Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2014, Article ID 163263, 12 pages
http://dx.doi.org/10.1155/2014/163263

Research Article

A Global Optimization Algorithm for Signomial Geometric Programming Problem

Xue-Ping Hou, Pei-Ping Shen, and Yong-Qiang Chen

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

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

Received 17 December 2013; Accepted 16 January 2014; Published 30 March 2014

Academic Editor: Yisheng Song

This paper presents a global optimization algorithm for solving the signomial geometric programming (SGP) problem. In the algorithm, by the straight forward algebraic manipulation of terms and by utilizing a transformation of variables, the initial nonconvex programming problem (SGP) is first converted into an equivalent monotonic optimization problem and then is reduced to a sequence of linear programming problems, based on the linearizing technique. To improve the computational efficiency of the algorithm, two range reduction operations are combined in the branch and bound procedure. The proposed algorithm is convergent to the global minimum of the (SGP) by means of the subsequent solutions of a series of relaxation linear programming problems. And finally, the numerical results are reported to vindicate the feasibility and effectiveness of the proposed method.

1. Introduction

The signomial geometric programming (SGP) problem can be formulated as the following nonlinear optimization problem:

$$\text{(SGP)}: \begin{cases} \min & \Phi_0(y) \\ \text{s.t.} & \Phi_m(y) \le 0, \quad m = 1, \dots, M_0, \\ & y \in \Omega_0, \end{cases} \tag{1}$$

where

$$\Phi_m(y) = \sum_{t=1}^{T_m} \delta_{mt} \prod_{i=1}^{n_0} y_i^{\eta_{mti}}, \quad m = 0, 1, \dots, M_0,$$
$$\Omega_0 = \left\{ y \in R_+^{n_0} \mid 0 < y_i^l \le y_i \le y_i^u < \infty,\ i = 1, \dots, n_0 \right\}. \tag{2}$$

T_m are positive integers and δ_{mt} and η_{mti} are all arbitrary real constant coefficients and exponents, respectively. In general, the problem (SGP) corresponds to a nonlinear optimization problem with nonconvex objective function and constraint set. As noted by [1, 2], many nonlinear programming problems may be restated as geometric programming with little additional effort by simple techniques such as change of variables or by straightforward algebraic manipulation of terms. Additionally, (SGP) problem has found a wide range of applications in production planning, location, distribution contexts in risk management problems, various chemical process design and engineering design situations, and so on [3–10]. Hence, it is necessary to present good algorithms for solving (SGP).

The theory of (SGP) was initially developed over three decades ago by Duffin et al. [11–13]. Subsequently, it had been studied by a number of researchers. In general, local optimization approaches for solving (SGP) problem include three kinds of methods as follows. First, successive approximation by posynomials has received the most popularity [14]. Second, Passy and Wilde [15] developed a weaker type of duality to accommodate this class of nonlinear optimization. Third, general nonlinear programming methods [16]. Though local optimization methods for solving SGP problem are ubiquitous, the global optimization algorithm based on the characteristics of (SGP) problem is scarce. When η_{mti} in $\Phi_m(y)$ is positive integer or rational number, some authors in [8, 17–19] developed the corresponding global solution methods for (SGP). In this case that each η_{mti} is real, Maranas et al. [20] proposed a global optimization branch and bound algorithm, by using the exponential variable transformation of (SGP) and the convex relaxation. Shen and Zhang [21] also proposed a global optimization algorithm based on the exponential variable transformation of (SGP) and the linear relaxation. Recently, Shen et al. [22] presented a robust algorithm for (SGP) problem by seeking an essential optimal solution. Wang et al. [23] developed a general algorithm for solving (SGP) problem with nonpositive degree of difficulty. Qu et al. [24] proposed a global optimization algorithm using linear relaxation for (SGP) problem.

In this paper we present a new global optimization algorithm for (SGP) problem by using several reduction operations and by solving a sequence of linear programming problems over partitioned subsets. The proposed method uses a convenient transformation based on the characteristics of (SGP) problem; thus, the original problem (SGP) is equivalently reformulated as a monotonic optimization problem (P), that is, the objective function is increasing and all the constrained functions can be denoted by the difference of two increasing functions in problem (P). A comparison of this method with other methods reviewed above is given below. First, the proposed linear relaxation is based on the monotonic optimization problem (P), which applies more information of the functions of (SGP). And what is more important is that the proposed reduction operations which are adopted in our global optimization algorithm can cut away a large part of the region in which the global optimal solution of (SGP) does not exist. This solution procedure will be more efficient than the methods in [21, 25, 26]. Second, the problem investigated in this paper generalizes those of [8, 17–19]. Furthermore, our method is more convenient in computation than the convex relaxation [19] because the main work is to solve the linear programs and the zeros of strictly monotonic functions of one variable over the interval [0,1), which can be solved very efficiently by the existing methods, for example, by the simplex method and the bisection search method. Third, numerical results and comparison with other methods are conducted to show the potential advantage of the proposed algorithm.

The remainder of this paper is organized as follows. The next section converts the (SGP) problem into a monotonic optimization problem. We discuss the rectangular branching operation, the lower bounding operation, and the reducing operations needed in our algorithm in Section 3. Section 4 incorporates this approach into an algorithm for solving (SGP) and shows the convergence property of the algorithm. In Section 5, we report the results of solving some numerical examples with the algorithm. A summary is presented in the last section.

2. Equivalent Problem

In order to convert (SGP) problem into an equivalent optimization problem (P), for each $m = 1, \ldots, M_0$, $i = 1, \ldots, n_0$, let us denote

$$\begin{aligned} \eta_{mi} &= \min\left\{\eta_{mti} \mid t = 1, \ldots, T_m\right\}, \\ \gamma_{0ti} &= \eta_{0ti}, \\ \gamma_{mti} &= \eta_{mti} - \eta_{mi}. \end{aligned} \tag{3}$$

By multiplying both sides of each constraint inequality of (SGP) with $\prod_{i=1}^{n_0} y_i^{(-\eta_{mi})}$ and by applying the exponent transformation

$$y_i = \exp\left(x_i\right), \quad i = 1, \ldots, n_0, \tag{4}$$

to the formulation (SGP), we can obtain the following equivalent problem:

$$\text{(SGP1)}: \begin{cases} \min & \sum_{t=1}^{T_0} \delta_{0t} \exp\left(\sum_{i=1}^{n_0} \gamma_{0ti} x_i\right) \\ \text{s.t.} & \sum_{t=1}^{T_m} \delta_{mt} \exp\left(\sum_{i=1}^{n_0} \gamma_{mti} x_i\right) \le 0, \quad m = 1, \ldots, M_0, \\ & x \in \Omega = \left\{x \in R^{n_0} \mid \ln y_i^l = x_i^l \le x_i \le x_i^u = \ln y_i^u,\ i = 1, \ldots, n_0\right\}. \end{cases} \tag{5}$$

Next, for convenience, for each $m = 0, 1, \ldots, M_0$, we assume, without loss of generality, that $\delta_{mt} > 0$ for $t = 1, \ldots, J_m$ and $\delta_{mt} < 0$ for $t = J_m + 1, \ldots, T_m$, and some notation is introduced as follows:

$$\begin{aligned} I_t^+ &= \left\{i \mid \gamma_{0ti} > 0,\ i = 1, \ldots, n_0\right\}, \\ I_t^- &= \left\{i \mid \gamma_{0ti} < 0,\ i = 1, \ldots, n_0\right\}. \end{aligned} \tag{6}$$

Thus, by using I_t^+, I_t^-, let us calculate

$$\begin{aligned} L_t &= \sum_{i \in I_t^-} \gamma_{0ti} \ln y_i^u \quad \text{for each } t = 1, \ldots, J_0, \\ U_t &= \sum_{i \in I_t^-} \gamma_{0ti} \ln y_i^l \quad \text{for each } t = 1, \ldots, J_0, \\ l_t &= \sum_{i \in I_t^+} \gamma_{0ti} \ln y_i^l \quad \text{for each } t = J_0 + 1, \ldots, T_0, \\ u_t &= \sum_{i \in I_t^+} \gamma_{0ti} \ln y_i^u \quad \text{for each } t = J_0 + 1, \ldots, T_0. \end{aligned} \tag{7}$$

