

## Research Article

# A New Global Optimization Algorithm for Solving Generalized Geometric Programming

San-Yang Liu, Chun-Feng Wang, and Li-Xia Liu

Department of Mathematical Sciences, Xidian University, Xi'an 710071, China

Correspondence should be addressed to Chun-Feng Wang, wangchunfeng09@126.com

Received 13 October 2010; Accepted 23 December 2010

Academic Editor: J. J. Judice

Copyright © 2010 San-Yang Liu 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.

A global optimization algorithm for solving generalized geometric programming (GGP) problem is developed based on a new linearization technique. Furthermore, in order to improve the convergence speed of this algorithm, a new pruning technique is proposed, which can be used to cut away a large part of the current investigated region in which the global optimal solution does not exist. Convergence of this algorithm is proved, and some experiments are reported to show the feasibility of the proposed algorithm.

## 1. Introduction

This paper considers generalized geometric programming GGP problem in the following form:

$$(GGP) \begin{cases} \min & \phi_0(x) \\ \text{s.t.} & \phi_j(x) \leq \beta_j, \quad j = 1, \dots, m, \\ & X^0 = \{x : 0 < l^0 \leq x \leq u^0\}, \end{cases} \quad (1.1)$$

where  $\phi_j(x) = \sum_{t=1}^{T_j} c_{jt} \prod_{i=1}^n x_i^{\gamma_{jti}}$ ,  $c_{jt}, \beta_j, \gamma_{jti} \in \mathbb{R}$ ,  $t = 1, \dots, T_j$ ,  $i = 1, 2, \dots, n$ ,  $j = 0, 1, \dots, m$ .

Generally speaking, GGP problem is a non convex programming problem, which has a wide variety of applications, such as in engineering design, economics and statistics, manufacturing, and distribution contexts in risk management problems [1–4].

During the past years, many local optimization approaches for solving GGP problem have been presented [5, 6], but global optimization algorithms based on the characteristics of GGP problem are scarce. Maranas and Floudas [7] proposed a global optimization algorithm

for solving GGP problem based on convex relaxation. Shen and Zhang [8] presented a method to globally solve GGP problem by using linear relaxation. Recently, several branch and bound algorithms have been developed [9, 10].

The purpose of this paper is to introduce a new global optimization algorithm for solving GGP problem. In this algorithm, by utilizing the special structure of GGP problem, a new linear relaxation technique is presented. Based on this technique, the initial GGP problem is systematically converted into a series of linear programming problems. The solutions of these converted problems can approximate the global optimal solution of GGP problem by successive refinement process.

The main features of this algorithm are: (1) a new linearization technique for solving GGP problem is proposed, which applies more information of the functions of GGP problem, (2) the generated relaxation linear programming problems are embedded within a branch and bound algorithm without increasing the number of variables and constraints, (3) a new pruning technique is presented, which can be used to improve the convergence speed of the proposed algorithm, and (4) numerical experiments are given, which show that the proposed algorithm can treat all of the test problems in finding global optimal solution within a prespecified tolerance.

The structure of this paper is as follows. In Section 2, first, we construct the lower approximate linear functions for the objective function and the constraint functions of GGP problem; then, we derive the relaxation linear programming (RLP) problem of GGP problem; finally, to improve the convergence speed of our algorithm, we present a new pruning technique. In Section 3, the proposed branch and bound algorithm is described, and the convergence of the algorithm is established. Some numerical results are reported in Section 4.

## 2. Linear Relaxation and Pruning Technique

The principal structure in the development of a solution procedure for solving GGP problem is the construction of lower bounds for this problem, as well as for its partitioned subproblems. A lower bound of GGP problem and its partitioned subproblems can be obtained by solving a linear relaxation problem. The proposed strategy for generating this linear relaxation problem is to underestimate every nonlinear function  $\phi_j(x)$  ( $j = 0, \dots, m$ ) with a linear function. In what follows, all the details of this procedure will be given.

Let  $X = [\underline{x}, \bar{x}]$  represents either the initial box  $X^0$ , or modified box as defined for some partitioned subproblem in a branch and bound scheme.

Consider term  $\prod_{i=1}^n x_i^{\gamma_{ji}}$  in  $\phi_j(x)$  ( $j = 0, \dots, m$ ). Let  $\prod_{i=1}^n x_i^{\gamma_{ji}} = \exp(y_{jt})$ , then, we have

$$y_{jt} = \sum_{i=1}^n \gamma_{jti} \ln x_i. \quad (2.1)$$

From (2.1), we can obtain the lower bound and upper bound of  $y_{jt}$  as follows:

$$\underline{y}_{jt} = \sum_{i=1}^n \min\{\gamma_{jti} \ln \underline{x}_i, \gamma_{jti} \ln \bar{x}_i\}, \quad \bar{y}_{jt} = \sum_{i=1}^n \max\{\gamma_{jti} \ln \underline{x}_i, \gamma_{jti} \ln \bar{x}_i\}. \quad (2.2)$$

To derive the linear relaxation problem, we will use a convex separation technique and a two-part relaxation technique.

