

Research Article

Global Minimization for Generalized Polynomial Fractional Program

Xue-Ping Hou, Pei-Ping Shen, and Chun-Feng Wang

College of Mathematics and Information Science, Henan Normal University, Xinxiang 453007, China

Correspondence should be addressed to Pei-Ping Shen; shenpp@htu.cn

Received 22 February 2014; Revised 15 May 2014; Accepted 21 May 2014; Published 9 June 2014

Academic Editor: Changzhi Wu

Copyright © 2014 Xue-Ping Hou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper is concerned with an efficient global optimization algorithm for solving a kind of fractional program problem (P), whose objective and constraints functions are all defined as the sum of ratios generalized polynomial functions. The proposed algorithm is a combination of the branch-and-bound search and two reduction operations, based on an equivalent monotonic optimization problem of (P). The proposed reduction operations specially offer a possibility to cut away a large part of the currently investigated region in which the global optimal solution of (P) does not exist, which can be seen as an accelerating device for the solution algorithm of (P). Furthermore, numerical results show that the computational efficiency is improved by using these operations in the number of iterations and the overall execution time of the algorithm, compared with other methods. Additionally, the convergence of the algorithm is presented, and the computational issues that arise in implementing the algorithm are discussed. Preliminary indications are that the algorithm can be expected to provide a practical approach for solving problem (P) provided that the number of variables is not too large.

1. Introduction

Consider the following generalized polynomial fractional programs:

$$(P) : \begin{cases} \min & g_0(y) = \sum_{j=1}^p c_j \frac{n_j(y)}{d_j(y)} \\ \text{s.t.} & g_m(y) \leq 0, \quad m = 1, 2, \dots, M_0, \\ & y \in Y = \{y \mid 0 < y_i^l \leq y_i \leq y_i^u < \infty, \\ & \quad \quad \quad i = 1, \dots, n_0\}, \end{cases} \quad (1)$$

where

$$n_j(y) = \sum_{t=1}^{\bar{T}_j} \bar{\alpha}_{jt} \prod_{i=1}^{n_0} y_i^{\bar{y}_{jti}}, \quad d_j(y) = \sum_{t=1}^{\hat{T}_j} \hat{\alpha}_{jt} \prod_{i=1}^{n_0} y_i^{\hat{y}_{jti}}, \quad j = 1, 2, \dots, p, \quad (2)$$

$$g_m(y) = \sum_{t=1}^{\bar{T}_m} \bar{\alpha}_{jt} \prod_{i=1}^{n_0} y_i^{\bar{y}_{jti}}, \quad m = 1, \dots, M_0,$$

and c_j , $\bar{\alpha}_{jt}$, $\hat{\alpha}_{jt}$, \bar{y}_{jti} , \hat{y}_{jti} , and \bar{y}_{jti} are all arbitrary real number.

Problem (P) is worth studying because it frequently appears in many applications, including financial optimization, portfolio optimization, engineering design, manufacturing, chemical equilibrium (see, e.g., [1–8]), etc. On the other hand, many other nonlinear problems, such as quadratic program, linear (or quadratic, polynomial) fractional program [9–13], linear multiplication program [14–16], polynomial program, and generalized geometric program [17–20], can be all put into this form.

The problem (P) is obviously multiextremal, for its special cases such as quadratic program, linear fractional program, and linear multiplication program are multiextremal, which are known to be NP-hard problems [21], and it, therefore, falls into the domain of global optimization problems.

In the last decades, many solution algorithms have been developed to globally solve special cases of problem (P) (see, e.g., [9–14, 17–19, 22, 23]), but the global optimization algorithms for the general form of (P) are scarce. Recently, by using the linear relaxation methods, Wang and Zhang [24], Shen and Yuan [25], and Jiao et al. [26] gave the corresponding global optimization algorithms for finding the global minimum of (P), respectively. Also, Fang et al. [27] presented

a canonical dual approach for minimizing the sum of a quadratic function and the ratio of two quadratic functions.

In this paper, we will suggest an efficient algorithm for solving globally problem (P). The goal of this research is fourfold. First, by introducing variables and by using a transformation, the original problem (P) is equivalently reformulated as a monotonic optimization problem (Q) based on the characteristics of problem (P). That is to say, the objective function is increasing and all the constrained functions can be denoted as the difference of two increasing functions in (Q). Second, in order to present an efficient algorithm for solving problem (Q), the two reduction operations are incorporated into the branch-and-bound framework to suppress the rapid growth of the branching trees so that the solution procedure is enhanced. The proposed reduction cut operation especially does not appear in other branch-and-bound methods (see [24, 25]) and is more easily implementable than the one in [28], because the latter (see (2.4) and (2.5) in [28]) is computed by solving the nonlinear nonconvex programming, but the former is involved in solving the roots of several equations in a single variable and with strict monotonicity. Third, by utilizing directly the proposed algorithm, one also can obtain the essential upper and lower bounds of denominator of each ratio in the objective function to problem (P), where these bounds are tighter than the ones given by Bernstein algorithm (see [24, 25]), and so the assumption 1 in [24, 25] is not necessary in this paper. Finally, numerical results show that the proposed algorithm is feasible and effective.

The paper is organized as follows. In Section 2, an equivalent reformulation of the original problem is given. Next, Section 3 presents and discusses the algorithm basis process for globally solving problem (P). The algorithm is presented and its convergence is shown in Section 4. In Section 5, the computational results are presented.

2. Equivalent Monotonic Reformulation

For the convenience of the following discussion, assume that there exist positive scalars L_j, U_j such that $0 < L_j \leq d_j(y) \leq U_j$ and $n_j(y) > 0$ for all $y \in Y$, for each $j = 1, 2, \dots, p$. In fact, L_j, U_j can be obtained by the algorithm to be proposed in this paper (see Section 5); define, therefore, the set

$$S = \{s \in R^p \mid L_j \leq s_j \leq U_j, j = 1, \dots, p\}. \quad (3)$$

Without loss of generality, assume that $c_j > 0, j = 1, 2, \dots, k$ and $c_j < 0, j = k + 1, k + 2, \dots, p$. By introducing variables $s_j, j = 1, \dots, p$, the problem (P) is then equivalent to the following problem:

$$(\bar{P}) : \begin{cases} \min & f(y, s) = \sum_{j=1}^p c_j s_j^{-1} n_j(y) \\ \text{s.t.} & s_j - d_j(y) \leq 0, \quad j = 1, \dots, k, \\ & d_j(y) - s_j \leq 0, \quad j = k + 1, \dots, p, \\ & g_m(y) \leq 0, \quad m = 1, \dots, M_0, \\ & y \in Y, \quad s \in S. \end{cases} \quad (4)$$

Theorem 1. *If (y^*, s^*) is a global optimal solution for problem (\bar{P}) , then $s_j^* = d_j(y^*)$, $j = 1, 2, \dots, p$, and y^* is a global*

optimal solution for problem (P). Conversely, if y^ is a global optimal solution for problem (P), then (y^*, s^*) is a global optimal solution for problem (\bar{P}) , where $s_j^* = d_j(y^*)$, $j = 1, 2, \dots, p$.*

Proof. See Theorem 1 in [24]; it is omitted here. \square

In what follows, we show that problem (\bar{P}) can be transformed into a monotonic optimization problem such that the objective function is increasing and all the constrained functions are the difference of two increasing functions. To see how such a reformulation is possible, we first consider each constraint of (\bar{P}) . Let

$$\begin{aligned} \hat{y}_{ji} &= \min \{ \{ \hat{y}_{jti}, 0 \} \mid t = 1, \dots, \hat{T}_j \}, \\ & \quad j = 1, \dots, p, \\ & \quad i = 1, \dots, n_0, \\ \tilde{y}_{mi} &= \min \{ \tilde{y}_{mti} \mid t = 1, \dots, \tilde{T}_m \}, \\ & \quad m = 1, \dots, M_0, \\ & \quad i = 1, \dots, n_0. \end{aligned} \quad (5)$$

For any $y \in Y, s \in S$, it follows from each constraint of (\bar{P}) that

$$\begin{aligned} & (s_j - d_j(y)) \cdot \prod_{i=1}^{n_0} y_i^{(-\hat{y}_{ji})} \\ &= s_j \prod_{i=1}^{n_0} y_i^{(-\hat{y}_{ji})} - \sum_{t=1}^{\hat{T}_j} \hat{\alpha}_{jt} \prod_{i=1}^{n_0} y_i^{\hat{y}_{jti} - \hat{y}_{ji}}, \\ & \quad j = 1, \dots, k, \\ & (d_j(y) - s_j) \cdot \prod_{i=1}^{n_0} y_i^{(-\hat{y}_{ji})} \\ &= \sum_{t=1}^{\hat{T}_j} \hat{\alpha}_{jt} \prod_{i=1}^{n_0} y_i^{\hat{y}_{jti} - \hat{y}_{ji}} - s_j \prod_{i=1}^{n_0} y_i^{(-\hat{y}_{ji})}, \\ & \quad j = k + 1, \dots, p, \\ & g_m(y) \cdot \prod_{i=1}^{n_0} y_i^{(-\tilde{y}_{mi})} = \sum_{t=1}^{\tilde{T}_m} \tilde{\alpha}_{mt} \prod_{i=1}^{n_0} y_i^{\tilde{y}_{mti} - \tilde{y}_{mi}}, \\ & \quad m = 1, \dots, M_0. \end{aligned} \quad (6)$$