Then, by introducing some additional variables x_i, $i = n_0 + 1, \ldots, n$, with $n = n_0 + T_0$, we can convert the problem (SGP1)into

$$
(P):\begin{cases}
\min & F_0(x)=\sum_{t=1}^{J_0}\delta_{0t}\exp\left(\sum_{i\in I_t^+}\gamma_{0ti}x_i+x_{n_0+t}\right)+\sum_{t=J_0+1}^{T_0}\delta_{0t}\exp\left(\sum_{i\in I_t^-}\gamma_{0ti}x_i-x_{n_0+t}\right)\\
\text{s.t.} & \sum_{t=1}^{J_m}\delta_{mt}\exp\left(\sum_{i=1}^{n_0}\gamma_{mti}x_i\right)+\sum_{t=J_m+1}^{T_m}\delta_{mt}\exp\left(\sum_{i=1}^{n_0}\gamma_{mti}x_i\right)\le 0,\\
& \qquad m=1,\ldots,M_0,\\
& x_{n_0+t}-\sum_{i\in I_t^-}\gamma_{0ti}x_i\ge 0,\quad t=1,\ldots,J_0,\\
& x_{n_0+t}+\sum_{i\in I_t^+}\gamma_{0ti}x_i\ge 0,\quad t=J_0+1,\ldots,T_0,\\
& x\in X^0,
\end{cases}
\tag{8}
$$

where

$$
\begin{aligned}
X^0&=\left\{x\in R^n\mid x_i^l\le x_i\le x_i^u,\ i=1,\ldots,n\right\}\\
&=\left\{x\in R^n\,\middle|\,\begin{array}{ll} x_i^l\le x_i\le x_i^u, & i=1,\ldots,n_0,\\ L_{i-n_0}\le x_i\le U_{i-n_0}, & i=n_0+1,\ldots,n_0+J_0,\\ -u_{i-n_0}\le x_i\le -l_{i-n_0}, & i=n_0+J_0+1,\ldots,n,\end{array}\right\}.
\end{aligned}
\tag{9}
$$

Additionally, for the sake of simplicity, let $M=M_0+T_0$; the problem (P) can be rewritten as the following form:

$$
\begin{aligned}
(P):\min\{&F_0(x)\mid F_m^+(x)-F_m^-(x)\le 0,\\
&m=1,\ldots,M,\ x\in X^0\},
\end{aligned}
\tag{10}
$$

where

$$
F_m^+(x)=\begin{cases}\sum_{t=1}^{J_m}\delta_{mt}\exp\left(\sum_{i=1}^{n_0}\gamma_{mti}x_i\right), & m=1,\ldots,M_0,\\ 0, & m=M_0+1,\ldots,M,\end{cases}
\tag{11}
$$

$$
F_m^-(x)=\begin{cases}-\sum_{t=J_m+1}^{T_m}\delta_{mt}\exp\left(\sum_{i=1}^{n_0}\gamma_{mti}x_i\right), & m=1,\ldots,M_0,\\
x_{m+n_0-M_0}-\sum_{i\in I_{m-M_0}^-}\gamma_{0(m-M_0)i}x_i, & m=M_0+1,\ldots,M_0+J_0,\\
x_{m+n_0-M_0}+\sum_{i\in I_{m-M_0}^+}\gamma_{0(m-M_0)i}x_i, & m=M_0+J_0+1,\ldots,M.\end{cases}
\tag{12}
$$

Note that each function $F_0(x), F_m^+(x), F_m^-(x)$ of problem (P) is increasing (i.e., a function $f: R^n\to R$ is said to be increasing if $f(x)\le f(y)$ for all $x,y\in R^n$ satisfying $x_i\le y_i$, $i=1,\ldots,n$). Thus problem (P) is a monotonic optimization problem, and the key equivalent result for problems (SGP) and (P) is given by Theorem 1.

Theorem 1. *$y^*\in R^{n_0}$ is a global optimal solution for problem (SGP) if and only if $x^*\in R^n$ is a global optimal solution for problem (P), where*

$$
x_i^*=\begin{cases}\ln y_i^*, & i=1,\ldots,n_0,\\ \sum_{i\in I_t^-}\gamma_{0ti}\ln y_i^*, & i=n_0+1,\ldots,n_0+J_0,\\ -\sum_{i\in I_t^+}\gamma_{0ti}\ln y_i^*, & i=n_0+J_0+1,\ldots,n.\end{cases}
\tag{13}
$$

Proof. The proof of this theorem follows easily from the definitions of problems (SGP) and (P); therefore, it is omitted here. □

From Theorem 1, notice that, in order to solve problem (SGP), we may solve problem (P) instead. In addition, it is easy to see that the global optimal values of problems (SGP) and (P) are equal. Based on the above discussion, here, from now on we assume that the original problem (SGP) has been converted into the problem (P); then a general approach will be considered for solving problem (P).

3. Key Algorithm Processes

To globally solve the problem (P), a branch-reduce-bound (BRB) algorithm will be proposed. This algorithm proceeds according to the standard branch and bound scheme with three key processes: branching, reducing, and bounding.

The branching process consists in a successive rectangular partition of the initial box $X^0=[x^l,x^u]$ following in an exhaustive subdivision rule, that is, such that any infinite nested sequence of partition sets generated through the algorithm shrinks to a singleton. A commonly used exhaustive subdivision rule is the standard bisection.

The reducing process consists in applying reduction operations to reduce the size of the current partition set $X=$

$[a,b] \subset X^0 = [x^l, x^u]$. The process aims at tightening the box containing the feasible portion currently still of interest.

The bounding process consists in using the linearization method to give a better lower bound.

Next, we begin to establish the approaches processes.

3.1. Lower Bound. At a given stage of the BRB algorithm for (P), let $X = [a,b] \subset X^0$ be a rectangle during the partitioning procedure and still of interest; we intend to compute a lower bound $\mathrm{LB}(X)$ of the optimal value of (P) over X. Restrict the problem (P) to X:

$$P(X) : \min \left\{F_0(x) \mid F_m(x) \le 0, m = 1, \ldots, M, x \in X\right\}. \tag{14}$$

Denote the optimal objective function value of problem $P(X)$ by $V[P(X)]$.

Since $F_0(x)$ is increasing, an obvious bound is $\mathrm{LB}(X) = F_0(a)$; although very simple, this bound suffices to ensure convergence of the algorithm. However, the following procedure may give a better bound.

Our main method for computing a lower bound of $V[P(X)]$ over X is to solve the relaxation linear programming of $P(X)$. The linear relaxation of the problem $P(X)$ can be realized by underestimating every function $F_0(x)$ and $F_m^+(x)$ and by overestimating every function $F_m^-(x)$, for each $m = 1, \ldots, M_0$. All the details for generating the linear relaxation will be given in the following.

Denote

$$\begin{aligned}
X_{0t} &= \begin{cases} \sum_{i \in I_t^+} \gamma_{0ti} x_i + x_{n_0+t}, & t = 1, \ldots, J_0, \\ \sum_{i \in I_t^-} \gamma_{0ti} x_i - x_{n_0+t}, & t = J_0+1, \ldots, T_0, \end{cases} \\
X_{0t}^l &= \begin{cases} \sum_{i \in I_t^+} \gamma_{0ti} a_i + a_{n_0+t}, & t = 1, \ldots, J_0, \\ \sum_{i \in I_t^-} \gamma_{0ti} b_i - b_{n_0+t}, & t = J_0+1, \ldots, T_0, \end{cases} \\
X_{0t}^u &= \begin{cases} \sum_{i \in I_t^+} \gamma_{0ti} b_i + b_{n_0+t}, & t = 1, \ldots, J_0, \\ \sum_{i \in I_t^-} \gamma_{0ti} a_i - a_{n_0+t}, & t = J_0+1, \ldots, T_0, \end{cases} \\
X_{mt} &= \sum_{i=1}^{n_0} \gamma_{mti} x_i, \quad t = 1, \ldots, T_m, \\
X_{mt}^l &= \sum_{i=1}^{n_0} \gamma_{mti} a_i, \quad t = 1, \ldots, T_m, \\
X_{mt}^u &= \sum_{i=1}^{n_0} \gamma_{mti} b_i, \quad t = 1, \ldots, T_m,
\end{aligned} \tag{15}$$

where $m = 1, \ldots, M_0$. In addition, let

$$\begin{aligned}
A_{mt} &= \frac{\exp\left(X_{mt}^u\right) - \exp\left(X_{mt}^l\right)}{X_{mt}^u - X_{mt}^l}, \\
\theta_{mt}(x) &= \exp\left(X_{mt}\right), \\
\overline{\theta}_{mt}(x) &= A_{mt}\left(X_{mt} - X_{mt}^l\right) + \exp\left(X_{mt}^l\right), \\
\underline{\theta}_{mt}(x) &= A_{mt}\left(X_{mt} - \ln A_{mt} + 1\right),
\end{aligned} \tag{16}$$

where $m = 0, 1, \ldots, M_0$, $t = 1, \ldots, T_m$.