### 2.1. First-Part Relaxation

Let  $\mathbf{y}_j = (y_{j1}, \dots, y_{jT_j})$ , then  $\phi_j(x)$  can be expressed in the following form:

$$\phi_j(x) = \sum_{t=1}^{T_j} c_{jt} \prod_{i=1}^n x_i^{y_{jt}} = \sum_{t=1}^{T_j} c_{jt} \exp(y_{jt}) \triangleq f_j(\mathbf{y}_j). \quad (2.3)$$

For  $f_j(\mathbf{y}_j)$ , we can derive its gradient and the Hessian matrix:

$$\begin{aligned} \nabla f_j(\mathbf{y}_j) &= \begin{pmatrix} c_{j1} \exp(y_{j1}) \\ \vdots \\ c_{jT_j} \exp(y_{jT_j}) \end{pmatrix}, \\ \nabla^2 f_j(\mathbf{y}_j) &= \begin{pmatrix} c_{j1} \exp(y_{j1}) & & & \\ & c_{j2} \exp(y_{j2}) & & \\ & & \ddots & \\ & & & c_{jT_j} \exp(y_{jT_j}) \end{pmatrix}. \end{aligned} \quad (2.4)$$

Therefore, by (2.4), the following relation holds:

$$\|\nabla^2 f_j(\mathbf{y}_j)\| \leq \max_{1 \leq t \leq T_j} |c_{jt}| \max_{1 \leq t \leq T_j} \exp(\bar{y}_{jt}). \quad (2.5)$$

Let  $\lambda_j = \max_{1 \leq t \leq T_j} |c_{jt}| \max_{1 \leq t \leq T_j} \exp(\bar{y}_{jt}) + 0.1$ , then, for all  $\mathbf{y}_j \in Y_j \triangleq [\underline{\mathbf{y}}_j, \bar{\mathbf{y}}_j]$ , we have

$$\|\nabla^2 f_j(\mathbf{y}_j)\| < \lambda_j. \quad (2.6)$$

Thus, the function  $(1/2)\lambda_j \|\mathbf{y}_j\|^2 + f_j(\mathbf{y}_j)$  is a convex function on  $Y_j$ . Consequently, the function  $f_j(\mathbf{y}_j)$  can be decomposed into the difference of two convex functions, that is,  $f_j(\mathbf{y}_j)$  is a d.c. function, which admits the following d.c. decomposition:

$$f_j(\mathbf{y}_j) = g_j(\mathbf{y}_j) - h_j(\mathbf{y}_j), \quad (2.7)$$

where

$$g_j(\mathbf{y}_j) = \frac{1}{2} \lambda_j \|\mathbf{y}_j\|^2 + f_j(\mathbf{y}_j), \quad h_j(\mathbf{y}_j) = \frac{1}{2} \lambda_j \|\mathbf{y}_j\|^2. \quad (2.8)$$

Let  $\mathbf{y}_{j\text{mid}} = (1/2)(\underline{\mathbf{y}}_j + \bar{\mathbf{y}}_j)$ . Since  $g_j(\mathbf{y}_j)$  is a convex function, we have

$$g_j(\mathbf{y}_j) \geq g_j(\mathbf{y}_{j\text{mid}}) + \nabla g_j(\mathbf{y}_{j\text{mid}})^T (\mathbf{y}_j - \mathbf{y}_{j\text{mid}}) \triangleq g_j^l(\mathbf{y}_j). \quad (2.9)$$

In addition, for  $\mathbf{y}_{jt} \in [\underline{\mathbf{y}}_{jt}, \bar{\mathbf{y}}_{jt}]$ , it is not difficult to show

$$\left( \underline{\mathbf{y}}_{jt} + \bar{\mathbf{y}}_{jt} \right) \mathbf{y}_{jt} - \underline{\mathbf{y}}_{jt} \bar{\mathbf{y}}_{jt} \geq \mathbf{y}_{jt}^2. \quad (2.10)$$

Furthermore, we can obtain

$$\sum_{t=1}^{T_j} \left[ \left( \underline{\mathbf{y}}_{jt} + \bar{\mathbf{y}}_{jt} \right) \mathbf{y}_{jt} - \underline{\mathbf{y}}_{jt} \bar{\mathbf{y}}_{jt} \right] \geq \|\mathbf{y}_j\|^2. \quad (2.11)$$

Since  $\lambda_j > 0$ , it follows that

$$h_j^l(\mathbf{y}_j) \triangleq \frac{1}{2} \lambda_j \sum_{t=1}^{T_j} \left[ \left( \underline{\mathbf{y}}_{jt} + \bar{\mathbf{y}}_{jt} \right) \mathbf{y}_{jt} - \underline{\mathbf{y}}_{jt} \bar{\mathbf{y}}_{jt} \right] \geq \frac{1}{2} \lambda_j \|\mathbf{y}_j\|^2 = h_j(\mathbf{y}_j). \quad (2.12)$$

Thus, from (2.7), (2.9), and (2.12), we have

$$f_j^l(\mathbf{y}_j) \triangleq g_j^l(\mathbf{y}_j) - h_j^l(\mathbf{y}_j) \leq f_j(\mathbf{y}_j). \quad (2.13)$$

