed by each algorithm increase mildly with increase in the size of (m, n), in constant to their rapid change depending on p. Specially, the overall result of CRBT is superior to that of BT1 for every (m, n) and p. Since the former always requires less branching operations (Bra), CPU time (Time) and the longest node number (Node) than the latter. Unfortunately, each number about Bra, Time and Node increases un-regularly on instance of size (m, n) lager than $(140, 120)$.

5 Concluding Remarks

In this paper, a new efficient algorithm is presented for solving globally the problem (P), which is based on the new cone reduction and bound tightening procedures and the trapezoidal method in Ref. [12]. The cone reduction can reduce the size of the currently investigated feasible region without losing any current feasible solution still of interest, and the bounding tightening is able to suppress the rapid growth of branching trees. Using these procedures as an accelerating device and applying them to the branch-and-bound algorithm, we can delete or reduce the current region not including the optimal solution as fast as possible. Numerical experiments show that computational efficiency can be improved obviously by using the cone reduction and the bound tightening, specially, the number of the branching operations, the maximal number of the active nodes and the execution time of CPU can be significantly reduced.

Table 2 Computational results of Problem 1 when $(m, n) = (60, 40), c = 10$

p	Node		Bra		Time	
	CRBT	BT1	CRBT	BT1	CRBT	BT1
4	23.1	58.2	43.4	126.5	2.3	7.2
6	44.0	93.5	81.5	301.3	4.8	12.4
8	0.5	185.6	127.0	485.6	10.4	23.7
10	120.0	339.4	185.7	590.0	17.4	34.1
13	191.6	541.3	292.0	674.5	31.1	50.9
15	277.0	887.5	464.7	887.3	52.8	78.1
18	358.5	626.8	839.0	2179.3	79.9	841.3
20	510.2	896.4	1522.0	3050.6	139.8	1305.6
23	681.0	1306.5	2531.5	3580.4	206.8	1735.5
25	786.4	1986.6	3206.3	5054.8	349.9	2456.2
28	1139.0	2113.5	4510.0	7491.3	514.5	4566.3
30	1642.0	2151.3	5350.5	8238.5	729.2	5673.4

Table 3 Computational results for Problem 1 when $c = 10.0$

(m, n)		$p = 4$		$p = 5$		$p = 6$		$p = 7$	
		CRBT	BT1	CRBT	BT1	CRBT	BT1	CRBT	BT1
(40,60)	Node	22.0	60.5	46.9	92.3	49.0	126.0	87.5	179.0
	Bra	40.0	138.4	61.7	271.1	75.0	335.0	188.0	591.7
	Time	2.2	7.1	5.0	18.9	5.1	18.2	18.5	47.5
(80,60)	Node	35.0	77.6	61.3	127.5	80.0	168.0	138.0	260.2
	Bra	60.0	177.1	123.1	315.4	162.0	398.0	197.8	686.7
	Time	9.2	20.0	26.9	51.1	30.7	85.6	41.5	105.1
(60,80)	Node	37.3	126.0	69.1	143.4	89.0	259.1	142.0	291.0
	Bra	62.0	385.0	112.0	541.1	187.0	727.0	188.0	909.9
	Time	16.9	65.6	31.6	74.9	45.2	95.3	74.9	191.3
(100,80)	Node	54.0	150.0	74.1	180.0	125.0	277.1	169.0	306.3
	Bra	77.5	473.0	189.0	644.3	251.1	747.0	505.0	951.1
	Time	33.5	84.1	81.5	245.3	144.8	401.6	291.5	430.0
(80,100)	Node	59.0	229.3	84.1	260.0	189.1	533.0	232.0	585.3
	Bra	173.0	1278.6	240.3	1424.0	215.3	1862.7	523.1	2045.0
	Time	100.3	308.9	106.2	355.7	143.8	649.2	379.5	905.9
(120,100)	Node	79.0	237.4	119.5	289.0	268.0	556.0	307.0	723.1
	Bra	262.0	1519.8	287.2	1757.5	439.0	2006.1	656.3	2310.4
	Time	123.5	804.8	123.0	1031.6	591.7	1234.1	1305.3	1502.8

Table 4 Computational results for Problem 1 when $c = 10.0$

(m,n)		$p=4$		$p=7$		$p=10$		$p=13$		$p=15$	
		CRBT	BT1	CRBT	BT1	CRBT	BT1	CRBT	BT1	CRBT	BT1
(140,120)	Node	99.5	456.6	300.2	620.3	432.7	2367.8	1043.3	1902.4	1318.0	2311.8
	Bra	317.0	2855.4	707.1	3802.6	1586.1	8523.6	2974.0	13281.4	5210.4	20494.0
	Time	287.2	2949.4	1577.6	4824.9	2357.1	32546.1	7095.4	66304.7	7978.8	85129.3
(160,140)	Node	127.2	500.1	427.0	1178.6	632.0	1490.4	1337.0	2068.5	1591.6	2864.3
	Bra	357.0	2432.2	822.6	5401.6	1786.1	3299.0	3760.5	13860.5	5392.3	15538.3
	Time	1140.1	5474.5	3287.0	12156.0	5349.2	36934.5	7032.8	68645.7	9181.4	120929.0

Acknowledgement

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

References

[1] Y. Almogy and O. Levin, *Parametric Analysis of A Multi-Stage Stochastic Shipping Problem*, Operational Research 69, Tavistok Publications, London, 1970.