By using the above notation, one can thus convert (\bar{P}) into the form

$$(P1) : \begin{cases} \min & f(y, s) = \sum_{j=1}^p c_j s_j^{-1} n_j(y) \\ \text{s.t.} & s_j \prod_{i=1}^{n_0} y_i^{(-\hat{y}_{ji})} - \sum_{t=1}^{\hat{T}_j} \hat{\alpha}_{jt} \prod_{i=1}^{n_0} y_i^{r_{jti}} \leq 0, \\ & \quad j = 1, \dots, k, \\ & \sum_{t=1}^{\hat{T}_j} \hat{\alpha}_{jt} \prod_{i=1}^{n_0} y_i^{r_{jti}} - s_j \prod_{i=1}^{n_0} y_i^{(-\hat{y}_{ji})} \leq 0, \\ & \quad j = k + 1, \dots, p, \\ & \sum_{t=1}^{\tilde{T}_m} \tilde{\alpha}_{mt} \prod_{i=1}^{n_0} y_i^{r_{mti}} \leq 0, \quad m = 1, \dots, M_0, \\ & \quad y \in Y, \quad s \in S, \end{cases} \quad (7)$$

where $r_{jti} = \widehat{\gamma}_{jti} - \widehat{\gamma}_{ji} \geq 0$ for $j = 1, \dots, p$ and $r_{mti} = \widehat{\gamma}_{mti} - \widehat{\gamma}_{mi} \geq 0$ for $m = 1, \dots, M_0$. Note that all the exponents are positive in the constraints of problem (P1). Thus, by applying the following exponent transformation

$$\begin{aligned} y_i &= \exp(\eta_i), \quad i = 1, \dots, n_0, \\ s_i &= \exp(\xi_i), \quad i = 1, \dots, p, \end{aligned} \quad (8)$$

to the formulation (P1), letting $N = n_0 + p$ and $z = (\eta, \xi) \in R^N$, and by changing the notation, an equivalent problem of problem (P1) can be then given by

$$(P2) : \begin{cases} \min & \Phi_0(z) \\ \text{s.t.} & \Phi_m(z) \leq 0, \quad m = 1, \dots, p + M_0 \\ & z \in Z^0 = \{z : z_i^l \leq z_i \leq z_i^u, \\ & \quad \forall i = 1, \dots, N\}, \end{cases} \quad (9)$$

where

$$\begin{aligned} \Phi_m(z) &= \sum_{t=1}^{T_m} \alpha_{mt} \exp\left(\sum_{i=1}^N \gamma_{mti} z_i\right), \quad m = 0, \dots, p + M_0, \\ z_i^l &= \ln(y_i^l) \leq \eta_i \leq \ln(y_i^u) = z_i^u, \quad i = 1, \dots, n_0, \\ z_i^l &= \ln L_j \leq \xi_j \leq \ln U_j = z_i^u, \quad j = i - n_0, \\ & \quad j = 1, \dots, p, \quad i = n_0 + 1, \dots, N. \end{aligned} \quad (10)$$

Next, we turn to consider the objective function of (P2). For convenience, for each $m = 0, 1, \dots, p + M_0$, we assume, without loss of generality, that $\alpha_{mt} > 0$ for $t = 1, \dots, J_m$ and $\alpha_{mt} < 0$ for $t = J_m + 1, \dots, T_m$. In addition, some notations are introduced as follows:

$$\begin{aligned} I_t^+ &= \{i \mid \gamma_{0ti} > 0, \quad i = 1, \dots, N\}, \\ I_t^- &= \{i \mid \gamma_{0ti} < 0, \quad i = 1, \dots, N\}, \\ L_t &= \sum_{i \in I_t^-} \gamma_{0ti} z_i^u, \quad U_t = \sum_{i \in I_t^+} \gamma_{0ti} z_i^l, \quad t = 1, \dots, J_0, \\ l_t &= \sum_{i \in I_t^+} \gamma_{0ti} z_i^l, \quad u_t = \sum_{i \in I_t^-} \gamma_{0ti} z_i^u, \quad t = J_0 + 1, \dots, T_0. \end{aligned} \quad (11)$$

Then, by introducing an additional vector $\omega = (\omega_1, \dots, \omega_{T_0})^T \in R^{T_0}$, we can convert the problem (P2) into

$$(P3) : \begin{cases} \min & \sum_{t=1}^{J_0} \alpha_{0t} \exp\left(\sum_{i \in I_t^+} \gamma_{0ti} z_i + \omega_t\right) \\ & + \sum_{t=J_0+1}^{T_0} \alpha_{0t} \exp\left(\sum_{i \in I_t^-} \gamma_{0ti} z_i - \omega_t\right) \\ \text{s. t.} & \sum_{t=1}^{J_m} \alpha_{mt} \exp\left(\sum_{i=1}^{n_0} \gamma_{mti} z_i\right) \\ & + \sum_{t=J_m+1}^{T_m} \alpha_{mt} \exp\left(\sum_{i=1}^{n_0} \gamma_{mti} z_i\right) \leq 0, \\ & \quad m = 1, \dots, p + M_0, \\ & \omega_t - \sum_{i \in I_t^-} \gamma_{0ti} z_i \geq 0, \quad t = 1, \dots, J_0, \\ & \omega_t + \sum_{i \in I_t^+} \gamma_{0ti} z_i \geq 0, \quad t = J_0 + 1, \dots, T_0, \\ & L_t \leq \omega_t \leq U_t, \quad t = 1, \dots, J_0, \\ & -u_t \leq \omega_t \leq -l_t, \quad t = J_0 + 1, \dots, T_0, \\ & \quad z \in Z^0, \end{cases} \quad (12)$$

where Z^0 is defined in (P2).

Note that the objective function of (P3) is increasing and each constrained function is the difference of two increasing functions. The key equivalent result for problems (P2) and (P3) is given by the following Theorem 2.

Theorem 2. z^* is a global optimal solution for problem (P2) if and only if (z^*, ω^*) is a global optimal solution for problem (P3), where

$$\begin{aligned} \omega_t^* &= \sum_{i \in I_t^-} \gamma_{0ti} z_i^*, \quad t = 1, \dots, J_0, \\ \omega_t^* &= -\sum_{i \in I_t^+} \gamma_{0ti} z_i^*, \quad t = J_0 + 1, \dots, T_0. \end{aligned} \quad (13)$$

Proof. The proof of this theorem follows easily from the definitions of problems (P2) and (P3); therefore, it is omitted here. \square

From Theorem 2, for solving problem (P2), we may solve problem (P3) instead. In addition, it is easy to see that the global optimal values of problems (P2) and (P3) are equal. Let $x = (z, \omega) \in R^{N+T_0}$ with $z \in R^N$ and $\omega \in R^{T_0}$ and let

$$n = N + T_0, \quad M = p + M_0 + T_0; \quad (14)$$

then, without loss of generality, by changing the notation, problem (P3) can be rewritten as the following form:

$$(Q) : \min \{F_0(x) \mid F_m^+(x) - F_m^-(x) \leq 0, \quad m = 1, \dots, M, \quad x \in X^0\}, \quad (15)$$

where

$$\begin{aligned}
F_0(x) &= \sum_{t=1}^{J_0} \alpha_{0t} \exp\left(\sum_{i \in I_t^+} \gamma_{0ti} x_i + x_{N+t}\right) \\
&\quad + \sum_{t=J_0+1}^{T_0} \alpha_{0t} \exp\left(\sum_{i \in I_t^-} \gamma_{0ti} x_i - x_{N+t}\right), \\
F_m^+(x) &= \begin{cases} \sum_{t=1}^{J_m} \alpha_{mt} \exp\left(\sum_{i=1}^N \gamma_{mti} x_i\right), \\ m = 1, \dots, p + M_0, \\ x_m, \quad m = p + M_0 + 1, \dots, M, \end{cases} \\
F_m^-(x) &= \begin{cases} -\sum_{t=J_m+1}^{T_m} \alpha_{mt} \exp\left(\sum_{i=1}^N \gamma_{mti} x_i\right), \\ m = 1, \dots, p + M_0, \\ x_{m+N-p-M_0} - \sum_{i \in I_{m-p-M_0}^-} \gamma_{0(m-p-M_0)i} x_i, \\ m = p + M_0 + 1, \dots, p + M_0 + J_0, \\ x_{m+N-p-M_0} + \sum_{i \in I_{m-p-M_0}^+} \gamma_{0(m-p-M_0)i} x_i, \\ m = p + M_0 + J_0 + 1, \dots, M, \end{cases} \\
X^0 &= \{x \in R^n \mid x_i^l \leq x_i \leq x_i^u, \quad i = 1, \dots, n\} \\
&= \left\{ x \in R^n \mid \begin{array}{l} z_i^l \leq x_i \leq z_i^u, \quad i = 1, \dots, N, \\ L_{i-N} \leq x_i \leq U_{i-N}, \quad i = N+1, \dots, N+J_0, \\ -u_{i-N} \leq x_i \leq -l_{i-N}, \quad i = N+J_0+1, \dots, n, \end{array} \right\}. \quad (16)
\end{aligned}$$

Based on the above discussion, to globally solve problem (P), the algorithm to be presented concentrates on solving the problem (Q); then a bound-reduction-bound (BRB) algorithm to be presented will be considered for the problem (Q).