Hence, by (2.1), (2.3), and (2.13), the first-part relaxation  $L_j(\mathbf{x})$  of  $\phi_j(\mathbf{x})$  can be obtained as follows

$$\begin{aligned} \phi_j(\mathbf{x}) &= f_j(\mathbf{y}_j) = g_j(\mathbf{y}_j) - h_j(\mathbf{y}_j) \\ &\geq g_j^l(\mathbf{y}_j) - h_j^l(\mathbf{y}_j) \\ &= g_j(\mathbf{y}_{j\text{mid}}) + \nabla g_j(\mathbf{y}_{j\text{mid}})^T (\mathbf{y}_j - \mathbf{y}_{j\text{mid}}) - \frac{1}{2} \lambda_j \sum_{t=1}^{T_j} \left[ \left( \underline{\mathbf{y}}_{jt} + \bar{\mathbf{y}}_{jt} \right) \mathbf{y}_{jt} - \underline{\mathbf{y}}_{jt} \bar{\mathbf{y}}_{jt} \right] \\ &= \sum_{t=1}^{T_j} \left[ c_{jt} \exp(\mathbf{y}_{j\text{mid}}(t)) + \lambda_j \mathbf{y}_{j\text{mid}}(t) - \frac{1}{2} \lambda_j \left( \underline{\mathbf{y}}_{jt} + \bar{\mathbf{y}}_{jt} \right) \right] \mathbf{y}_{jt} \\ &\quad + g_j(\mathbf{y}_{j\text{mid}}) - \nabla g_j(\mathbf{y}_{j\text{mid}})^T \mathbf{y}_{j\text{mid}} + \frac{1}{2} \lambda_j \sum_{t=1}^{T_j} \underline{\mathbf{y}}_{jt} \bar{\mathbf{y}}_{jt} \\ &= \sum_{t=1}^{T_j} \left[ c_{jt} \exp(\mathbf{y}_{j\text{mid}}(t)) + \lambda_j \mathbf{y}_{j\text{mid}}(t) - \frac{1}{2} \lambda_j \left( \underline{\mathbf{y}}_{jt} + \bar{\mathbf{y}}_{jt} \right) \right] \sum_{i=1}^n \gamma_{jti} \ln x_i \\ &\quad + g_j(\mathbf{y}_{j\text{mid}}) - \nabla g_j(\mathbf{y}_{j\text{mid}})^T \mathbf{y}_{j\text{mid}} + \frac{1}{2} \lambda_j \sum_{t=1}^{T_j} \underline{\mathbf{y}}_{jt} \bar{\mathbf{y}}_{jt} \\ &= \sum_{i=1}^n \left( \sum_{t=1}^{T_j} \left( c_{jt} \exp(\mathbf{y}_{j\text{mid}}(t)) + \lambda_j \mathbf{y}_{j\text{mid}}(t) - \frac{1}{2} \lambda_j \left( \underline{\mathbf{y}}_{jt} + \bar{\mathbf{y}}_{jt} \right) \right) \gamma_{jti} \right) \ln x_i \\ &\quad + g_j(\mathbf{y}_{j\text{mid}}) - \nabla g_j(\mathbf{y}_{j\text{mid}})^T \mathbf{y}_{j\text{mid}} + \frac{1}{2} \lambda_j \sum_{t=1}^{T_j} \underline{\mathbf{y}}_{jt} \bar{\mathbf{y}}_{jt} \triangleq L_j(\mathbf{x}). \end{aligned} \quad (2.14)$$

## 2.2. Second-Part Relaxation

Consider the function  $\ln x_i$  on the interval  $[\underline{x}_i, \bar{x}_i]$ . As we all know, its linear lower bound function and upper bound function can be derived as follows:

$$K_i(x_i - \underline{x}_i) + \ln \underline{x}_i \leq \ln x_i \leq K_i x_i - 1 - \ln K_i, \quad (2.15)$$

where  $K_i = (\ln \bar{x}_i - \ln \underline{x}_i) / (\bar{x}_i - \underline{x}_i)$ .

In  $L_j(x)$  ( $j = 0, \dots, m$ ), let

$$\alpha_{ji} = \sum_{t=1}^{T_j} \left( c_{jt} \exp \left( (y_{j\text{mid}}(t) + \lambda_j y_{j\text{mid}}(t) - \frac{1}{2} \lambda_j (\underline{y}_{jt} + \bar{y}_{jt})) \right) \right), \quad i = 1, \dots, n. \quad (2.16)$$

Then, from (2.14) and (2.15), we can obtain the linear lower bound function of  $\phi_j(x)$  denoted by  $\phi_j^l(x)$  as follows:

$$\phi_j^l(x) \triangleq \sum_{i=1}^n \alpha_{ji} \varphi_i(x_i) + g_j(y_{j\text{mid}}) + \nabla g_j(y_{j\text{mid}})^T y_{j\text{mid}} + \frac{1}{2} \lambda_j \sum_{t=1}^{T_j} \underline{y}_{jt} \bar{y}_{jt}, \quad (2.17)$$

where

$$\varphi_i(x_i) = \begin{cases} K_i(x_i - \underline{x}_i) + \ln \underline{x}_i, & \text{if } \alpha_{ji} \geq 0, \\ K_i x_i - 1 - \ln K_i, & \text{else.} \end{cases} \quad (2.18)$$

Obviously,  $\phi_j^l(x) \leq L_j(x) \leq \phi_j(x)$ .