Theorem 2. *Consider the functions $\theta_{mt}(x)$, $\underline{\theta}_{mt}(x)$, and $\overline{\theta}_{mt}(x)$, for any $x \in X$, where $m = 0, \ldots, M_0$ and $t = 1, \ldots, T_m$. Then the following two statements are valid.*

(i) *The function $\overline{\theta}_{mt}(x)$ is the concave envelope of the function $\theta_{mt}(x)$ over X, and the function $\underline{\theta}_{mt}(x)$ is a supporting hyperplane of $\theta_{mt}(x)$, which is parallel with $\overline{\theta}_{mt}(x)$. Moreover, the functions $\theta_{mt}(x)$, $\underline{\theta}_{mt}(x)$, and $\overline{\theta}_{mt}(x)$ satisfy*

$$\underline{\theta}_{mt}(x) \le \theta_{mt}(x) \le \overline{\theta}_{mt}(x), \quad \forall x \in X. \tag{17}$$

(ii) *The differences $\Delta_{mt}^1(x) = \overline{\theta}_{mt}(x) - \theta_{mt}(x)$ and $\Delta_{mt}^2(x) = \theta_{mt}(x) - \underline{\theta}_{mt}(x)$ satisfy $\max_{x \in X} \Delta_{mt}^1(x) = \max_{x \in X} \Delta_{mt}^2(x) = \exp(X_{mt}^l)(1 - Z_{mt} + Z_{mt} \ln Z_{mt}) \to 0$ as $\omega_{mt} \to 0$, where*

$$\omega_{mt} = X_{mt}^u - X_{mt}^l, \qquad Z_{mt} = \frac{\exp\left(\omega_{mt}\right) - 1}{\omega_{mt}}. \tag{18}$$

Proof. The proof is similar to Theorem 1 in [21]; therefore, it is omitted here. □

Remark 3. From Theorem 2, we can follow that the functions $\underline{\theta}_{mt}(x)$ and $\overline{\theta}_{mt}(x)$ enough approximate the function $\theta_{mt}(x)$ as $\omega_{mt} \to 0$, respectively.

From Theorem 2, it is obvious that for all $x \in X$ we have

$$\begin{aligned}
F_0(x) &\ge \mathrm{LF}_0(x) = \sum_{t=1}^{J_0} \delta_{0t} \underline{\theta}_{0t}(x) + \sum_{t=J_0+1}^{T_0} \delta_{0t} \overline{\theta}_{0t}(x), \\
F_m^+(x) &\ge \mathrm{LF}_m^+(x) = \sum_{t=1}^{J_m} \delta_{mt} \underline{\theta}_{mt}(x), \\
F_m^-(x) &\le \mathrm{UF}_m^-(x) = -\sum_{t=J_m+1}^{T_m} \delta_{mt} \overline{\theta}_{mt}(x),
\end{aligned} \tag{19}$$

where $m = 1, \ldots, M_0$.

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

$$\mathrm{RLP}(X)$$

$$: \begin{cases} \min & \mathrm{LF}_0(x) \\ \text{s.t.} & \mathrm{LF}_m^+(x) - \mathrm{UF}_m^-(x) \le 0, \quad m = 1, \ldots, M_0, \\ & F_m^+(x) - F_m^-(x) \le 0, \quad m = M_0 + 1, \ldots, M, \\ & x \in X. \end{cases} \tag{20}$$

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

$$V[\mathrm{RLP}(X)] \le V[P(X)], \tag{21}$$

and thus, from (21), the optimal value $V[\mathrm{RLP}(X)]$ of RLP(X) provides a valid lower bound for the optimal value $V[P(X)]$ of $P(X)$ over X.

Based on the above discussion, for any rectangle X, in order to obtain a lower bound LB(X) of the optimal value $V[P(X)]$ of the problem $P(X)$, we may compute LB(X) such that

$$\mathrm{LB}(X) = \max\{V[\mathrm{RLP}(X)], F_0(a)\}. \tag{22}$$

Clearly, LB(X) defined in (22) satisfies

$$F_0(a) \le \mathrm{LB}(X) \le V[P(X)] \tag{23}$$

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

3.2. Reduction Operations. Clearly, the smaller the rectangle X is, the tighter the lower bound LB(X) of $P(X)$ will be, and therefore the closer the feasible solution of (P) will be to the optimal solution of (P). To show this, the next results give two reduction operations (i.e., reduction rules A and B) to reduce the size of this partitioned rectangle without losing any feasible solution currently still of interest.

3.2.1. Reduction Rule A. Rule A is based on the monotonic structure of the problem (P). At a given stage of the BRB algorithm for (P), for a rectangle $X = [a, b]$ generated during the partitioning procedure and still of interest, let UB be the object function value of the best so far feasible solution to problem (P). Given an $\varepsilon > 0$, we want to find a feasible solution $x \in X$ of (P) such that $F_0(x) \le \mathrm{UB} - \varepsilon$ or else establish that no such x exists. So the search for such x can then be restricted to the set $H \cap [a, b]$, where

$$H := \{x \mid F_0(x) \le \mathrm{UB} - \varepsilon,\ F_m(x) \le 0,\ m = 1, \ldots, M\}. \tag{24}$$

The reduction rule aims at replacing the rectangle $[a, b]$ with a smaller rectangle $[a', b'] \subset [a, b]$ without losing any point $x \in H \cap [a, b]$, that is, such that $H \cap [a', b'] = H \cap [a, b]$. The rectangle $[a', b']$ satisfying this condition is denoted by $\mathrm{red}_\nu[a, b]$ with

$$\nu = \mathrm{UB} - \varepsilon. \tag{25}$$

To illustrate how $\mathrm{red}_\nu[a, b] = [a', b']$ is deduced by this rule, we first define the following functions.

Definition 4. Given two boxes $[a, b]$ and $[a', b']$ with $[a', b'] \subseteq [a, b]$, for $i = 1, \ldots, n$, $m = 1, \ldots, M$, the functions $\varphi_m^i(\alpha)$, $\psi_m^i(\alpha)$, and $\psi_0^i(\alpha) : [0, 1] \to R$ are defined by

$$\begin{aligned} \varphi_m^i(\alpha) &= F_m^-\left(b - \alpha(b_i - a_i)e^i\right) - F_m^+(a), \\ \psi_m^i(\alpha) &= F_m^+\left(a' + \alpha\left(b_i - a_i'\right)e^i\right) - F_m^-(b), \\ \psi_0^i(\alpha) &= F_0\left(a' + \alpha\left(b_i - a_i'\right)e^i\right) - \nu, \end{aligned} \tag{26}$$

where e^i denotes the ith unit vector of R^n, that is, a vector such that $e_i^i = 1$, $e_j^i = 0, \forall j \neq i$, and the functions $F_0(x)$, $F_m^+(x)$, and $F_m^-(x)$ are given in problem (P), respectively.

Clearly, the functions $\varphi_m^i(\alpha)$, $\psi_m^i(\alpha)$, and $\psi_0^i(\alpha)$ are either constant or strictly monotonic over the interval [0,1) from the properties of $F_m^-(x)$, $F_m^+(x)$, and $F_0(x)$. By using these functions, $\mathrm{red}_\nu[a, b]$ can be given as follows.