3. Basic Operations

In order to solve globally the problem (Q), the main idea of (BRB) approach to be proposed consists of several basic operations: successively refined partitioning of the feasible set; estimation of lower bound for the optimal value of the objective function over each subset generated by the partitions; and the reduction operations by reducing the size of each partition subset without losing any feasible solution currently still of interest. Next, we begin the establishment of the approach with the basic operations needed in a branch and bound scheme.

Let $X = [a, b] = \{x \mid a_i \leq x_i \leq b_i, \quad i = 1, \dots, n\}$ denote the rectangle or subrectangle of X^0 generated by the algorithm. Consider the following subproblem:

$$\begin{aligned}
Q(X) : \min \{F_0(x) \mid F_m(x) = F_m^+(x) - F_m^-(x) \leq 0, \\ m = 1, \dots, M, \quad x \in X\}. \quad (17)
\end{aligned}$$

3.1. Partition Rule. The critical element in guaranteeing convergence to a minimum of (Q) is the choice of a suitable partitioning strategy. In this paper, we choose the standard branching rule. This method is sufficient to ensure convergence since it derives all the intervals to a singleton for all the variables that are associated with the term that yields the greatest discrepancy in the employed approximation along any infinite branch of the branch-and-bound tree.

Consider any node subproblem identified by rectangle X . The procedure for dividing X into two subrectangles X_+ and X_- can be described as follows.

(i) Let

$$\begin{aligned}
\tau &= \arg \max \{b_i - a_i \mid i = 1, \dots, n\}, \\
\pi_\tau &= \frac{(a_\tau + b_\tau)}{2}. \quad (18)
\end{aligned}$$

(ii) Let

$$\begin{aligned}
X_+ &= \{x \mid a_i \leq x_i \leq b_i, \quad i = 1, \dots, n, \\ &\quad i \neq \tau, \quad \pi_\tau \leq x_\tau \leq b_\tau\}, \\
X_- &= \{x \mid a_i \leq x_i \leq b_i, \quad i = 1, \dots, n, \\ &\quad i \neq \tau, \quad a_\tau \leq x_\tau \leq \pi_\tau\}. \quad (19)
\end{aligned}$$

Through this branching rule, the rectangle X is partitioned into two subrectangles X_+ and X_- .

3.2. Lower Bound. For each rectangle X , we intend to compute a lower bound $LB(X)$ of the optimal value of (Q) over X ; that is, compute a number $LB(X)$ such that

$$\begin{aligned}
LB(X) \leq \min \{F_0(x) \mid F_m(x) \leq 0, \\ m = 1, \dots, M, \quad x \in X\}. \quad (20)
\end{aligned}$$

To ensure convergence, this lower bound must be consistent in the sense that, for any infinite nested sequence of boxes X^k shrinking to a single point x^* ,

$$\lim_{k \rightarrow +\infty} LB(X^k) = F_0(x^*). \quad (21)$$

Clearly, a lower bound is $F_0(a)$, and any bound such that

$$LB(X) \geq F_0(a) \quad (22)$$

will satisfy (21) since $F_0(x)$ is increasing.

Although the bound $F_0(a)$ (for a box $X = [a, b]$) is sufficient for guaranteeing convergence, for a better performance of the lower bound procedure, tighter bounds are often

necessary to achieve reasonable efficiency. For instance, the following procedure may give a better bound.

Consider the subproblem $Q(X)$ and denote the optimal value of problem $Q(X)$ by $V[Q(X)]$. Our main method for computing a valid lower bound $LB(X)$ of $V[Q(X)]$ over $X \subseteq X^0$ is to solve the relaxation linear programming $RLP(X)$ of $Q(X)$ by using a linearization technique. This technique can be realized by underestimating every function $F_0(x)$ and $F_m^+(x)$ and by overestimating every function $F_m^-(x)$, for each $m = 1, \dots, p + M_0$. All the details of this linearization technique for generating the linear relaxation will be given in what follows. For this purpose, let us denote

$$X_{0t} = \begin{cases} \sum_{i \in I_t^+} \gamma_{0ti} x_i + x_{N+t}, & t = 1, \dots, J_0, \\ \sum_{i \in I_t^-} \gamma_{0ti} x_i - x_{N+t}, & t = J_0 + 1, \dots, T_0, \end{cases} \quad (23)$$

$$X_{mt} = \sum_{i=1}^N \gamma_{mti} x_i, \quad t = 1, \dots, T_m;$$

then we have $X_{0t} \in [X_{0t}^l, X_{0t}^u]$ and $X_{mt} \in [X_{mt}^l, X_{mt}^u]$ for any box $X = [a, b]$, where

$$X_{0t}^l = \begin{cases} \sum_{i \in I_t^+} \gamma_{0ti} a_i + a_{N+t}, & t = 1, \dots, J_0, \\ \sum_{i \in I_t^-} \gamma_{0ti} b_i - b_{N+t}, & t = J_0 + 1, \dots, T_0, \end{cases} \quad (24)$$

$$X_{0t}^u = \begin{cases} \sum_{i \in I_t^+} \gamma_{0ti} b_i + b_{N+t}, & t = 1, \dots, J_0, \\ \sum_{i \in I_t^-} \gamma_{0ti} a_i - a_{N+t}, & t = J_0 + 1, \dots, T_0, \end{cases}$$

and for each $m = 1, \dots, p + M_0$,

$$X_{mt}^l = \sum_{i=1}^N \gamma_{mti} a_i, \quad X_{mt}^u = \sum_{i=1}^N \gamma_{mti} b_i, \quad (25)$$

$$t = 1, \dots, T_m.$$

Additionally, let

$$A_{mt} = \frac{\exp(X_{mt}^u) - \exp(X_{mt}^l)}{X_{mt}^u - X_{mt}^l},$$

$$\Delta_{mt}^1(x) = A_{mt} (X_{mt} - X_{mt}^l) + \exp(X_{mt}^l) - \exp(X_{mt}),$$

$$\Delta_{mt}^2(x) = \exp(X_{mt}) - A_{mt} (X_{mt} - \ln A_{mt} + 1), \quad (26)$$

where $m = 0, 1, \dots, p + M_0, t = 1, \dots, T_m$.

Thus, from Theorem 1 in [20], it follows that

$$A_{mt} (X_{mt} - \ln A_{mt} + 1) \leq \exp(X_{mt})$$

$$\leq A_{mt} (X_{mt} - X_{mt}^l) + \exp(X_{mt}^l) \quad (27)$$

and that $\Delta_{mt}^1(x)$ and $\Delta_{mt}^2(x)$ satisfy

$$\max_{x \in X} \Delta_{mt}^1(x) = \max_{x \in X} \Delta_{mt}^2(x)$$

$$= \exp(X_{mt}^l) (1 - W_{mt} + W_{mt} \ln W_{mt}) \quad (28)$$

$$\rightarrow 0 \quad \text{as } \omega_{mt} \rightarrow 0,$$

where

$$\omega_{mt} = X_{mt}^u - X_{mt}^l, \quad W_{mt} = \frac{\exp(\omega_{mt}) - 1}{\omega_{mt}}. \quad (29)$$

Next, we will give the relaxation linear functions of $F_0(x)$, $F_m^+(x)$, and $F_m^-(x)$ over X . Based on the above discussion, it is obvious that we have, for all $x \in X$,

$$F_0(x) \geq \sum_{t=1}^{J_0} \alpha_{0t} (A_{0t} (X_{0t} - \ln A_{0t} + 1))$$

$$+ \sum_{t=J_0+1}^{T_0} \alpha_{0t} (A_{0t} (X_{0t} - X_{0t}^l) + \exp(X_{0t}^l))$$

$$\triangleq LF_0(x),$$

$$F_m^+(x) \geq \sum_{t=1}^{J_m} \alpha_{mt} (A_{mt} (X_{mt} - \ln A_{mt} + 1)) \triangleq LF_m^+(x),$$

$$F_m^-(x) \leq - \sum_{t=J_m+1}^{T_m} \alpha_{mt} (A_{mt} (X_{mt} - X_{mt}^l) + \exp(X_{mt}^l))$$

$$\triangleq UF_m^-(x), \quad (30)$$

where $m = 1, \dots, p + M_0$.