Consequently, the corresponding approximation relaxation linear programming (RLP) problem of GGP problem on  $X$  can be obtained

$$(\text{RLP}) \begin{cases} \min & \phi_0^l(x) \\ \text{s.t.} & \phi_j^l(x) \leq \beta_j, \quad j = 1, \dots, m, \\ & x \in X. \end{cases} \quad (2.19)$$

**Theorem 2.1.** Let  $\delta_i = \underline{x}_i - \bar{x}_i$ ,  $i = 1, \dots, n$ . Then, for all  $x \in X$ , the difference of  $\phi_j(x)$  and  $\phi_j^l(x)$  satisfies  $\phi_j(x) - \phi_j^l(x) \rightarrow 0$  as  $\delta_i \rightarrow 0$ ,  $i = 1, \dots, n$ ,  $j = 0, \dots, m$ .

*Proof.* For all  $x \in X$ , let

$$\Delta = \phi_j(x) - \phi_j^l(x) = \phi_j(x) - L_j(x) + L_j(x) - \phi_j^l(x), \quad (2.20)$$

and let  $\Delta^1 = \phi_j(x) - L_j(x)$ ,  $\Delta^2 = L_j(x) - \phi_j^l(x)$ . Then, it is obvious that we only need to prove

$$\Delta^1 \rightarrow 0, \quad \Delta^2 \rightarrow 0, \quad \text{as } \delta_i \rightarrow 0. \quad (2.21)$$

To this end, first, we consider the difference  $\Delta^1 = \phi_j(x) - L_j(x)$ . By (2.7), (2.13), and (2.14), it follows that

$$\begin{aligned}
\Delta^1 &= \phi_j(x) - L_j(x) \\
&= \sum_{t=1}^{T_j} c_{jt} \prod_{i=1}^n x_i^{y_{jt}^{ji}} - \left[ \sum_{t=1}^{T_j} \left( c_{jt} \exp \left( (y_{j\text{mid}}(t)) + \lambda_j y_{j\text{mid}}(t) - \frac{1}{2} \lambda_j (\underline{y}_{jt} + \bar{y}_{jt}) \right) \right) \right. \\
&\quad \left. \times \sum_{i=1}^n y_{jti} \ln x_i + g_j(y_{j\text{mid}}) - \nabla g_j(y_{j\text{mid}})^T y_{j\text{mid}} + \frac{1}{2} \lambda_j \sum_{t=1}^{T_j} \underline{y}_{jt} \bar{y}_{jt} \right] \\
&= \sum_{t=1}^{T_j} c_{jt} \exp(y_{jt}) - \left[ \sum_{t=1}^{T_j} \left( c_{jt} \exp \left( (y_{j\text{mid}}(t)) + \lambda_j y_{j\text{mid}}(t) - \frac{1}{2} \lambda_j (\underline{y}_{jt} + \bar{y}_{jt}) \right) \right) y_{jt} \right. \\
&\quad \left. + g_j(y_{j\text{mid}}) - \nabla g_j(y_{j\text{mid}})^T y_{j\text{mid}} + \frac{1}{2} \lambda_j \sum_{t=1}^{T_j} \underline{y}_{jt} \bar{y}_{jt} \right] \tag{2.22} \\
&= f_j(y_j) - f_j^l(y_j) \\
&= g_j(y_j) - g_j^l(y_j) + \frac{1}{2} \lambda_j \|y_j\|^2 - \frac{1}{2} \lambda_j \sum_{t=1}^{T_j} \left[ \left( \underline{y}_{jt} + \bar{y}_{jt} \right) y_{jt} - \underline{y}_{jt} \bar{y}_{jt} \right] \\
&\leq (\nabla g_j(\xi) - \nabla g_j(y_{j\text{mid}}))^T (y_j - y_{j\text{mid}}) + \frac{1}{2} \lambda_j \left\| \bar{y}_j - \underline{y}_j \right\|^2 \\
&\leq 2\lambda_j \|\xi - y_{j\text{mid}}\| \|y_j - y_{j\text{mid}}\| + \frac{1}{2} \lambda_j \left\| \bar{y}_j - \underline{y}_j \right\|^2 \\
&\leq \frac{5}{2} \lambda_j \left\| \bar{y}_j - \underline{y}_j \right\|^2,
\end{aligned}$$

where  $\xi$  is a constant vector, and satisfies  $g_j(y_j) - g_j(y_{j\text{mid}}) = \nabla g_j(\xi)^T (y_j - y_{j\text{mid}})$ .