Theorem 5. (i) *If $F_0(a) > \nu$ or $F_m^+(a) - F_m^-(b) > 0$ for some $m = 1, \ldots, M$, then $\mathrm{red}_\nu[a, b] = [a', b'] = \emptyset$.*

(ii) *If $F_0(a) \le \nu$ and $F_m^+(a) - F_m^-(b) \le 0$ for each $m = 1, \ldots, M$, then $\mathrm{red}_\nu[a, b] = [a', b']$, where*

$$\begin{aligned} a' &= b - \sum_{i=1}^{n} \min_{m=1,\ldots,M} \left\{\alpha_m^i\right\}(b_i - a_i)e^i, \\ b' &= a' + \sum_{i=1}^{n} \min_{m=0,1,\ldots,M} \left\{\beta_m^i\right\}\left(b_i - a_i'\right)e^i \end{aligned} \tag{27}$$

are given by

$$\begin{aligned} \alpha_m^i &= \begin{cases} 1, & \text{if } \varphi_m^i(1) \ge 0 \\ \alpha \text{ with } \varphi_m^i(\alpha) = 0, & \text{otherwise}, \end{cases} \\ \beta_m^i &= \begin{cases} 1, & \text{if } \psi_m^i(1) \le 0 \\ \alpha \text{ with } \psi_m^i(\alpha) = 0, & \text{otherwise}. \end{cases} \end{aligned} \tag{28}$$

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

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

$$F_0(x) \le \nu, \qquad F_m^+(x) - F_m^-(x) \le 0, \quad m = 1, \ldots, M, \tag{29}$$

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

$$\begin{aligned} \alpha_{m'}^i &= \min\left\{\alpha_m^i \mid m = 1, \ldots, M\right\}, \\ \beta_{m''}^i &= \min\left\{\beta_m^i \mid m = 0, 1, \ldots, M\right\}. \end{aligned} \tag{30}$$

Firstly, we will show that $x \geq a'$. If $x \ngeq a'$, then there exists index i such that

$$\begin{aligned} &x_i < a_i' = b_i - \alpha_{m'}^i (b_i - a_i), \\ &\text{i.e., } x_i = b_i - \alpha (b_i - a_i) \quad \text{with } \alpha_{m'}^i < \alpha \leq 1. \end{aligned} \tag{31}$$

We consider the following two cases.

Case 1. If $\alpha_{m'}^i = 1$, then from (31) 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 $\varphi_{m'}^i(\alpha)$ must be strictly decreasing in single variable α over the interval [0,1). If the function $\varphi_{m'}^i(\alpha)$ is not strictly decreasing in single variable α, we get $\varphi_{m'}^i(\alpha)$ must be a constant over the interval [0,1). In this case, we have

$$\varphi_{m'}^i (1) = \varphi_{m'}^i (0) = F_{m'}^- (b) - F_{m'}^+ (a) \geq 0. \tag{32}$$

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 $\varphi_{m'}^i(\alpha)$ is strictly decreasing, it follows from (31) and the definition of $\alpha_{m'}^i$ that

$$\begin{aligned} &F_{m'}^- \left(b - (b_i - x_i) e^i\right) - F_{m'}^+ (a) \\ &\quad = F_{m'}^- \left(b - \alpha (b_i - a_i) e^i\right) - F_{m'}^+ (a) \\ &\quad = \varphi_{m'}^i (\alpha) < \varphi_{m'}^i \left(\alpha_{m'}^i\right) = 0; \end{aligned} \tag{33}$$

hence,

$$F_{m'}^- \left(b - (b_i - x_i) e^i\right) < F_{m'}^+ (a). \tag{34}$$

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

$$F_{m'}^- (x) \leq F_{m'}^- \left(b - (b_i - x_i) e^i\right) < F_{m'}^+ (a), \tag{35}$$

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

$$x \leq b', \quad \text{i.e., } x \in \left[a', b'\right]. \tag{36}$$

Supposed that $x \nleq b'$, then there exists some i such that

$$x_i > b_i' = a_i' + \beta_{m''}^i \left(b_i - a_i'\right); \tag{37}$$

that is, there exists α such that

$$x_i = a_i' + \alpha \left(b_i - a_i'\right), \quad \beta_{m''}^i < \alpha \leq 1. \tag{38}$$

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

Case 1. If $\beta_{m''}^i = 1$, then from (38) 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 $\psi_{m''}^i(\alpha)$ is strictly increasing in single variable α. If the function $\psi_{m''}^i(\alpha)$ is not strictly increasing in single variable α, we get $\psi_{m''}^i(\alpha)$ must be a constant over the interval [0,1). In this case, we have

$$\psi_0^i (1) = \psi_0^i (0) = F_0 \left(a'\right) - \nu \leq 0, \tag{39}$$

or

$$\psi_{m''}^i (1) = \psi_{m''}^i (0) = F_{m''}^+ \left(a'\right) - F_{m''}^- (b) \leq 0. \tag{40}$$

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 $\psi_{m''}^i(\alpha)$ is strictly increasing, from (31) and the definition of $\beta_{m''}^i$, it implies that

$$F_0 \left(a' + \alpha \left(b_i - a_i'\right) e^i\right) - \nu = \psi_0^i (\alpha) > \psi_0^i \left(\beta_0^i\right) = 0, \tag{41}$$

or

$$\begin{aligned} &F_{m''}^+ \left(a' + \alpha \left(b_i - a_i'\right) e^i\right) - F_{m''}^- (b) \\ &\quad = \psi_{m''}^i (\alpha) > \psi_{m''}^i \left(\beta_{m''}^i\right) = 0. \end{aligned} \tag{42}$$

Assume that (41) holds; we can derive from (38) that

$$F_0 \left(a' + \left(x_i - a_i'\right) e^i\right) = F_0 \left(a' + \alpha \left(b_i - a_i'\right) e^i\right) > \nu. \tag{43}$$

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

$$F_0 (x) \geq F_0 \left(a' + \left(x_i - a_i'\right) e^i\right) > \nu, \tag{44}$$

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

If (42) holds, we obtain from (38) that

$$\begin{aligned} &F_{m''}^+ \left(a' + \left(x_i - a_i'\right) e^i\right) \\ &\quad = F_{m''}^+ \left(a' + \alpha \left(b_i - a_i'\right) e^i\right) > F_{m''}^- (b); \end{aligned} \tag{45}$$

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

$$F_{m''}^+ (x) \geq F_{m''}^+ \left(a' + \left(x_i - a_i'\right) e^i\right) > F_{m''}^- (b). \tag{46}$$

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. □

Remark 6. Clearly, for any $i = 1, \ldots, n$, $\alpha_m^i (m = 1, \ldots, M)$ and β_m^i $(m = 0, 1, \ldots, M)$ defined in Theorem 5 must exist and be unique, since the functions $F_0(x)$, $F_m^+(x)$, and $F_m^-(x)$ are all continuous and increasing.

3.2.2. Reduction Rule B. For any $x \in X = (X_i)_{n\times 1}$ with $X_i = [a_i, b_i]$ $(i = 1, \ldots, n)$, without loss of generality, we assume the above relaxation linear problem RLP(X) can be rewritten as

$$\text{RLP}(X): \begin{cases} \min & \sum_{i=1}^{n} \lambda_{0i} x_i + t_0 \\ \text{s.t.} & \sum_{i=1}^{n} \lambda_{ji} x_i + t_j \leq 0, \quad j = 1, \ldots, M, \\ & x \in X \subseteq X^0. \end{cases} \tag{47}$$

Let

$$\mathrm{RL}_j = \sum_{i=1}^{n} \min\left\{\lambda_{ji}a_i, \lambda_{ji}b_i\right\} + t_j, \quad j = 0, 1, \ldots, M, \tag{48}$$

$$\rho_i = \frac{\mathrm{UB} - \mathrm{RL}_0 + \min\left\{\lambda_{0i}a_i, \lambda_{0i}b_i\right\}}{\lambda_{0i}} \quad \text{with } \lambda_{0i} \neq 0, \tag{49}$$

$$\tau_{ji} = \frac{-\mathrm{RL}_j + \min\left\{\lambda_{ji}a_i, \lambda_{ji}b_i\right\}}{\lambda_{ji}} \quad \text{with } \lambda_{ji} \neq 0, \tag{50}$$

where $j = 1, \ldots, M$, $i = 1, \ldots, n$.