Consequently, we obtain the following linear programming $RLP(X)$ as a linear relaxation of $Q(X)$ over the partition set X :

$$RLP(X) : \begin{cases} \min & LF_0(x) \\ \text{s.t.} & LF_m(x) \leq 0 \quad m = 1, \dots, M, \\ & x \in X, \end{cases} \quad (31)$$

where

$$LF_m(x) = \begin{cases} LF_m^+(x) - UF_m^-(x), & m = 1, \dots, p + M_0, \\ F_m^+(x) - F_m^-(x), & m = p + M_0 + 1, \dots, M. \end{cases} \quad (32)$$

An important property of $RLP(X)$ is that its optimal value $V[RLP(X)]$ satisfies

$$V[RLP(X)] \leq V[Q(X)]. \quad (33)$$

Thus, from (33), the optimal value $V[RLP(X)]$ of $RLP(X)$ provides a valid lower bound for the optimal value $V[Q(X)]$ of $Q(X)$ over $X \subseteq X^0$.

Based on the above result, for any rectangle $X \subseteq X^0$, in order to obtain a lower bound $LB(X)$ of the optimal value $V[Q(X)]$ to subproblem $Q(X)$, we may compute $LB(X)$ such that

$$LB(X) = \max \{V[\text{RLP}(X)], F_0(a)\}, \quad (34)$$

where $V[\text{RLP}(X)]$ is the optimal value of the problem $\text{RLP}(X)$.

Clearly, $LB(X)$ defined in (34) satisfies

$$F_0(a) \leq LB(X) \leq V[Q(X)] \quad (35)$$

and is consistent. It can provide a valid lower bound and guarantee convergence.

3.3. Reduction Operations. Clearly, the smaller the rectangle X is, the tighter the lower bound $LB(X)$ of $Q(X)$ will be and, therefore, the closer the feasible solution of (Q) will be to the corresponding optimal solution. To show this, the next results give two reduction operations, including the reduction cut and the deleting technique, to reduce the size of the partitioned rectangle without losing any feasible solution currently still of interest.

(1) Reduction Cut. At a given stage of the branch and bound algorithm for (Q), for a rectangle $[a, b]$ generated during the partitioning procedure and still of interest, let UB be the objective function value of the best so far feasible solution to problem (Q). Given an $\varepsilon > 0$, we want to find a feasible solution $x \in X$ of (Q) such that $F_0(x) \leq UB - \varepsilon$ or else establish that no such x exists. So, the search for such x can then be restricted to the set $\mathcal{F} \cap [a, b]$, where

$$\mathcal{F} := \{x \mid F_0(x) \leq UB - \varepsilon, F_m(x) \leq 0, m = 1, \dots, M\}. \quad (36)$$

The reduction cut is based on the monotonic structure of the problem (Q). The reduction cut aims at replacing the rectangle $[a, b]$ with a smaller rectangle $[a', b'] \subset [a, b]$ without losing any point $x \in \mathcal{F} \cap [a, b]$, that is, such that $\mathcal{F} \cap [a', b'] = \mathcal{F} \cap [a, b]$. The rectangle $[a', b']$ satisfying this condition is denoted by $\text{red}_\nu[a, b]$ with $\nu = UB - \varepsilon$. To illustrate how $\text{red}_\nu[a, b] = [a', b']$ is deduced by reduction cut, we first define the following functions.

Definition 3. Given two boxes $[a, b]$ and $[a', b']$ with $[a', b'] \subseteq [a, b]$, for $i = 1, \dots, n$, the functions $\varphi^i(\alpha)$ and $\psi^i(\beta) : [0, 1] \rightarrow R^n$ are defined by

$$\begin{aligned} \varphi^i(\alpha) &= b - \alpha(b_i - a_i)e^i, \\ \psi^i(\beta) &= a' + \beta(b_i - a'_i)e^i, \end{aligned} \quad (37)$$

where e^i is a unit vector with 1 at the i th position and 0 everywhere else, $i = 1, \dots, n$.

From the functions $F_m^-(x)$, $F_m^+(x)$, and $F_0(x)$, we have the following result.

Theorem 4. Let $\varepsilon > 0$ be given and let $\nu = UB - \varepsilon$. If $F_0(a) > \nu$ or $F_m^+(a) - F_m^-(b) > 0$ for some $m \in \{1, \dots, M\}$, then $\text{red}_\nu[a, b] = [a', b'] = \emptyset$. Otherwise, $\text{red}_\nu[a, b] = [a', b']$ are given by

$$\begin{aligned} a' &= b - \sum_{i=1}^n \min_{m=1, \dots, M} \{\alpha_m^i\} (b_i - a_i) e^i, \\ b' &= a' + \sum_{i=1}^n \min_{m=0, 1, \dots, M} \{\beta_m^i\} (b_i - a'_i) e^i \end{aligned} \quad (38)$$

satisfying

$$\begin{aligned} \alpha_m^i &= \begin{cases} 1, & \text{if } F_m^-(\varphi^i(1)) \geq F_m^+(a) \\ \alpha & \text{with } F_m^-(\varphi^i(\alpha)) = F_m^+(a), \end{cases} \text{ otherwise,} \\ \beta_m^i &= \begin{cases} 1, & \text{if } F_m^+(\psi^i(1)) \leq F_m^-(b) \\ \beta & \text{with } F_m^+(\psi^i(\beta)) = F_m^-(b), \end{cases} \text{ otherwise.} \end{aligned} \quad (39)$$

Proof. (i) By the increasing property of $F_0(x)$, $F_m^+(x)$, and $F_m^-(x)$, if $F_0(a) > \nu$, then $F_0(x) \geq F_0(a) > \nu$ for every $x \in [a, b]$. If there exists $m \in \{1, \dots, M\}$ such that $F_m^+(a) - F_m^-(b) > 0$, then $F_m(x) = F_m^+(x) - F_m^-(x) \geq F_m^+(a) - F_m^-(b) > 0$ for every $x \in [a, b]$. In both cases, $\mathcal{F} \cap [a, b] = \emptyset$.

(ii) Given any point $x \in [a, b]$ satisfying

$$F_0(x) \leq \nu, \quad F_m^+(x) - F_m^-(x) \leq 0, \quad m = 1, \dots, M, \quad (40)$$

we will show that $x \in [a', b']$. Let

$$\begin{aligned} \alpha_{m'}^i &= \min \{\alpha_m^i \mid m = 1, \dots, M\}, \\ \beta_{m''}^i &= \min \{\beta_m^i \mid m = 0, 1, \dots, M\}. \end{aligned} \quad (41)$$

Firstly, we will show that $x \geq a'$. If $x \not\geq a'$, then there exists index i such that

$$x_i < a'_i = b_i - \alpha_{m'}^i (b_i - a_i), \quad \text{i.e.,} \quad (42)$$

$$x_i = b_i - \alpha (b_i - a_i) \quad \text{with } \alpha_{m'}^i < \alpha \leq 1.$$

We consider the following two cases.

Case 1. If $\alpha_{m'}^i = 1$, then, from (42), we have $x_i < a'_i = b_i - \alpha_{m'}^i (b_i - a_i) = a_i$, conflicting with $x \in [a, b]$; that is, $x_i \geq a_i$.

Case 2. If $0 \leq \alpha_{m'}^i < 1$, the function $\Phi_{m'}^i(\alpha) = F_{m'}^-(\varphi^i(\alpha)) - F_{m'}^+(a)$ must be strictly decreasing in single variable α over the interval $[0, 1]$. If the function $\Phi_{m'}^i(\alpha)$ is not strictly decreasing in single variable α , we get that $\Phi_{m'}^i(\alpha)$ must be a constant over the interval $[0, 1]$. In this case, we have

$$\Phi_{m'}^i(1) = \Phi_{m'}^i(0) = F_{m'}^-(b) - F_{m'}^+(a) \geq 0. \quad (43)$$

It follows from the definition of $\alpha_{m'}^i$ that $\alpha_{m'}^i = 1$, contradicting with $0 \leq \alpha_{m'}^i < 1$.

Since the function $\Phi_{m'}^i(\alpha)$ is strictly decreasing, it follows, from (42) and the definition of $\alpha_{m'}^i$ that $F_{m'}^-(b - (b_i - x_i)e^i) - F_{m'}^+(a) = \Phi_{m'}^i(\alpha) < \Phi_{m'}^i(\alpha_{m'}^i) = 0$. Hence, $F_{m'}^-(b - (b_i - x_i)e^i) < F_{m'}^+(a)$. In addition, since $F_{m'}^-(x)$ is an increasing function in n -dimension variable x and $x \leq b - (b_i - x_i)e^i$, we have

$$F_{m'}^-(x) \leq F_{m'}^-(b - (b_i - x_i)e^i) < F_{m'}^+(a), \quad (44)$$

conflicting with $F_{m'}^-(x) \geq F_{m'}^+(x) \geq F_{m'}^+(a)$.

Based on the above discussion, we have $x \geq a'$; that is, $x \in [a', b]$ in either case.

Secondly, we also can show from $x \in [a', b]$ that $x \leq b'$; that is, $x \in [a', b']$. Supposing that $x \not\leq b'$, then there exists some i such that

$$x_i > b'_i = a'_i + \beta_{m''}^i (b_i - a'_i); \quad (45)$$

that is, there exists β such that

$$x_i = a'_i + \beta (b_i - a'_i), \quad \beta_{m''}^i < \beta \leq 1. \quad (46)$$

By the definition of $\beta_{m''}^i$, there are the following two cases to consider.