By the definition of  $y_{jt}$ , we have  $\|\bar{y}_j - \underline{y}_j\| \rightarrow 0$  as  $\delta_i \rightarrow 0$  ( $i = 1, \dots, n$ ). Thus, we have  $\Delta^1 \rightarrow 0$  as  $\delta_i \rightarrow 0$  ( $i = 1, \dots, n$ ).

Second, we consider the difference  $\Delta^2 = L_j(x) - \phi_j^l(x)$ . From the definitions of  $L_j(x)$  and  $\phi_j^l(x)$ , it follows that

$$\begin{aligned}
\Delta^2 &= L_j(x) - \phi_j^l(x) \\
&= \sum_{i=1}^n \alpha_{ji} (\ln x_i - \varphi_i(x_i)) \\
&= \sum_{\alpha_{ji} \geq 0} \alpha_{ji} (\ln x_i - K_i(x_i - \underline{x}_i) - \ln \underline{x}_i) + \sum_{\alpha_{ji} < 0} \alpha_{ji} (\ln x_i - K_i x_i + 1 + \ln K_i).
\end{aligned} \tag{2.23}$$

By [8], we know that  $|\ln x_i - K_i(x_i - \underline{x}_i) - \ln \underline{x}_i| \rightarrow 0$  and  $|\ln x_i - K_i x_i + 1 + \ln K_i| \rightarrow 0$ , as  $\delta_i \rightarrow 0$  ( $i = 1, \dots, n$ ). Thus, we have  $\Delta^2 \rightarrow 0$  as  $\delta_i \rightarrow 0$  ( $i = 1, \dots, n$ ).

Taken together the above, it implies that

$$\Delta = \Delta^1 + \Delta^2 \rightarrow 0, \quad \text{as } \delta_i \rightarrow 0 \quad (i = 1, \dots, n), \quad (2.24)$$

and this completes the proof.  $\square$

From Theorem 2.1, it follows that  $\phi_j^l(x)$  will approximate the function  $\phi_j(x)$  as  $\delta_i \rightarrow 0$  ( $i = 1, \dots, n$ ).

Based on the above discussion, it can be seen that the optimal value of RLP problem is smaller than or equal to that of GGP problem for all feasible points, that is, the optimal value of RLP problem provides a valid lower bound for the optimal value of GGP problem. Thus, for any problem (P), let us denote the optimal value of (P) by  $V(P)$ , then we have  $V(\text{RLP}) \leq V(\text{GGP})$ .

### 2.3. Pruning Technique

In order to improve the convergence speed of this algorithm, we present a new pruning technique, which can be used to eliminate the region in which the global optimal solution of GGP problem does not exist.

Assume that UB is current known upper bound of the optimal value  $\phi_0^*$  of GGP problem. Let

$$\tau_i = \alpha_{0i} K_i, \quad i = 1, \dots, n,$$

$$\bar{T} = \sum_{i=1}^n \bar{T}_i, \quad \text{where } \bar{T}_i = \begin{cases} \alpha_{0i} (\ln \underline{x}_i - K_i \underline{x}_i), & \text{if } \alpha_{0i} \geq 0, \\ \alpha_{0i} (-1 - \ln K_i), & \text{else,} \end{cases} \quad (2.25)$$

$$T = \bar{T} + g_0(\mathbf{y}_{0\text{mid}}) - \nabla g_0(\mathbf{y}_{0\text{mid}})^T \mathbf{y}_{0\text{mid}} + \frac{1}{2} \lambda_0 \sum_{t=1}^{T_0} \mathbf{y}_{-0t} \bar{\mathbf{y}}_{0t}.$$

**Theorem 2.2.** For any subrectangle  $X = (X_i)_{n \times 1} \subseteq X^0$  with  $X_i = [\underline{x}_i, \bar{x}_i]$ . Let

$$\rho_k = \text{UB} - \sum_{i=1, i \neq k}^n \min\{\tau_i \underline{x}_i, \tau_i \bar{x}_i\} - T, \quad k = 1, \dots, n. \quad (2.26)$$

If there exists some index  $k \in \{1, \dots, n\}$  such that  $\tau_k > 0$  and  $\rho_k < \tau_k \bar{x}_k$ , then there is no globally optimal solution of GGP problem on  $X^1$ ; if  $\tau_k < 0$  and  $\rho_k < \tau_k \underline{x}_k$ , then there is no globally optimal solution of GGP problem on  $X^2$ , where

$$X^1 = (X_i^1)_{n \times 1} \subseteq X, \quad \text{with } X_i^1 = \begin{cases} X_i, & i \neq k, \\ \left( \frac{\rho_k}{\tau_k}, \bar{x}_k \right] \cap X_i, & i = k, \end{cases} \quad (2.27)$$

$$X^2 = (X_i^2)_{n \times 1} \subseteq X, \quad \text{with } X_i^2 = \begin{cases} X_i, & i \neq k, \\ \left[ \underline{x}_k, \frac{\rho_k}{\tau_k} \right) \cap X_i, & i = k. \end{cases}$$