Theorem 7. *For any rectangle $X = (X_i)_{n\times 1} \subseteq X^0$, if $RL_0 > UB$, then there exists no optimal solution of $RLP(X^0)$ over X; otherwise, consider the following two cases: if there exists some $h \in \{1, \ldots, n\}$ satisfying $\lambda_{0h} > 0$ and $\rho_h < b_h$, then there is no optimal solution of $RLP(X^0)$ over X_a; conversely, if $\lambda_{0h} < 0$ and $\rho_h > a_h$ for some $h \in \{1, \ldots, n\}$, then there does not exist optimal solution of $RLP(X^0)$ over X_b, where*

$$\begin{gathered} X_a = (X_{ai})_{n\times 1} \subseteq X^0 \\ \text{with } X_{ai} = \begin{cases} X_i, & \text{if } i \neq h, \\ (\rho_h, b_h] \bigcap X_h, & \text{if } i = h, \end{cases} \\ X_b = (X_{bi})_{n\times 1} \subseteq X^0 \\ \text{with } X_{bi} = \begin{cases} X_i, & \text{if } i \neq h, \\ [a_h, \rho_h) \bigcap X_h, & \text{if } i = h. \end{cases} \end{gathered} \tag{51}$$

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

$$\begin{gathered} X_c = (X_{ci})_{n\times 1} \subseteq X^0 \\ \text{with } X_{ci} = \begin{cases} X_i, & \text{if } i \neq h, \\ (\tau_{jh}, b_h] \bigcap X_h, & \text{if } i = h, \end{cases} \\ X_d = (X_{di})_{n\times 1} \subseteq X^0 \\ \text{with } X_{di} = \begin{cases} X_i, & \text{if } i \neq h, \\ [a_h, \tau_{jh}) \bigcap X_h, & \text{if } i = h. \end{cases} \end{gathered} \tag{52}$$

Proof. The proof of the Theorems 7 and 8 is similar to Theorems 2 and 3 in [27], respectively; therefore, it is omitted here. □

By Theorems 7 and 8, we can give a new reduction rule B to reject some regions in which the globally optimal solution of $\mathrm{RLP}(X^0)$ does not exist. The computation procedure of this rule is summarized as follows.

Step 1. Compute RL_0 in (48). If $\mathrm{RL}_0 > \mathrm{UB}$, let $X = \emptyset$; otherwise, compute ρ_i $(i = 1, \ldots, n)$ in (49). If $\lambda_{0h} > 0$ and $\rho_h < b_h$ for some $h \in \{1, \ldots, n\}$, then let $b_h = \rho_h$ and $X = (X_i)_{n\times 1}$ with $X_i = [a_i, b_i]$ $(i = 1, \ldots, n)$. If $\lambda_{0h} < 0$ and $\rho_h > a_h$ for some $h \in \{1, \ldots, n\}$, then let $a_h = \rho_h$ and $X = (X_i)_{n\times 1}$ with $X_i = [a_i, b_i]$ $(i = 1, \ldots, n)$.

Step 2. For any $j = 1, \ldots, M$, compute RL_j in (48). If $\mathrm{RL}_j > 0$ for some $j \in \{1, \ldots, M\}$, then let $X = \emptyset$; otherwise, compute τ_{ji} in (50) $(j = 1, \ldots, M,\ i = 1, \ldots, n)$. If $\lambda_{jh} > 0$ and $\tau_{jh} < b_h$ for some $j \in \{1, \ldots, M\}$ and $h \in \{1, \ldots, n\}$, then let $b_h = \tau_{jh}$ and $X = (X_i)_{n\times 1}$ with $X_i = [a_i, b_i]$ $(i = 1, \ldots, n)$. If $\lambda_{jh} < 0$ and $\tau_{jh} > a_h$ for some $j \in \{1, \ldots, M\}$ and $h \in \{1, \ldots, n\}$, then let $a_h = \tau_{jh}$ and $X = (X_i)_{n\times 1}$ with $X_i = [a_i, b_i]$ $(i = 1, \ldots, n)$.

Rule B provides a possibility to cut away all or a large part of the rectangle X which is currently investigated by the algorithm procedure.

4. Algorithm and Its Convergence

In this section, a branch-reduce-bound (BRB) algorithm is developed to solve the problem (P) 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 $\mathcal{T}_k$, that is, each associated with a rectangle $X \subseteq X^0$, for all $X \in \mathcal{T}_k$. For each such node $X = [a, b]$, we will compute a lower bound $\mathrm{LB}(X)$ of the optimal objective function value of (P) via the optimal value of the $\mathrm{RLP}(X)$ and $F_0(a)$, so the lower bound of the optimal value of (P) at stage k is given by $\min\{\mathrm{LB}(X), \forall X \in \mathcal{T}_k\}$. We now select an active node to subdivide its associated rectangle into two subrectangles according to the standard branch rule for each new node, reducing 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.

Algorithm 1. Consider the following steps.

Step 0 (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 H and let $\mathrm{UB}_0 = \min\{F_0(x) \mid x \in H\}$; otherwise, let $H = \emptyset$ and $\mathrm{UB}_0 = +\infty$. Set $k = 0$.

Step 1 (Reduction). (i) Delete every box $[a, b] \in \mathcal{P}_k$ such that $F_0(a) > \mathrm{UB}_k - \varepsilon$ or $F_m^+(a) - F_m^-(b) > 0$ for some $m \in \{1, \ldots, M\}$, and denote the remaining still as $\mathcal{P}_k$. If $\mathcal{P}_k \neq \emptyset$, apply the reduction rule A described in Theorem 5 in Section 3.2 to each box $[a,b] \in \mathscr{P}_k$. Let $\mathscr{P}'_k = \{\text{red}_\nu[a,b] \mid [a,b] \in \mathscr{P}_k\}$ with $\nu = \text{UB}_k - \varepsilon$.

(ii) If $\mathscr{P}'_k \neq \emptyset$, for each box $[a,b] \in \mathscr{P}'_k$ that is currently investigated, we use the reduction rule B in Section 3.2 to cut away X and denote the left still as $\mathscr{P}''_k$.

Step 2 (Bounding). If $\mathscr{P}''_k \neq \emptyset$, begin to do for each $[a,b] \in \mathscr{P}''_k$ the following.

(i) Solve the problem RLP(X) to obtain the optimal solution $x(X)$ and the optimal value $V[\text{RLP}(X)]$. Let $\text{LB}(X) = \max\{V[\text{RLP}(X)], F_0(a)\}$.

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

(iii) If $F_0(b) > \text{UB} - \varepsilon$, compute a point $\hat{x} = a + \theta(b-a)$ such that $F_0(a + \theta(b-a)) = \text{UB} - \varepsilon$; otherwise, let $\hat{x} = b$.

(iv) If $H \neq \emptyset$, define the new upper bound $\text{UB}_k = \min\{F_0(x) \mid x \in H\}$, and the best known feasible point is denoted by $x^* = \text{argmin}\{F_0(x) \mid x \in H\}$. Set $\mathscr{T}_k = (\mathscr{T}_k \setminus X^k) \cup \mathscr{P}''_k$.

Step 3 (Convergence Checking). Set $\mathscr{T}_{k+1} = \mathscr{T}_k \backslash \{X \mid \text{LB}(X) > \text{UB}_k - \varepsilon, X \in \mathscr{T}_k\}$.

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

Step 4 (Branching). Divide X^{k+1} into two new subrectangles using the standard branch rule and let $\mathscr{P}_{k+1}$ be the collection of these two subrectangles. Set $k = k+1$ and return to Step 1.

Convergence Analysis. In this subsection, we give the convergence of the proposed algorithm. Assume that the number of globally optimal solutions of (SGP) is finite. Then the above proposed algorithm either terminates finitely at a globally optimal solution or generates an infinite sequence of iteration nodes. If the algorithm terminates at some iteration k, then obviously the point x^* is a globally optimal solution and UB is the optimal value of problem (P). If the algorithm is infinite, its convergence is discussed as follows.