Case 1. If $\beta_{m''}^i = 1$, then, from (45), we have $x_i > b'_i = a'_i + (b_i - a'_i) = b_i$, conflicting with $x \in [a', b]$; that is, $x_i \leq b_i$.

Case 2. If $0 \leq \beta_{m''}^i < 1$, the function $\Psi_{m''}^i(\beta) = F_m^+(\psi^i(\beta)) - F_m^-(b)$ is strictly increasing in single variable β . If $\Psi_{m''}^i(\beta)$ is not strictly increasing in β , we get that $\Psi_{m''}^i(\beta)$ must be a constant over $[0, 1]$. In this case, we have

$$F_0(a') - \nu \leq 0 \quad (47)$$

or

$$\Psi_{m''}^i(1) = \Psi_{m''}^i(0) = F_{m''}^+(a') - F_{m''}^-(b) \leq 0. \quad (48)$$

It follows from the definition of $\beta_{m''}^i$ that $\beta_{m''}^i = 1$, which is a contradiction with $0 \leq \beta_{m''}^i < 1$.

Since the function $\Psi_{m''}^i(\beta)$ is strictly increasing, from (46) and the definition of $\beta_{m''}^i$, it implies that

$$F_0(\psi^i(\beta)) - \nu > F_0(\psi^i(\beta_0^i)) - \nu = 0 \quad (49)$$

or

$$F_{m''}^+(\psi^i(\beta)) - F_{m''}^-(b) = \Psi_{m''}^i(\beta) > \Psi_{m''}^i(\beta_{m''}^i) = 0. \quad (50)$$

Assume that (49) holds; we can derive, from (46), that

$$F_0(a' + (x_i - a'_i)e^i) = F_0(\psi^i(\beta)) > \nu. \quad (51)$$

It follows from $x \geq a' + (x_i - a'_i)e^i$ and $F_0(x)$ that

$$F_0(x) \geq F_0(a' + (x_i - a'_i)e^i) > \nu, \quad (52)$$

conflicting with $F_0(x) \leq \nu$.

If (50) holds, we obtain, from (46), that

$$F_{m''}^+(a' + (x_i - a'_i)e^i) = F_{m''}^+(\psi^i(\beta)) > F_{m''}^-(b). \quad (53)$$

Since $x \geq a' + (x_i - a'_i)e^i$ and $F_{m''}^+(x)$ is increasing, we have

$$F_{m''}^+(x) \geq F_{m''}^+(a' + (x_i - a'_i)e^i) > F_{m''}^-(b). \quad (54)$$

It is a contradiction with $F_{m''}^+(x) \leq F_{m''}^-(x) \leq F_{m''}^-(b)$.

From the above results, we must have $x \leq b'$; that is, $x \in [a', b']$ in both cases and this ends the proof. \square

Remark 5. α_m^i and β_m^i given in Theorem 4 must exist and be unique, since the functions $F_0(x)$, $F_m^+(x)$, and $F_m^-(x)$ are all continuous and increasing.

Remark 6. In order to obtain $\text{red}_\nu[a, b]$, the computation of α_m^i and β_m^i is more easily implementable than that of (2.4) and (2.5) in [28]. This is because the latter is computed by solving the nonlinear nonconvex programming problem, but the former is involved in solving the single variable equation with monotonicity.

(2) *Deleting Technique.* For any $x \in X = (X_i)_{n \times 1}$ with $X_i = [a_i, b_i]$ ($i = 1, \dots, n$), without loss of generality, we assume that the relaxation linear problem $\text{RLP}(X)$ can be rewritten as

$$\text{RLP}(X) : \begin{cases} \min & \sum_{i=1}^n \lambda_{0i} x_i + \mu_0 \\ \text{s.t.} & \sum_{i=1}^n \lambda_{ji} x_i + \mu_j \leq 0, \quad j = 1, \dots, M, \\ & x \in X \subseteq X^0 \end{cases} \quad (55)$$

and let UB be a known upper bound of the optimum of $Q(X^0)$. Define

$$\text{RL}_j = \sum_{i=1}^n \min \{ \lambda_{ji} a_i, \lambda_{ji} b_i \} + \mu_j, \quad j = 0, 1, \dots, M, \quad (56)$$

$$\rho_i = \frac{\text{UB} - \text{RL}_0 + \min \{ \lambda_{0i} a_i, \lambda_{0i} b_i \}}{\lambda_{0i}} \quad \text{with } \lambda_{0i} \neq 0, \quad (57)$$

$$\tau_{ji} = \frac{-\text{RL}_j + \min \{ \lambda_{ji} a_i, \lambda_{ji} b_i \}}{\lambda_{ji}} \quad \text{with } \lambda_{ji} \neq 0, \quad (58)$$

$$j = 1, \dots, M,$$

where $i = 1, \dots, n$.

Theorem 7. For any $X = (X_i)_{n \times 1} \subseteq X^0$, if $\text{RL}_0 > \text{UB}$, then there exists no optimal solution of the problem $Q(X^0)$ over X . Otherwise, if $\lambda_{0h} > 0$ and $\rho_h < b_h$, for some $h \in \{1, \dots, n\}$, then there is no optimal solution of the problem $\text{P4}(X^0)$ over the subrectangle X_a ; conversely, if $\lambda_{0h} < 0$ and $\rho_h > a_h$, for

some $h \in \{1, \dots, n\}$, then there does not exist optimal solution of P4(X^0) over X_b , where

$$\begin{aligned} X_a &= (X_{ai})_{n \times 1} \subseteq X^0 \quad \text{with} \\ X_{ai} &= \begin{cases} X_i, & \text{if } i \neq h, \\ (\rho_h, b_h] \cap X_h, & \text{if } i = h, \end{cases} \\ X_b &= (X_{bi})_{n \times 1} \subseteq X^0 \quad \text{with} \\ X_{bi} &= \begin{cases} X_i, & \text{if } i \neq h, \\ [a_h, \rho_h) \cap X_h, & \text{if } i = h. \end{cases} \end{aligned} \quad (59)$$

Proof. The proof is similar to Theorem 2 in [27]; it is omitted here. \square

Theorem 8. For any $X = (X_i)_{n \times 1} \subseteq X^0$, if $RL_j(x) > 0$, for some $j \in \{1, \dots, M\}$, then there exists no feasible solution of problem P4(X^0) over X . Otherwise, consider the following two cases: if there exists some index $h \in \{1, \dots, n\}$ satisfying $\lambda_{jh} > 0$ and $\tau_{jh} < b_h$, for some $j \in \{1, \dots, M\}$, then there is no feasible solution of the problem P4(X^0) over X_c ; conversely, if $\lambda_{jh} < 0$ and $\tau_{jh} > a_h$, for some $j \in \{1, \dots, M\}$ and $h \in \{1, \dots, n\}$, then there exists no feasible solution of the problem P4(X^0) over X_d , where

$$\begin{aligned} X_c &= (X_{ci})_{n \times 1} \subseteq X^0 \quad \text{with} \\ X_{ci} &= \begin{cases} X_i, & \text{if } i \neq h, \\ (\tau_{jh}, b_h] \cap X_h, & \text{if } i = h, \end{cases} \\ X_d &= (X_{di})_{n \times 1} \subseteq X^0 \quad \text{with} \\ X_{di} &= \begin{cases} X_i, & \text{if } i \neq h, \\ [a_h, \tau_{jh}) \cap X_h, & \text{if } i = h. \end{cases} \end{aligned} \quad (60)$$

Proof. The proof is similar to Theorem 3 in [27]; it is omitted here. \square

By Theorems 7 and 8, we can give a new deleting technique to reject some regions in which the globally optimal solution of Q(X^0) does not exist. Let $X = (X_i)_{n \times 1}$ with $X_i = [a_i, b_i]$ ($i = 1, \dots, n$) be any subrectangle of X^0 . The content of deleting technique is summarized as follows.

(S1) *Optimality Rule.* Compute RL_0 in (56). If $RL_0 > UB$, let $X = \emptyset$; otherwise, compute ρ_i ($i = 1, \dots, n$) in (57). If $\lambda_{0h} > 0$ and $\rho_h < b_h$, for some $h \in \{1, \dots, n\}$, then let $b_h = \rho_h$ and $X = (X_i)_{n \times 1}$ with $X_i = [a_i, b_i]$ ($i = 1, \dots, n$). If $\lambda_{0h} < 0$ and $\rho_h > a_h$, for some $h \in \{1, \dots, n\}$, then let $a_h = \rho_h$ and $X = (X_i)_{n \times 1}$ with $X_i = [a_i, b_i]$ ($i = 1, \dots, n$).

(S2) *Feasibility Rule.* For any $j = 1, \dots, M$, compute RL_j in (56). If $RL_j > 0$, for some $j \in \{1, \dots, M\}$, then let $X = \emptyset$; otherwise, compute τ_{ji} in (58) ($j = 1, \dots, M$, $i = 1, \dots, n$). If $\lambda_{jh} > 0$ and $\tau_{jh} < b_h$, for some $j \in \{1, \dots, M\}$ and $h \in \{1, \dots, n\}$, then let $b_h = \tau_{jh}$ and $X = (X_i)_{n \times 1}$ with $X_i = [a_i, b_i]$ ($i = 1, \dots, n$). If $\lambda_{jh} < 0$ and $\tau_{jh} > a_h$, for some

$j \in \{1, \dots, M\}$ and $h \in \{1, \dots, n\}$, then let $a_h = \tau_{jh}$ and $X = (X_i)_{n \times 1}$ with $X_i = [a_i, b_i]$ ($i = 1, \dots, n$).