*Proof.* First, we show that for all  $x \in X^1$ ,  $\phi_0(x) > \text{UB}$ . When  $x \in X^1$ , consider the  $k$ th component  $x_k$  of  $x$ . Since  $x_k \in (\rho_k/\tau_k, \bar{x}_k] \cap X_k$ , it follows that

$$\frac{\rho_k}{\tau_k} < x_k \leq \bar{x}_k. \quad (2.28)$$

Note  $\tau_k > 0$ , we have  $\rho_k < \tau_k x_k$ . From the definition of  $\rho_k$  and the above inequality, we obtain

$$\begin{aligned} \text{UB} &< \sum_{i=1, i \neq k}^n \min\{\tau_i \underline{x}_i, \tau_i \bar{x}_i\} + \tau_k x_k + T \\ &\leq \sum_{i=1}^n \tau_i x_i + T \\ &= \sum_{i=1}^n \tau_i \varphi_i(x_i) + g_0(y_{0\text{mid}}) - \nabla g_0(y_{0\text{mid}})^T y_{0\text{mid}} + \frac{1}{2} \lambda_0 \sum_{t=1}^{T_0} y_{0t} \bar{y}_{0t} \\ &= \phi_0^l(x). \end{aligned} \quad (2.29)$$

This implies that, for all  $x \in X^1$ ,  $\phi_0(x) \geq \phi_0^l(x) > \text{UB} \geq \phi_0^*$ . In other words, for all  $x \in X^1$ ,  $\phi_0(x)$  is always greater than the optimal value of GGP problem. Therefore, there is no globally optimal solution of GGP problem on  $X^1$ .

For all  $x \in X^2$ , if  $\tau_k < 0$  and  $\rho_k < \tau_k \underline{x}_k$  with some  $k$ , by arguments similar to the above, we can derive that there is no globally optimal solution of GGP problem on  $X^2$ .  $\square$

### 3. Algorithm and Its Convergence

In this section, based on the former relaxation linear programming (RLP) problem, a branch and bound algorithm is presented to globally solve GGP problem. In order to ensure convergence to the global optimal solution, this algorithm needs to solve a sequence of (RLP) problems over partitioned subsets of  $X^0$ .

In this algorithm, the set  $X^0$  will be partitioned into subrectangles. Each subrectangle is concerned with a node of the branch and bound tree, and is associated with a relaxation linear subproblem.

At stage  $k$  of the algorithm, suppose that we have a collection of active nodes denoted by  $Q_k$ . For each node  $X \in Q_k$ , we will have computed a lower bound for the optimal value of GGP problem via solution  $\text{LB}(X)$  of RLP problem, so that the lower bound of the optimal value of GGP problem on the whole initial box region  $X^0$  at stage  $k$  is given by  $\text{LB}_k = \min\{\text{LB}(X), \text{for all } X \in Q_k\}$ . Whenever the solution of RLP problem for any node subproblem turns out to be feasible to GGP problem, we update the upper bound  $\text{UB}$  if necessary. Then, for each stage  $k$ , the active nodes collection  $Q_k$  will satisfy  $\text{LB}(X) < \text{UB}$ , for all  $X \in Q_k$ . We now select an active node to partition its associated rectangle into two subrectangles according to the following branching rule. For such two subrectangles, the fathoming step is applied in order to identify whether the subrectangles should be eliminated. In the end, we obtain a collection of active nodes for the next stage. This process is repeated until convergence is obtained.

### 3.1. Branching Rule

As we all know, the critical element in guaranteeing convergence to the global optimal solution is the choice of a suitable partitioning strategy. In this paper, we choose a simple and standard bisection rule. This rule is sufficient to ensure convergence since it drives all the intervals to a singleton for all variables. Consider any node subproblem identified by rectangle

$$X = \{x \in R^n \mid \underline{x}_i \leq x_i \leq \bar{x}_i, i = 1, \dots, n\} \subseteq X^0. \quad (3.1)$$

The branching rule is as follows.

- (a) Let  $s = \operatorname{argmax}\{\bar{x}_i - \underline{x}_i, i = 1, \dots, n\}$ .
- (b) Let  $\gamma_s$  satisfy  $\gamma_s = (1/2)(\underline{x}_s + \bar{x}_s)$ .
- (c) Let

$$\begin{aligned} X^1 &= \{x \in R^n \mid \underline{x}_i \leq x_i \leq \bar{x}_i, i \neq s, \underline{x}_s \leq x_s \leq \gamma_s\}, \\ X^2 &= \{x \in R^n \mid \underline{x}_i \leq x_i \leq \bar{x}_i, i \neq s, \gamma_s \leq x_s \leq \bar{x}_s\}. \end{aligned} \quad (3.2)$$

Through this branching rule, the rectangle  $X$  is partitioned into two subrectangles  $X^1$  and  $X^2$ .

### 3.2. Algorithm Statement

Based on the former results, the basic steps of the proposed global optimization algorithm are summarized as follows. Let  $\operatorname{LB}(X)$  refer to the optimal value of RLP problem on the rectangle  $X$ .