Theorem 9. *Assume that the above algorithm is infinite; then it generates an infinite sequence of iterations such that along any infinite branch-and-bound tree any accumulation point of the sequence $\{\text{LB}_k\}$ will be the global minimum of problem (P).*

Proof. Since 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 $l = 1,2,\ldots$. In this case, for every iteration $k = 0,1,2,\ldots$, from [28, 29] there is at least an infinite subsequence $\{\text{LB}_{k_l}\}$ of $\{\text{LB}_k\}$ such that

$$\text{LB}_{k_l} \leqslant \min_{x\in X} F_0(x), \qquad X^{k_l} \in \arg\min_{X\in\mathscr{T}_{k_l}} \text{LB}(X),$$
$$x^{k_l} = x\left(X^{k_l}\right) \in X^{k_l} \subseteq X^0, \tag{53}$$

where X denotes the feasible region of problem (P). We see from [28–30] that $\{\text{LB}_{k_l}\}$ is a nondecreasing sequence bounded above by $\min_{x\in X} F_0(x)$, which guarantees the existence of the limit $\lim_{l\to\infty}\text{LB}_{k_l} := \text{LB}$ and $\text{LB} \leq \min_{x\in X} F_0(x)$. Since $\{x^{k_l}\}$ is an infinite sequence on a compact set, it follows that there exists a convergent subsequence $\{x^q\}$ of $\{x^{k_l}\}$ satisfying $\lim_{q\to\infty} x^q = \hat{x}, x^q \in X^q$ and $\text{LB}_q = \text{LB}(X^q) = \text{LF}_0(x^q)$, where $\{X^q\}$ is a subsequence of $\{X^{k_l}\}$. The linear functions LF_j $(j = 0,1,\ldots,M)$ used in the problem RLP(X) are strongly consistent on X^0. Thus, $\lim_{q\to\infty}\text{LB}_q = \text{LB} = F_0(\hat{x})$. All that remains is to show that $\hat{x} \in X$. Since X^0 is a closed set, it follows that $\hat{x} \in X^0$. Suppose that $\hat{x} \notin X$. Then there exists some F_j, $j \in \{1,\ldots,M\}$, such that $F_j(\hat{x}) = \delta > 0$. Since $\text{LF}_j(x)$ is continuous, the sequence $\{\text{LF}_j(x^q)\}$ converges to $F_j(\hat{x})$ as $q \to \infty$. By definition of convergence, $\exists\, q_\delta$ 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 RLP(X) is infeasible. This contradicts the assumption of $x^q = x(X^q)$. Therefore, $\hat{x} \in X$; that is, $\text{LB} = F_0(\hat{x}) = \min_{x\in X} F_0(x)$, and the proof is complete. □

5. Numerical Results

To verify the performance of the proposed algorithm, we will give some computational results through ten test problems. The algorithm is coded in Compaq Visual Fortran. The simplex method is applied to solve the relaxation linear programming problems. All test problems are implemented in an Athlon(tm) CPU 2.31 GHz with 960 MB RAM microcomputer.

Example 1 (see [22, 31, 32]). Consider

$$\begin{aligned} \min \quad & y_3^{0.8} y_4^{1.2} \\ \text{s.t.} \quad & y_1 y_4^{-1} + y_2^{-1} y_4^{-1} \leq 1, \\ & -y_1^{-2} y_3^{-1} - y_2 y_3^{-1} \leq 1, \\ & 0.1 \leq y_1 \leq 1, \quad 5 \leq y_2 \leq 10, \\ & 8 \leq y_3 \leq 15, \quad 0.01 \leq y_4 \leq 1. \end{aligned} \tag{54}$$

Example 2 (see [22, 32]). Consider

$$\begin{aligned} \min \quad & y_0 \\ \text{s.t.} \quad & 3.7 y_0^{-1} y_1^{0.85} + 1.985 y_0^{-1} y_1 + 700.3 y_0^{-1} y_2^{-0.75} \leq 1, \\ & 0.7673 y_2^{0.05} - 0.05 y_1 \leq 1, \quad 5 \leq y_0 \leq 15, \\ & 0.1 \leq y_1 \leq 5, \quad 380 \leq y_2 \leq 450. \end{aligned} \tag{55}$$

Example 3 (see [21, 22, 31, 33]). Consider

$$\begin{aligned} \min \quad & 0.5 y_1 y_2^{-1} - y_1 - 5 y_2^{-1} \\ \text{s.t.} \quad & 0.01 y_2 y_3^{-1} + 0.01 y_2 + 0.0005 y_1 y_3 \leq 1, \\ & 70 \leq y_1 \leq 150, \quad 1 \leq y_2 \leq 30, \\ & 0.5 \leq y_3 \leq 21. \end{aligned} \tag{56}$$

Example 4 (see [21, 22, 33]). Consider

$$\begin{aligned}
\min \quad & 168 y_1 y_2 + 3651.2 y_1 y_2 y_3^{-1} + 4 \times 10^4 y_4^{-1} \\
\text{s.t.} \quad & 1.0425 y_1 y_2^{-1} \leq 1, \\
& 3.5 \times 10^{-4} y_1 y_2 \leq 1, \\
& 1.25 y_1^{-1} y_4 + 41.63 y_1^{-1} \leq 1, \\
& 40 \leq y_1 \leq 44, \quad 40 \leq y_2 \leq 45, \\
& 60 \leq y_3 \leq 70, \quad 0.1 \leq y_4 \leq 1.4.
\end{aligned} \tag{57}$$

Example 5 (see [21, 22, 33]). Consider

$$\begin{aligned}
\min \quad & 5.3578 y_3^2 + 0.8357 y_1 y_5 + 37.2392 y_1 \\
\text{s.t.} \quad & 0.00002584 y_3 y_5 - 0.00006663 y_2 y_5 \\
& \quad - 0.0000734 y_1 y_4 \leq 1, \\
& 0.000853007 y_2 y_5 + 0.00009395 y_1 y_4 \\
& \quad - 0.00033085 y_3 y_5 \leq 1, \\
& 1330.3294 y_2^{-1} y_5^{-1} - 0.42 y_1 y_5^{-1} \\
& \quad - 0.30586 y_2^{-1} y_3^2 y_5^{-1} \leq 1, \\
& 0.00024186 y_2 y_5 + 0.00010159 y_1 y_2 \\
& \quad + 0.00007379 y_3^2 \leq 1, \\
& 2275.1327 y_3^{-1} y_5^{-1} - 0.2668 y_1 y_5^{-1} \\
& \quad - 0.40584 y_4 y_5^{-1} \leq 1, \\
& 0.00029955 y_3 y_5 + 0.00007992 y_1 y_3 \\
& \quad + 0.00012157 y_3 y_4 \leq 1, \\
& 78.0 \leq y_1 \leq 102.0, \quad 33.0 \leq y_2 \leq 45.0, \\
& 27.0 \leq y_3 \leq 45.0, \quad 27.0 \leq y_4 \leq 45.0, \\
& 27.0 \leq y_5 \leq 45.0.
\end{aligned} \tag{58}$$

Example 6 (see [21, 24, 34]). Consider

$$\begin{aligned}
\min \quad & 5x_1 + 5 \times 10^4 x_1^{-1} + 46.2 x_2 + 7.2 \times 10^4 x_1^{-1} \\
& \quad + 1.44 \times 10^5 x_3^{-1} \\
\text{s.t.} \quad & 4x_1^{-1} + 32 x_2^{-1} + 120 x_3^{-1} \leq 1, \\
& 1 \leq x_1, x_2, x_3 \leq 220.
\end{aligned} \tag{59}$$

Example 7 (see [24]). Consider

$$\begin{aligned}
\min \quad & -4x_2 + (x_1 - 1)^2 + x_2^2 - 10 x_3^2 \\
\text{s.t.} \quad & x_1^2 + x_2^2 + x_3^2 \leq 2, \\
& (x_1 - 2)^2 + x_2^2 + x_3^2 \leq 2, \\
& 2 - \sqrt{2} \leq x_1 \leq \sqrt{2}, \quad -\sqrt{2} \leq x_2 \leq \sqrt{2}, \\
& -\sqrt{2} \leq x_3 \leq \sqrt{2}.
\end{aligned} \tag{60}$$