This deleting technique provides a possibility to cut away all or a large part of the subrectangle X which is currently investigated by the algorithm procedure.

4. Algorithm and Its Convergence

Now, a branch-reduce-bound (BRB) algorithm is developed to solve the problem (Q) based on the former discussion. This method needs to solve a sequence of (RLP) problems over partitioned subsets of X^0 .

The BRB algorithm is based on partitioning the rectangle X^0 into subrectangles, each concerned with a node of the branch-and-bound tree. Hence, at any stage k of the algorithm, suppose that we have a collection of active nodes denoted by \mathcal{T}_k , where each node is associated with a rectangle $X \subseteq X^0$ and $\forall X \in \mathcal{T}_k$. For each such node $X = [a, b]$, we will compute a lower bound $LB(X)$ of the optimal objective function value of (P4) via the optimal value of the RLP(X) and $F_0(a)$, so the lower bound of the optimal value of (P4) at stage k is given by $\min\{LB(X), \forall X \in \mathcal{T}_k\}$. We now select an active node to subdivide its associated rectangle into two subrectangles according to branch rule described in the Section 3.1. For each new node, reduce it and then compute the lower bound as before. At the same time, if necessary, we will update the upper bound UB_k . Upon fathoming any nonimproving node, we obtain a collection of active nodes for the next stage, and this process is repeated until convergence is obtained.

4.1. Algorithm Statement

Step 1 (initialization). Choose the convergence tolerance $\varepsilon > 0$. Let $\mathcal{P}_0 = \{X^0\}$ and $\mathcal{T}_0 = \{X^0\}$. If some feasible solutions are available, add them to \mathcal{F} and let $UB_0 = \min\{F_0(x) \mid x \in \mathcal{F}\}$; otherwise, let $\mathcal{F} = \emptyset$ and $UB_0 = +\infty$. Set $k = 0$.

Step 2 (reduction). (i) Apply the reduction cut described in Section 3.3 to each box $[a, b] \in \mathcal{P}_k$. Let $\mathcal{P}'_k = \{\text{red}_\nu[a, b] \mid [a, b] \in \mathcal{P}_k\}$ with $\nu = UB_k - \varepsilon$.

(ii) If $\mathcal{P}'_k \neq \emptyset$, for each box $[a, b] \in \mathcal{P}'_k$, we use the deleting technique in Section 3.3 to cut away X and denote the left still as \mathcal{P}''_k .

Step 3 (bounding). If $\mathcal{P}''_k \neq \emptyset$, begin to do, for each $X = [a, b] \in \mathcal{P}''_k$, the following.

(i) Solve the problem RLP(X) to obtain the optimal solution $x(X)$ and the optimal value $V[\text{RLP}(X)]$. If $x(X)$ is feasible to problem (Q), then set $\mathcal{F} = \mathcal{F} \cup \{x(X)\}$. Let $LB(X) = \max\{V[\text{RLP}(X)], F_0(a)\}$.

(ii) If $F_m(a) \leq 0$ for every $m = 1, \dots, M$, then set $\mathcal{F} = \mathcal{F} \cup \{a\}$.

(iii) If $\mathcal{F} \neq \emptyset$, define the new upper bound $UB_k = \min\{F_0(x) \mid x \in \mathcal{F}\}$, and the best known feasible point is denoted by $x^k = \arg \min\{F_0(x) \mid x \in \mathcal{F}\}$. Set $\mathcal{T}_k = (\mathcal{T}_k \setminus X^k) \cup \mathcal{P}''_k$.

Step 4 (convergence checking). Set $\mathcal{T}_{k+1} = \mathcal{T}_k \setminus \{X | \text{LB}(X) > \text{UB}_k - \varepsilon, X \in \mathcal{T}_k\}$.

If $\mathcal{T}_{k+1} = \emptyset$, then stop; if $\text{UB}_k = +\infty$, the problem is infeasible; otherwise, UB_k is the optimal value and x^k is the optimal solution. Otherwise, select an active node $X^{k+1} = \arg \min\{\text{LB}(X) | X \in \mathcal{T}_{k+1}\}$ for further consideration and let $\text{LB}_{k+1} = \text{LB}(X^{k+1})$.

Step 5 (branching). Divide X^{k+1} into two new subrectangles using the branching rule and let \mathcal{P}_{k+1} be the collection of these two subrectangles. Set $k = k + 1$ and return to Step 2.

4.2. Convergence Analysis. In this subsection, we give the convergence of the proposed algorithm.

Theorem 9. *If the presented algorithm finishes at finite step, when it stops, x^k must be a global optimum solution of the problem (Q). Otherwise, for any infinite branch of initial rectangle domain, an infinite partitioned rectangle sequence X^s will be produced and any accumulation point of which must be a global optimal solution of the initial problem (Q).*

Proof. Assume that this algorithm terminates finitely at some stage $k, k \geq 0$. Thus, when the algorithm terminates, it follows that $\text{UB}_k - \text{LB}_k \leq \varepsilon$. By Steps 2 and 4 of the investigated algorithm, there exists a feasible solution x_k of the problem (Q) satisfying $F_0(x_k) = \text{UB}_k$, which implies that $F_0(x^k) - \text{LB}_k \leq \varepsilon$. Let F_0^* be the optimal value of the problem (Q); then, by the structure of this algorithm, we have $\text{LB}_k \leq F_0^*$. Since x^k is a feasible solution of the problem (Q), $F_0^* \leq F_0(x^k)$. Connecting the above inequalities, we have $F_0^* \leq F_0(x^k) \leq \text{LB}_k + \varepsilon \leq F_0^* + \varepsilon$; that is, $F_0^* \leq F_0(x^k) \leq F_0^* + \varepsilon$. Thus, x^k is an ε -global optimum solution of the problem (Q).

If the algorithm is infinite, it generates an infinite sequence $\{X^k\}$ such that a subsequence $\{X^{k_l}\}$ of $\{X^k\}$ satisfies $X^{k_{l+1}} \subset X^{k_l}$ for $l = 1, 2, \dots$. Thus, it follows from [28, 29] that this rectangle subdivision is exhaustive. Hence, for every iteration $k = 0, 1, 2, \dots$, by design of the algorithm, there is at least an infinite subsequence $\{\text{LB}_{k_l}\}$ of $\{\text{LB}_k\}$ such that

$$\text{LB}_{k_l} \leq \min_{x \in X} F_0(x), \quad X^{k_l} \in \arg \min_{X \in \mathcal{T}_{k_l}} \text{LB}(X), \quad (61)$$

$$x^{k_l} = x(X^{k_l}) \in X^{k_l} \subseteq X^0.$$

Since $\{\text{LB}_{k_l}\}$ is a nondecreasing sequence bounded above by $\min_{x \in D} F_0(x)$, where D is the feasible set to problem (Q), this guarantees the existence of the limit $\lim_{l \rightarrow \infty} \text{LB}_{k_l} := \text{LB}$ and $\text{LB} \leq \min_{x \in X} F_0(x)$. Since $\{x^{k_l}\}$ is an infinite sequence on a compact set X^0 , there exists a convergent subsequence $\{x^q\}$ of $\{x^{k_l}\}$ satisfying $\lim_{q \rightarrow \infty} x^q = \hat{x}$, $x^q \in X^q$ and $\text{LB}_q = \text{LB}(X^q) = \text{LF}_0(x^q)$, where $\{X^q\}$ is a subsequence of $\{X^{k_l}\}$. By using Theorem 1 and Lemma 1 of [25], we see that the linear subfunctions LF_m ($m = 0, 1, \dots, M$) used in the problem $\text{RLP}(X)$ are strongly consistent on X^0 . Thus, $\lim_{q \rightarrow \infty} \text{LB}_q = \lim_{q \rightarrow \infty} \text{LF}_0(x^q) = \text{LF}_0(\lim_{q \rightarrow \infty} x^q) = \text{LF}_0(\hat{x}) = \text{LB}$. All that remains is to show that $\hat{x} \in D$. Since X^0 is a closed set,

TABLE 1: The numerical results for upper and lower bounds of $d_j(y)$.