*Step 1.* Choose  $\epsilon > 0$ . Find an optimal solution  $x^0$  and the optimal value  $\operatorname{LB}(X^0)$  for RLP problem with  $X = X^0$ . Set  $\operatorname{LB}_0 = \operatorname{LB}(X^0)$ . If  $x^0$  is feasible to GGP problem, then update the upper bound  $\operatorname{UB}_0 = \phi_0(x^0)$ . If  $\operatorname{UB}_0 - \operatorname{LB}_0 \leq \epsilon$ , then stop:  $x^0$  is a global  $\epsilon$ -optimal solution of GGP problem. Otherwise, set  $Q_0 = \{X^0\}, F = \emptyset, k = 1$ .

*Step 2.* Set  $\operatorname{LB}_k = \operatorname{LB}_{k-1}$ . Subdivide  $X^{k-1}$  into two rectangles  $X^{k,1}, X^{k,2}$  via the branching rule. Let  $F = F \cup \{X^{k-1}\}$ .

*Step 3.* For each new subrectangle  $X^{k,t}, t = 1, 2$ , utilize the pruning technique of Theorem 2.2 to prune box  $X^{k,t}, t = 1, 2$ . Update the corresponding parameters  $K_i, \underline{y}_j, \bar{y}_j$  ( $i = 1, \dots, n, j = 0, \dots, m$ ). Compute  $\operatorname{LB}(X^{k,t})$  and find an optimal solution  $x^{k,t}$  for RLP problem with  $X = X^{k,t}$ , where  $t = 1, 2$ . If possible, update the upper bound  $\operatorname{UB}_k = \min\{\operatorname{UB}_k, \phi_0(x^{k,t})\}$ , and let  $x^k$  denote the point which satisfies  $\operatorname{UB}_k = \phi_0(x^{k,t})$ .

Step 4. If  $UB_k \leq LB(X^{k,t})$ , then set  $F = F \cup \{X^{k,t}\}$ .

Step 5.  $F = F \cup \{X \in Q_{k-1} \mid UB_k \leq LB(X)\}$ .

Step 6. Set  $Q_k = \{X \mid X \in Q_{k-1} \cup \{X^{k,1}, X^{k,2}\}, X \notin F\}$ .

Step 7. Set  $LB_k = \min\{LB(X) \mid X \in Q_k\}$ , and let  $X^k \in Q_k$  satisfy  $LB_k = LB(X^k)$ . If  $UB_k - LB_k \leq \epsilon$ , then stop:  $x^k$  is a global  $\epsilon$ -optimal solution of GGP problem. Otherwise, set  $k = k + 1$ , and go to Step 2.

### 3.3. Convergence of the Algorithm

The convergence properties of the algorithm are given in the following theorem.

**Theorem 3.1.** (a) *If the algorithm is finite, then upon termination,  $x^k$  is a global  $\epsilon$ -optimal solution of GGP problem.*

(b) *If the algorithm is infinite, then, along any infinite branch of the branch and bound tree, an infinite sequence of iterations will be generated, and any accumulation point of the sequence  $\{x^k\}$  will be a global solution of GGP problem.*

*Proof.* (a) If the algorithm is finite, then it terminates in some stage  $k$ ,  $k \geq 0$ . Upon termination, by the algorithm, it follows that  $UB_k - LB_k \leq \epsilon$ . From Steps 1 and 3, this implies that  $\phi_0(x^k) - LB_k \leq \epsilon$ . Let  $\phi_0^*$  denote the optimal value of GGP problem, then, by Section 2, we know that  $LB_k \leq \phi_0^*$ . Since  $x^k$  is a feasible solution of GGP problem,  $\phi_0(x^k) \geq \phi_0^*$ .

Taken together above, this implies that  $\phi_0^* \leq \phi_0(x^k) \leq LB_k + \epsilon \leq \phi_0^* + \epsilon$ . Therefore,  $\phi_0^* \leq \phi_0(x^k) \leq \phi_0^* + \epsilon$ , and the proof of part (a) is complete.

(b) Let  $D$  denote the feasible region of GGP problem. When the algorithm is infinite, from the construction of the algorithm, we know that  $LB_k$  is a nondecreasing sequence bounded above by  $\min_{x \in D} \phi_0(x)$ , which guarantees the existence of the limit  $LB = \lim_{k \rightarrow \infty} LB_k \leq \min_{x \in D} \phi_0(x)$ . Since  $\{x^k\}$  is contained in a compact set  $X^0$ , there exists one convergent subsequence  $\{x^s\} \subseteq \{x^k\}$ , and suppose  $\lim_{s \rightarrow \infty} x^s = \bar{x}$ . Then, from the proposed algorithm, there exists a decreasing subsequence  $\{X^r\} \subseteq \{X^s\}$ , where  $x^r \in X^r$ ,  $LB_r = LB(X^r) = \phi_0^l(x^r)$ , and  $\lim_{r \rightarrow \infty} X^r = \bar{x}$ . According to Theorem 2.1, we have

$$\lim_{r \rightarrow \infty} \phi_0^l(x^r) = \lim_{r \rightarrow \infty} \phi_0(x^r) = \phi_0(\bar{x}). \quad (3.3)$$