Example 8 (see [22]). Consider

$$\begin{aligned}
\min \quad & x_{12} (12.62626 - 1.231059 y_1) \\
& + x_{13} (12.62626 - 1.231059 x_2) \\
& + x_{14} (12.62626 - 1.231059 y_3) \\
& + x_{15} (12.62626 - 1.231059 x_4) \\
& + x_{16} (12.62626 - 1.231059 x_5) \\
\text{s.t.} \quad & x_{12} - x_{11} \leq 0, \quad x_{11} - x_{12} \leq 50, \\
& x_{10} - x_4 \leq 0, \quad x_9 - x_{10} \leq 0, \\
& x_8 - x_9 \leq 0, \quad 2x_7 - x_1 \leq 1, \\
& x_3 - x_4 \leq, \quad x_3 x_2 - x_3 \leq 0, \\
& x_1 - x_2 \leq 0, \\
& x_4 x_{16} - 50 x_4 - x_5 x_{16} \leq -450, \\
& 50 x_4 + x_5 x_{16} + x_{10} x_{15} - 50 x_{10} \\
& - x_4 x_{15} - x_4 x_{16} \leq 0, \\
& 50 x_{10} + x_4 x_5 + x_9 x_{14} - 50 x_9 \\
& - x_3 x_{14} - x_8 x_{15} \leq 0, \\
& 50 x_8 + 50 x_9 + x_3 x_{14} + x_8 x_{13} \\
& - x_2 x_{13} - x_9 x_{14} \leq 500, \\
& 50 x_7 + x_2 x_{13} + x_7 x_{12} - 50 x_8 \\
& - x_1 x_{12} - x_8 x_{13} \leq 0, \\
& 50 x_8 + x_1 x_{12} + x_8 x_{13} - 50 x_7 \\
& - x_2 x_{13} - x_7 x_{12} \leq 0, \\
& x_6 x_{11} + x_1 x_{12} + x_7 x_{11} - x_6 x_{12} \leq 0, \\
& 100 x_6 + 0.0975 x_1^2 - 3.475 x_1 - 9.75 x_1 x_6 \leq 0, \\
& 100 x_7 + 0.0975 x_2^2 - 3.475 x_2 - 9.75 x_2 x_7 \leq 0, \\
& 100 x_8 + 0.0975 x_3^2 - 3.475 x_3 - 9.75 x_3 x_8 \leq 0,
\end{aligned}$$

Table 1: The numerical results for Examples 1–4.

Example		Optimal solution	Optimal value	Iter	ε
1	Ours	(0.1, 9.9999, 8.0, 0.2)	0.7651	58	10^{-5}
	[22]	(0.1, 9.9999, 8.0, 0.2)	0.7651	132	10^{-5}
	[31]	(0.1015, 7.331972, 8.0169, 0.2395)	0.9514	175	10^{-5}
	[32]	(0.1358, 9.9324, 8.6973, 0.2365)	1.0000	171	10^{-5}
2	Ours	(7.8922, 0.1002, 450.0000)	7.8922	14	10^{-2}
	[22]	(12.0475, 0.8167, 444.9416)	12.0475	6472	10^{-2}
	[32]	(11.9538, 0.8150, 445.1249)	11.9538	67	10^{-2}
3	Ours	(149.99608, 29.9530, 3.9456)	−147.6591	156	10^{-2}
	[22]	(150, 30, 4.9620)	−147.6667	328	10^{-2}
	[21]	(88.724706796, 7.672652781, 1.317862596)	−83.2497...	1829	10^{-2}
	[31]	(88.6274, 7.9621, 1.3215)	−83.6898	1754	10^{-2}
	[33]		−83.2497, ...	1809	10^{-2}
	[24]	(88.875643887, 7.5637589, 1.3124563877)	−83.661573642	754	10^{-8}
4	Ours	(43.1601, 44.9944, 69.9968, 1.2241)	460224.676188	23	10^{-1}
	[22]	(43.0473, 44.9317, 69.9359, 1.1338)	461200.00	968	10^{-1}
	[21]	(43.0137, ..., 44.8148, ..., 66.4239, ..., 1.1070, ...)	623249.876, ...	2100	10^{-1}
	[33]		623249.8752, ...	1717	10^{-1}
	[24]	(43.0899785, 44.9997852, 66.419945664, 1.1069987564)	468479.996875421	1987	10^{-8}

Table 2: The numerical results for Examples 5–10.

Example		Optimal solution	Optimal value	Iter	ε
5	Ours	(78.0, 33.0, 29.9957, 44.9999, 36.7753)	10122.4932	92	10^{-1}
	[22]	(78.2135, 33.2135, 29.6588, 44.757, 37.6808)	10088.51	122	10^{-1}
	[21]	(78, 32.9999, ..., 29.9957, ..., 45, 36.7753, ...)	10122.4931, ...	341	10^{-1}
	[33]		10122.3811, ...	204	10^{-1}
	[24]	(78, 32.99998, 29.99737, 45, 36.77533)	10122.85643	324	10^{-4}
6	Ours	(175.3433, 74.119872, 219.9999)	5651.37804758	2319	10^{-6}
	[24]	(109.32546781, 84.04821454, 214.32459429)	6217.46548921	612	10^{-8}
	[21]	(108.734706796, 85.126214158, 204.32459429)	6299.842427922	550	
	[34]	(107.4, 84.9, 204.5)	6300.00		
7	Ours	(0.9999, 0.18181812, 0.98333213)	−10.363636	837	10^{-6}
	[24]	(0.99712235, 0.18184214, 0.98034321)	−10.305022, ...	3	10^{-8}
8	Ours	(8.037732, 8.9999, 9, 9, 9, 1, 1, 1.15686, 1.15686, 1.15686, 50, 0, 1, 50, 50, 0)	156.2196	67	10^{-5}
	[22]	(8.03772, 9, 9, 9, 9, 1, 1, 1.1568, 1.1568, 1.1568, 50, 0, 1, 50, 50, 0)	156.2196	3	10^{-5}
9	Ours	(5.0, 0.021584, 0.044603, 5.0)	5.8894162	3	10^{-1}
	[22]	(4.99671, 0.02158, 0.044603, 4.99584)	5.888618	12097	10^{-1}
10	Ours	(5.0, 5.0, 0.13996, 1.177798, 0.94773)	28660.8648	2831	10^{-1}
	[22]	(4.992904, 4.99136, 0.1460728, 1.173758, 0.95455)	28745.107539	12014	10^{-1}

$$\begin{aligned}
&100x_9 + 0.0975x_4^2 - 3.475x_4 - 9.75x_4x_9 \leq 0,\\
&100x_{10} + 0.0975x_5^2 - 3.475x_5 - 9.75x_5x_{10} \leq 0,\\
&x \geq (1, 1, 9, 9, 9, 1, 1, 1, 1, 1, 50, 0, 1, 50, 50, 0),\\
&x \leq (8.037732, 9, 9, 9, 9, 1, 4.518866, 9, 9, 9, 100,\\
&\qquad 50, 50, 50, 50, 0).
\end{aligned} \tag{61}$$

Example 9 (see [22]). Consider

$$\begin{aligned}
\min \quad & (3 + x_1x_3)(x_1x_2x_3x_4 + 2x_1x_3 + 2)^{2/3}\\
\text{s.t.} \quad & -3(2x_1x_2 + 3x_1x_2x_4)(2x_1x_3 + 4x_1x_4 - x_2)\\
& \quad - (x_1x_3 + 3x_1x_2x_4)\\
& \quad \times (4x_3x_4 + 4x_1x_3x_4 + x_1x_3 - 4x_1x_2x_4)^{1/3}
\end{aligned}$$

$$\begin{aligned}
&+3\left(x_4+3x_1x_3x_4\right)\\
&\times\left(3x_1x_2x_3+3x_1x_4+2x_3x_4-3x_1x_2x_4\right)^{1/4}\\
&\leq -309.219315,\\
&-2\left(3x_3+3x_1x_2x_3\right)\left(x_1x_2x_3+4x_2x_4-x_3x_4\right)^2\\
&+\left(3x_1x_2x_3\right)\left(3x_3+2x_1x_2x_3+3x_4\right)^4\\
&-\left(x_2x_3x_4+x_1x_3x_4\right)\left(4x_1-1\right)^{3/4}\\
&-3\left(3x_3x_4+2x_1x_3x_4\right)\\
&\times\left(x_1x_2x_3x_4+x_3x_4-4x_1x_2x_3-2x_1\right)^4\\
&\leq -78243.910551,\\
&-3\left(4x_1x_3x_4\right)\left(2x_4+2x_1x_2-x_2-x_3\right)^2\\
&+2\left(x_1x_2x_4+3x_1x_3x_4\right)\\
&\times\left(x_1x_2+2x_2x_3+4x_2-x_2x_3x_4-x_1x_3\right)^4\\
&\leq 9618,\\
&0\leq x_i\leq 5,\quad i=1,\ldots,4.
\end{aligned}\tag{62}$$

Example 10 (see [22]). Consider

$$\begin{aligned}
\min\quad & 4\left(x_1^2x_3+2x_1^2x_2x_3^2x_5+2x_1^2x_2x_3\right)\\
&\times\left(5x_1^2x_3x_4^2x_5+3x_2\right)^{3/5}\\
&+3\left(2x_4^2x_5^2\right)\left(4x_1^2x_4+4x_2x_5\right)^{5/3}\\
\text{s.t.}\quad & -2\left(2x_1x_5+5x_1^2x_2x_4^2x_5\right)\left(3x_1x_4x_5^2+5+4x_3x_5^2\right)^{1/2}\\
&\leq -7684.470329,\\
& 2\left(2x_1x_2^2x_3x_4^2\right)\left(2x_1x_2x_3x_4^2+2x_2x_4^2x_5-x_1^2x_5^2\right)^{3/2}\\
&\leq 1286590.314422,\\
& 0\leq x_i\leq 5,\quad i=1,\ldots,5.
\end{aligned}\tag{63}$$

Tables 1 and 2 summarize the computational results on the above examples, where Iter denotes the number of algorithm iteration.