Example	Reference	L_1	U_1	L_2	U_2
Example 1	BRB	2	3.25	4	13
	[25]	2	4	4	16
Example 2	BRB	2	3	4	12.38
	[25]	2	4	4	16
Example 3	BRB	3	17.337	2	4.67
	[25]	2	149.2961	3	601
Example 4	BRB	2	3	4	12.38
Example 5	BRB	2	4.67	3	17.337

it follows that $\hat{x} \in X^0$. Suppose that $\hat{x} \notin D$. Then there exists some $F_j, j \in \{1, \dots, M\}$, such that $F_j(\hat{x}) = \delta > 0$. Since LF_j is continuous, by Theorem 1 and Lemma 1 of [25], we have $\text{LF}_j(x^q) \rightarrow F_j(\hat{x})$ as $q \rightarrow \infty$, that is, $\exists q_\delta$ such that $|\text{LF}_j(x^q) - F_j(\hat{x})| < \delta$ as $q > q_\delta$, and so when $q > q_\delta$, $\text{LF}_j(x^q) > 0$ implies that the problem $\text{RLP}(X)$ is infeasible. This contradicts the assumption that $x^q = x(X^q)$ is the optimal solution to $\text{RLP}(X)$. Therefore, $\hat{x} \in D$; that is, $\text{LB} = F_0(\hat{x}) = \min_{x \in D} F_0(x)$. \square

5. Numerical Experiments

There are two computational issues that may arise in using the suggested implementations of the global algorithm.

The first computational issue is concerned with the fact that we need to obtain the positive scalars L_j and U_j such that $0 < L_j \leq d_j(y) \leq U_j$ for all $y \in Y$ before using the suggested implementations of the algorithm (see Section 2). Actually, L_j and U_j are available through solving the following two problems:

$$\begin{aligned} \text{(PP1)} : & \begin{cases} \min & d_j(y) \\ \text{s.t.} & g_m(y) \leq 0, \quad m = 1, 2, \dots, M_0, \\ & 0 < y_i^l \leq y_i \leq y_i^u < \infty, \quad i = 1, 2, \dots, n_0, \end{cases} \\ \text{(PP2)} : & \begin{cases} \max & d_j(y) \\ \text{s.t.} & g_m(y) \leq 0, \quad m = 1, 2, \dots, M_0, \\ & 0 < y_i^l \leq y_i \leq y_i^u < \infty, \quad i = 1, 2, \dots, n_0. \end{cases} \end{aligned} \quad (62)$$

The problems (PP1) and (PP2) are special cases of the original problem (P), by using the proposed algorithm; therefore, the values of L_j and U_j can be obtained directly without requiring other special procedure (see [24, 25]). Furthermore, the interval $[L_j, U_j]$ obtained is tighter than one by making use of Bernstein algorithm in [24, 25] (see Table 1) so that the convergence of the algorithm may be improved.

The second computational issue concerns the lower bound computing process. From Section 3, each lower bound in the algorithm is computed by solving a relaxation linear programming of the form of problem (RLP). Here, we adopt the simplex algorithm to solve the relaxation linear programming. So the implement of the proposed global algorithm will depend upon the simplex algorithm.

TABLE 2: The numerical results for Examples 1–5.

Example		Optimal solution	Optimal value	Iter	Time	ϵ
Example 1	BRB	(1.0, 1.7438231783465)	-4.060819175	1765	1.97	10^{-8}
	[25]	(1.0, 1.743823132)	-4.060819161	2638	16.23	10^{-8}
Example 2	BRB	(1.618033989, 1.0)	1.16653785326	197	0.42	10^{-8}
	[25]	(1.618033989, 1.0)	1.166537841203	420	3.52	10^{-8}
Example 3	BRB	(2.698690689, 1.207585549)	-2.33221836	5835	8.18	10^{-7}
	[25]	(2.698690670, 1.20758556)	-1.96149893	15243	130.62	10^{-7}
Example 4	BRB	(1, 1)	3.3333	64	0.1563	10^{-3}
	[24]	(1, 1)	3.3333	262	2	10^{-3}
Example 5	BRB	(1, 1)	5.5167	149	0.2	10^{-3}
	[24]	(1, 1)	5.5167	280	1.6	10^{-3}

We now report our numerical experiments through five test examples and some randomly produced problems to demonstrate the performance of the proposed optimization algorithm. The algorithm is coded in Compaq Visual Fortran, and all test problems are implemented in an Athlon(tm) CPU 2.31 GHz with 960 MB RAM microcomputer. The numerical results for all test problems are summarized in Tables 2 and 3. Numerical results show that the proposed algorithm can globally solve the problem (P) effectively.

In the following tables, the notations have been used for column headers: Iter: number of algorithm iterations; Time: execution time in seconds; and ϵ : convergence tolerance. And for row headers, BRB denotes the corresponding numerical results in the proposed BRB algorithm.

Example 1 (see [25]). Consider

$$\begin{aligned}
 \min \quad & h(x) = -\frac{-x_1^2 + 3x_1 - x_2^2 + 3x_2 + 3.5}{x_1 + 1} \\
 & - \frac{x_2}{x_1^2 - 2x_1 + x_2^2 - 8x_2 + 20} \\
 \text{s.t.} \quad & 2x_1 + x_2 \leq 6, \\
 & 3x_1 + x_2 \leq 8, \\
 & -x_1 + x_2 \geq -1, \\
 & 1 \leq x_1 \leq 3, \quad 1 \leq x_2 \leq 3.
 \end{aligned} \tag{63}$$

Example 2 (see [25]). Consider

$$\begin{aligned}
 \min \quad & h(x) = 0.25 \frac{-x_1^2 + 3x_1 + 2x_2^2 + 3x_2 + 3.5}{x_1 + 1} \\
 & + 1.75 \frac{x_2}{x_1^2 - 2x_1 + x_2^2 - 8x_2 + 20} \\
 \text{s.t.} \quad & 3x_1 + x_2 \leq 8, \\
 & x_1 - x_1^{-1}x_2 \leq 1, \\
 & 2x_1x_2^{-1} + x_2 \leq 6, \\
 & 1 \leq x_1 \leq 3, \quad 1 \leq x_2 \leq 3.
 \end{aligned} \tag{64}$$

Example 3 (see [25]). Consider

$$\begin{aligned}
 \min \quad & h(x) = -1.35 \frac{x_1^2x_2^{0.5} - 2x_1x_2^{-1} + x_2^2 - 2.8x_1^{-1}x_2 + 7.5}{x_1x_2^{1.5} + 1} \\
 & + 12.99 \frac{x_2 + 0.1}{x_1^2x_2^{-1} - 3x_1^{-1} + 2x_1x_2^2 - 9x_2^{-1} + 12} \\
 \text{s.t.} \quad & 2x_1^{-1} + x_1x_2 \leq 4, \\
 & x_1 + 3x_1^{-1}x_2 \leq 5, \\
 & x_1^2 - 3x_2^3 \leq 2, \\
 & 1 \leq x_1 \leq 3, \quad 1 \leq x_2 \leq 3.
 \end{aligned} \tag{65}$$

Example 4 (see [24]). Consider

$$\begin{aligned}
 \min \quad & h(x) = \frac{-x_1^2 + 3x_1 - x_2^2 + 3x_2 + 3.5}{x_1 + 1} \\
 & - \frac{x_2}{x_1^2 - 2x_1 + x_2^2 - 8x_2 + 20} \\
 \text{s.t.} \quad & 3x_1 + x_2 \leq 8, \\
 & x_1 - x_1^{-1}x_2 \leq 1, \\
 & 2x_1x_2^{-1} + x_2 \leq 6, \\
 & 1 \leq x_1 \leq 3, \quad 1 \leq x_2 \leq 3.
 \end{aligned} \tag{66}$$

Example 5 (see [24]). Consider

$$\begin{aligned}
 \min \quad & h(x) = \frac{-x_1^2x_2^{0.5} + 2x_1x_2^{-1} - x_2^2 + 2.8x_1^{-1}x_2 + 7.5}{x_1x_2^{1.5} + 1} \\
 & + \frac{x_2 + 0.1}{x_1^2x_2^{-1} - 3x_1^{-1} + 2x_1x_2^2 - 9x_2^{-1} + 12}
 \end{aligned}$$

$$\begin{aligned}
 \text{s.t.} \quad & 2x_1^{-1} + x_1x_2 \leq 4, \\
 & x_1 + 3x_1^{-1}x_2 \leq 5, \\
 & x_1^2 - 3x_2^3 \leq 2, \\
 & 1 \leq x_1 \leq 3, \quad 1 \leq x_2 \leq 3.
 \end{aligned} \tag{67}$$

From Table 1, by using the proposed method, the upper and lower bounds of $d_j(y)$ are better than other methods [25]; that is, the values of L_1 and L_2 are all bigger than other methods and the values of U_1 and U_2 are all smaller than other methods.

From Table 2, numerical results show that the computational efficiency is obviously improved by using the proposed algorithm in the number of iterations and the overall execution time of the algorithm, compared with other methods [24, 25].