[2] H.P. Benson, On the global optimization of sums of linear fractional functions over a convex set, *Journal of Optimization Theory and Applications* 1 (2004) 19–39.

[3] H.P. Benson, A simplicial branch and bound duality-bounds algorithm for the linear sum-of-ratios problem, *European Journal of Operational Research* 182 (2007) 597–611.

[4] D.Z. Chen, O. Daescu, Y. Dai, N. Katoh, X. Wu and J. Xu, Optimizing the sum of linear fractional functions and applications, in *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms*, ACM, New York, 2000.

[5] C.S. Colantoni, R.P. Manes and A. Whinston, Programming, profit rates and pricing decisions, *The Accounting Review* 44 (1969) 467–481.

[6] Z. Drezner, S. Schaible and D. Simchi-Levi, Queueing-location problems on the plane, *Naval Research Logistics* 37 (1990) 929–935.

[7] J.E. Falk and S.W. Palocsay, Image space analysis of generalized fractional programs, *Journal of Global Optimization* 4 (1994) 63–88.

[8] Y. Gao and S. Jin, A global optimization algorithm for sum of linear ratios problem, *Journal of Applied Mathematics* (2013) http://dx.doi.org/10.1155/2013/276245.

[9] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, third ed., Springer, Berlin, 1996.

[10] Y. Ji, K.C. Zhang and S.J. Qu, A deterministic global optimization algorithm. *Applied Mathematics and Computation* 185 (2007) 382–387.

[11] T. Kuno, A branch-and-bound algorithm for maximizing the sum of several linear ratios, *Journal of Global Optimization* 22 (2002) 155–174.

[12] T. Kuno, A revision of the trapezoidal branch-and-bound algorithm for linear sum-of-ratios problems, *Journal of Global Optimization* 33 (2005) 215–234.

[13] H. Konno and K. Fukaishi, A branch and bound algorithm for solving low rank linear multiplicative and fractional programming problems, *Journal of Global Optimization* 18 (2000) 283–299.

[14] H. Konno and M. Inori, Bond portfolio optimization by bilinear fractional programming, *Journal of the Operations Research Society of Japan* 32 (1989) 143–158.

[15] H. Konno, Y. Yajima and T. Matsui, Parametric simplex algorithms for solving a special class of nonconvex minimization problems, *Journal of Global Optimization* 1 (1991) 65–81. (1991)

[16] H. Konno and H. Yamashita, Minimizing sums and products of linear fractional functions over a polytope, *Naval Research Logistics* 46 (1999) 583–596.

[17] T. Matsui, NP-hardness of linear multiplicative programming and related problems, *Journal of Global Optimization* 9 (1996) 113–119.

[18] M.R. Rao, Cluster analysis and mathematical programming, *Journal of the American Statistical Association* 66 (1971) 622–626.

[19] S. Schaible, A note on the sum of a linear and linear-fractional function, *Naval Research Logistics Quarterly* 24 (1977) 691–693.

[20] S. Schaible and J. Shi, Fractional programming: The sum-of-ratios case, *Optimization Methods and Software* 18 (2003) 219–229.

[21] P.P. Shen and C.F. Wang, Global optimization for sum of linear ratios problem with coefficient, *Applied Mathematics and Computation* 176 (2006) 219–229.

[22] C.F. Wang and P.P. Shen, A global optimization algorithm for linear fractional programming, *Applied Mathematics and Computation* 204 (2008) 281–287.

[23] Y.J. Wang, P.P. Shen and Z.A. Liang, A branch-and-bound algorithm to globally solve the sum of several linear ratios, *Applied Mathematics and Computation* 168 (2005) 89–101.

Manuscript received 22 April 2014
revised 5 September 2014
accepted for publication 28 November 2014

Pei-Ping Shen
College of Mathematics and Information Science
Henan Normal University, Xinxiang 453007, PR China
E-mail address: shenpp@htu.cn

Wei-Min Li
College of Mathematics and Information Science
Henan Normal University, Xinxiang 453007, PR China
E-mail address: liwm81@126.com

Yan-Chao Liang
College of Mathematics and Information Science
Henan Normal University, Xinxiang 453007, PR China
E-mail address: liangyanchao83@163.com