From the computational results, we can see that the proposed BRB algorithm can solve the problem (SGP) effectively. This illustrates the potential advantage of the proposed algorithm: not only is a feasible optimal solution obtained, but also less computational effort may be required for finding a better objective function value.

6. Conclusion

To globally solve the problem (SGP), a new branch-reduce-bound algorithm is proposed, based on an equivalent monotonic optimization problem and a linear relaxation method. The algorithm can attain the global minimum through the successive refinement of a linear relaxation and the subsequent solutions of a series of linear programming problems. To improve the convergence speed, two range reduction operations are proposed, which can cut away a large part of the region in which the global optimal solution of (SGP) does not exist. The convergence of the algorithm is proved and numerical results are reported to vindicate the feasibility and effectiveness of the proposed algorithm.

Conflict of Interests

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

Acknowledgment

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

References

[1] P. Hansen and B. Jaumard, "Reduction of indefinite quadratic programs to bilinear programs," *Journal of Global Optimization*, vol. 2, no. 1, pp. 41–60, 1992.

[2] C. S. Beightler and D. T. Phillips, *Applied Geometric Programming*, John Wiley & Sons, New York, NY, USA, 1976.

[3] M. Avriel and A. C. Williams, "An extension of geometric programming with applications in engineering optimization," *Journal of Engineering Mathematics*, vol. 5, no. 2, pp. 187–194, 1971.

[4] N. K. Jha, "Geometric programming based robot control design," *Computers and Industrial Engineering*, vol. 29, no. 1–4, pp. 631–635, 1995.

[5] T. R. Jefferson and C. H. Scott, "Generalized geometric programming applied to problems of optimal control. I. Theory," *Journal of Optimization Theory and Applications*, vol. 26, no. 1, pp. 117–129, 1978.

[6] J. G. Ecker, "Geometric programming: methods, computations and applications," *SIAM Review*, vol. 22, no. 3, pp. 338–362, 1980.

[7] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, Springer, Berlin, Germany, 2nd edition, 1993.

[8] C. A. Floudas and V. Visweswaran, "Quadratic optimization," in *Handbook of Global Optimization, Nonconvex Optimization and Its Applications*, R. Horst and P. M. Pardalos, Eds., vol. 2, pp. 217–269, Kluwer Academic Publishers, 1995.

[9] Y. K. Sui, "The expansion of functions under transformation and its application to optimization," *Computer Methods in Applied Mechanics and Engineering*, vol. 113, no. 3-4, pp. 253–262, 1994.

[10] 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.

[11] R. J. Duffin and E. L. Peterson, "Duality theory for geometric programming," *SIAM Journal on Applied Mathematics*, vol. 14, pp. 1307–1349, 1966.

[12] R. J. Duffin, E. L. Peterson, and C. Zener, "The geometric inequality and the main lemma," in *Geometric Programming Theory and Applocations*, pp. 115–140, John Wiley & Sons, New York, NY, USA, 1967.

[13] R. J. Duffin and E. L. Peterson, "Geometric programming with signomials," *Journal of Optimization Theory and Applications*, vol. 11, pp. 3–35, 1973.

[14] U. Passy, "Generalized weighted mean programming," *SIAM Journal on Applied Mathematics*, vol. 20, pp. 763–778, 1971.

[15] U. Passy and D. J. Wilde, "Generalized polynomial optimization," *SIAM Journal on Applied Mathematics*, vol. 15, pp. 1344–1356, 1967.

[16] K. O. Kortanek, X. Xu, and Y. Ye, "An infeasible interior-point algorithm for solving primal and dual geometric programs," *Mathematical Programming*, vol. 76, no. 1, pp. 155–181, 1997.

[17] H. D. Sherali and C. H. Tuncbilek, "A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique," *Journal of Global Optimization*, vol. 2, no. 1, pp. 101–112, 1992.

[18] H. D. Sherali and C. H. Tuncbilek, "A reformulation-convexification approach for solving nonconvex quadratic programming problems," *Journal of Global Optimization*, vol. 7, no. 1, pp. 1–31, 1995.

[19] H. D. Sherali, "Global optimization of nonconvex polynomial programming problems having rational exponents," *Journal of Global Optimization*, vol. 12, no. 3, pp. 267–283, 1998.

[20] C. D. Maranas and C. A. Floudas, "Global optimization in generalized geometric programming," *Computers and Chemical Engineering*, vol. 21, no. 4, pp. 351–369, 1997.

[21] P. P. Shen and K. C. Zhang, "Global optimization of signomial geometric programming using linear relaxation," *Applied Mathematics and Computation*, vol. 150, no. 1, pp. 99–114, 2004.

[22] P. P. Shen, Y. Ma, and Y. Q. Chen, "A robust algorithm for generalized geometric programming," *Journal of Global Optimization*, vol. 41, no. 4, pp. 593–612, 2008.

[23] Y. J. Wang, T. Li, and Z. A. Liang, "A general algorithm for solving generalized geometric programming with nonpositive degree of difficulty," *Computational Optimization and Applications*, vol. 44, no. 1, pp. 139–158, 2009.

[24] S. J. Qu, K. C. Zhang, and F. S. Wang, "A global optimization using linear relaxation for generalized geometric programming," *European Journal of Operational Research*, vol. 190, no. 2, pp. 345–356, 2008.

[25] H. Tuy, "Effect of the subdivision strategy on convergence and efficiency of some global optimization algorithms," *Journal of Global Optimization*, vol. 1, no. 1, pp. 23–36, 1991.

[26] J.-M. Peng and Y.-X. Yuan, "Optimality conditions for the minimization of a quadratic with two quadratic constraints," *SIAM Journal on Optimization*, vol. 7, no. 3, pp. 579–594, 1997.

[27] P. P. Shen, X. D. Bai, and W. M. Li, "A new accelerating method for globally solving a class of nonconvex programming problems," *Nonlinear Analysis. Theory, Methods & Applications*, vol. 71, no. 7-8, pp. 2866–2876, 2009.

[28] R. Horst, P. M. Pardalos, and N. V. Thoai, *Introduction to Global Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[29] R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, Springer, Berlin, Germany, 3rd edition, 2003.

[30] R. Horst, "Deterministic global optimization with partition sets whose feasibility is not known: application to concave minimization, reverse convex constraints, DC-programming, and Lipschitzian optimization," *Journal of Optimization Theory and Applications*, vol. 58, no. 1, pp. 11–37, 1988.

[31] S.-J. Qu, K.-C. Zhang, and Y. Ji, "A new global optimization algorithm for signomial geometric programming via Lagrangian relaxation," *Applied Mathematics and Computation*, vol. 184, no. 2, pp. 886–894, 2007.

[32] Y. J. Wang and Z. A. Liang, "A deterministic global optimization algorithm for generalized geometric programming," *Applied Mathematics and Computation*, vol. 168, no. 1, pp. 722–737, 2005.

[33] P. P. Shen and H. W. Jiao, "A new rectangle branch-and-pruning approach for generalized geometric programming," *Applied Mathematics and Computation*, vol. 183, no. 2, pp. 1027–1038, 2006.

[34] M. J. Rijckaert and X. M. Martens, "Comparison of generalized geometric programming algorithms," *Journal of Optimization Theory and Application*, vol. 26, pp. 205–241, 1978.