Additionally, we choose the following problem to test our algorithm further, which is generated randomly:

$$\begin{aligned}
 \text{(Problem 1):} \quad & \left\{ \begin{array}{l} \min \quad \frac{\sum_{i=1}^q p_i^1 \prod_{i \in N_i^1} x_j^{\alpha_{ij}^1}}{\sum_{i=1}^q p_i^2 \prod_{i \in N_i^2} x_j^{\alpha_{ij}^2}} \\ \quad + \frac{\sum_{i=1}^q p_i^3 \prod_{i \in N_i^3} x_j^{\alpha_{ij}^3}}{\sum_{i=1}^q p_i^4 \prod_{i \in N_i^4} x_j^{\alpha_{ij}^4}} \\ \text{s.t.} \quad Ax \leq b, \\ \quad \sum_{i=1}^n x_i = 10, \\ \quad 0 < x_i \leq 10, \quad i = 1, \dots, n, \end{array} \right. \tag{68}
 \end{aligned}$$

where q is an integer number (e.g., q is taken to be n in Table 3), for each $k = 1, 2, 3, 4$, $p_i^k \in (1, 2)$, $N_i^k \subset \{1, 2, \dots, n\}$ with $1 \leq |N_i^k| \leq 4$, each element of N_i^k is randomly generated from $\{1, 2, \dots, n\}$, and α_{ij}^k are randomly generated from $\{1, 2, 3\}$. The elements of A and b with $A \in R^{m \times n}$ and $b \in R^m$ are generated by using random number in the intervals $(0, 1)$ and $(0, 10)$, respectively.

For the above test problem, the convergence tolerance parameter is set as $\varepsilon = 10^{-2}$ and $q = 3$. Numerical results are summarized in Table 3, where the average number of iterations, average number of list nodes, and average CPU times (seconds) are obtained by running the BRB algorithm for 20 times to this problem.

It is seen from Table 3 that the size of n (the number of variable) is the main factor affecting the performance of the algorithm. This is mainly because we have to take much time to compute the bound of introduced variables, which is increased as the size of the number of variable increasing. However, due to the constrained functions value in reduction operation and linear relaxation, the CPU time also increases, while m (the number of inequality constrained) is increasing but not as sharply as n .

TABLE 3: The numerical results for the random problem.

n	m	Ave. Iter	Ave. L	Ave. CPU (s)
4	5	94	93	40.35
4	7	94	93	40.36
4	10	98	95	40.37
6	5	638	637	77.87
6	7	675	674	78.18
6	10	682	677	80.85
10	5	528	527	232.5
10	7	554	546	232.35
10	10	598	583	235.6

Conflict of Interests

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

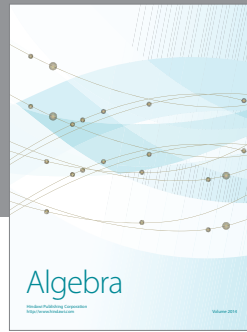
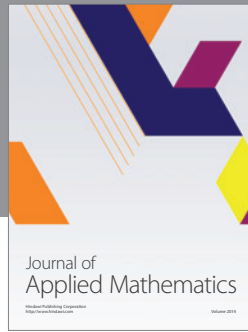
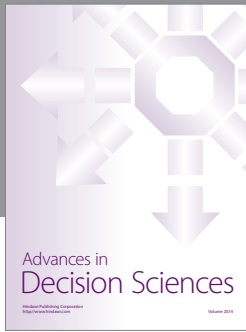
Acknowledgment

The research is supported by the National Natural Science Foundation of China (11171094 and 11171368).

References

- [1] C. D. Maranas, I. P. Androulakis, C. A. Floudas, A. J. Berger, and J. M. Mulvey, "Solving long-term financial planning problems via global optimization," *Journal of Economic Dynamics & Control*, vol. 21, no. 8-9, pp. 1405-1425, 1997.
- [2] H. M. Markowitz, *Portfolio Selection*, Basil Blackwell, Oxford, UK, 2nd edition, 1991.
- [3] I. Quesada and I. E. Grossmann, "Alternative bounding approximations for the global optimization of various engineering design problems," in *Global Optimization in Engineering Design*, vol. 9 of *Nonconvex Optimization and Its Applications*, pp. 309-331, Kluwer Academic Publishers, Norwell, Mass, USA, 1996.
- [4] I. M. Stancu-Minasian, *Fractional Programming*, vol. 409 of *Mathematics and Its Applications*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.
- [5] H. Konno and H. Watanabe, "Bond portfolio optimization problems and their applications to index tracking: a partial optimization approach," *Journal of the Operations Research Society of Japan*, vol. 39, no. 3, pp. 295-306, 1996.
- [6] N. K. Jha, "Geometric programming based robot control design," *Computers and Industrial Engineering*, vol. 29, no. 1-4, pp. 631-635, 1995.
- [7] K. Das, T. K. Roy, and M. Maiti, "Multi-item inventory model with quantity-dependent inventory costs and demand-dependent unit cost under imprecise objective and restrictions: A geometric programming approach," *Production Planning and Control*, vol. 11, no. 8, pp. 781-788, 2000.
- [8] J. C. Choi and D. L. Bricker, "Effectiveness of a geometric programming algorithm for optimization of machining economics models," *Computers and Operations Research*, vol. 23, no. 10, pp. 957-961, 1996.
- [9] R. Yamamoto and H. Konno, "An efficient algorithm for solving convex-convex quadratic fractional problems," *Journal of Optimization Theory and Applications*, vol. 133, no. 2, pp. 241-255, 2007.

- [10] H. P. Benson, "A simplicial branch and bound duality-bounds algorithm for the linear sum-of-ratios problem," *European Journal of Operational Research*, vol. 182, no. 2, pp. 597–611, 2007.
- [11] P. Shen, Y. Chen, and Y. Ma, "Solving sum of quadratic ratios fractional programs via monotonic function," *Applied Mathematics and Computation*, vol. 212, no. 1, pp. 234–244, 2009.
- [12] H. Jiao, Y. Guo, and P. Shen, "Global optimization of generalized linear fractional programming with nonlinear constraints," *Applied Mathematics and Computation*, vol. 183, no. 2, pp. 717–728, 2006.
- [13] S.-J. Qu, K.-C. Zhang, and J.-K. Zhao, "An efficient algorithm for globally minimizing sum of quadratic ratios problem with nonconvex quadratic constraints," *Applied Mathematics and Computation*, vol. 189, no. 2, pp. 1624–1636, 2007.
- [14] H. P. Benson, "Decomposition branch-and-bound based algorithm for linear programs with additional multiplicative constraints," *Journal of Optimization Theory and Applications*, vol. 126, no. 1, pp. 41–61, 2005.
- [15] H.-S. Ryoo and N. V. Sahinidis, "Global optimization of multiplicative programs," *Journal of Global Optimization*, vol. 26, no. 4, pp. 387–418, 2003.
- [16] S. Schaible and C. Sodini, "Finite algorithm for generalized linear multiplicative programming," *Journal of Optimization Theory and Applications*, vol. 87, no. 2, pp. 441–455, 1995.
- [17] P. Shen, Y. Ma, and Y. Chen, "A robust algorithm for generalized geometric programming," *Journal of Global Optimization*, vol. 41, no. 4, pp. 593–612, 2008.
- [18] S. Qu, K. Zhang, and F. Wang, "A global optimization using linear relaxation for generalized geometric programming," *European Journal of Operational Research*, vol. 190, no. 2, pp. 345–356, 2008.
- [19] Y. Wang, K. Zhang, and Y. Gao, "Global optimization of generalized geometric programming," *Computers & Mathematics with Applications*, vol. 48, no. 10-11, pp. 1505–1516, 2004.
- [20] P. Shen and K. Zhang, "Global optimization of signomial geometric programming using linear relaxation," *Applied Mathematics and Computation*, vol. 150, no. 1, pp. 99–114, 2004.
- [21] T. Matsui, "NP-hardness of linear multiplicative programming and related problems," *Journal of Global Optimization*, vol. 9, no. 2, pp. 113–119, 1996.
- [22] J. E. Falk and S. W. Palocsay, "Optimizing the sum of linear fractional functions," in *Recent Advances in Global Optimization (Princeton, NJ, 1991)*, Princeton Ser. Comput. Sci., pp. 221–258, Princeton University Press, Princeton, NJ, USA, 1992.
- [23] T. Kuno, "A branch-and-bound algorithm for maximizing the sum of several linear ratios," *Journal of Global Optimization*, vol. 22, no. 1-4, pp. 155–174, 2002.
- [24] Y.-J. Wang and K.-C. Zhang, "Global optimization of nonlinear sum of ratios problem," *Applied Mathematics and Computation*, vol. 158, no. 2, pp. 319–330, 2004.
- [25] P.-P. Shen and G.-X. Yuan, "Global optimization for the sum of generalized polynomial fractional functions," *Mathematical Methods of Operations Research*, vol. 65, no. 3, pp. 445–459, 2007.
- [26] H. Jiao, Z. Wang, and Y. Chen, "Global optimization algorithm for sum of generalized polynomial ratios problem," *Applied Mathematical Modelling*, vol. 37, no. 1-2, pp. 187–197, 2013.
- [27] S.-C. Fang, D. Y. Gao, R.-L. Sheu, and W. Xing, "Global optimization for a class of fractional programming problems," *Journal of Global Optimization*, vol. 45, no. 3, pp. 337–353, 2009.
- [28] H. Tuy, F. Al-Khayyal, and P. T. Thach, "Monotonic optimization: branch and cut methods," in *Essays and Surveys in Global Optimization*, C. Audet, P. Hansen, and G. Savard, Eds., vol. 7 of *GERAD 25th Anniv. Ser.*, pp. 39–78, Springer, New York, NY, USA, 2005.
- [29] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, Springer, Berlin, Germany, 2nd edition, 1993.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