The only remaining problem is to prove that  $\bar{x} \in D$ . Since  $X^0$  is a closed set, it follows that  $\bar{x} \in X^0$ . Assume that  $\bar{x} \notin D$ , then there exists some  $\phi_j(x)$  ( $j = 1, \dots, m$ ) such that  $\phi_j(\bar{x}) > \beta_j$ . Since  $\phi_j^l(x)$  is continuous, by Theorem 2.1, the sequence  $\{\phi_j^l(x^r)\}$  will converge to  $\phi_j(\bar{x})$ . By the definition of convergence, there exists  $\bar{r}$ , such that for any  $r > \bar{r}$ ,  $|\phi_j^l(x^r) - \phi_j(\bar{x})| < \phi_j(\bar{x}) - \beta_j$ . Therefore, for any  $r > \bar{r}$ , we have  $\phi_j^l(x^r) > \beta_j$ , which implies that  $RLP(X^r)$  is infeasible. This contradicts the assumption of  $x^r = x(X^r)$ . Therefore,  $\bar{x} \in D$ , that is,  $\bar{x}$  is a global solution of GGP problem, and the proof of part (b) is complete.  $\square$

#### 4. Numerical Experiment

In this section, we report some numerical results to verify the performance of the proposed algorithm. The test problems are implemented on a Pentium IV (1.66 GHZ) microcomputer. The algorithm is coded in Matlab 7.1 and uses simplex method to solve the relaxation linear programming problems. In our experiments, the convergence tolerance set to  $\epsilon = 1.0e - 3$ .

*Example 4.1* (see [8]). We have the following:

$$\begin{aligned}
 \min \quad & 0.5x_1x_2^{-1} - x_1 - 5x_2^{-1} \\
 \text{s.t.} \quad & 0.01x_2x_3^{-1} + 0.01x_2 + 0.0005x_1x_3 \leq 1, \\
 & x \in X^0 = \{x \mid 70 \leq x_1 \leq 150, 1 \leq x_2 \leq 30, 0.5 \leq x_3 \leq 21\}.
 \end{aligned} \tag{4.1}$$

By using the method in this paper, the optimal solution is (150, 30, 1.3189), the optimal value is  $-147.6667$ , and the number of algorithm iterations is 557. But using the method in [8], the optimal solution is (88.72470, 7.67265, 1.31786), the optimal value is  $-83.249728406$ , and the number of algorithm iterations is 1829.

*Example 4.2* (see [10]). We have the following:

$$\begin{aligned}
 \min \quad & x_1 \\
 \text{s.t.} \quad & 4x_2 - 4x_1^2 \leq 1, \\
 & -x_1 - x_2 \leq -1, \\
 & x \in X^0 = \{x \mid 0.01 \leq x_1 \leq 15, 0.01 \leq x_2 \leq 15\}.
 \end{aligned} \tag{4.2}$$

Through using the method in this paper, the optimal solution (0.5, 0.5) with optimal value 0.5 is found after 16 iterations. But using the method in [10], the optimal solution (0.5, 0.5) with optimal value 0.5 is found after 96 iterations.

*Example 4.3*. We have the following:

$$\begin{aligned}
 \min \quad & -x_1 + x_1^{1.2}x_2^{0.5} - x_2^{0.8} \\
 \text{s.t.} \quad & -6x_1x_2^3 + 8x_2^{0.5} \leq 3, \\
 & 3x_1 - x_2 \leq 3, \\
 & x \in X^0 = \{x \mid 0.5 \leq x_1 \leq 15, 0.5 \leq x_2 \leq 15\}.
 \end{aligned} \tag{4.3}$$

By utilizing the method in this paper, the optimal value  $-1.3501$  is found after 14 iterations at an optimal solution (0.5, 1.5).

*Example 4.4.* We have the following:

$$\begin{aligned}
 \min \quad & 0.3578x_3^{0.1} + 0.8357x_1x_5 \\
 \text{s.t.} \quad & 0.00002584x_3x_5 - 0.00006663x_2x_5 - 0.0000734x_1x_4 \leq 5, \\
 & 0.00085303x_2x_5 + 0.00009395x_1x_4 - 0.00033085x_3x_5 \leq 5, \\
 & 1.3294x_2^{-1}x_5^{-1} - 0.4200x_1x_5^{-1} - 0.30586x_2^{-1}x_3^2x_5^{-1} \leq 5, \\
 & 0.00024186x_2x_5 + 0.00010159x_1x_2 + 0.00007379x_3^2 \leq 5, \\
 & 2.1327x_3^{-1}x_5^{-1} - 0.26680x_1x_5^{-1} - 0.40584x_4x_5^{-1} \leq 5, \\
 & 0.00029955x_3x_5 + 0.00007992x_1x_3 + 0.00012157x_3x_4 \leq 5, \\
 & x \in X^0 = \{x \mid 1 \leq x_1 \leq 60, 1 \leq x_2 \leq 60, 1 \leq x_3 \leq 60, 1 \leq x_4 \leq 60, 1 \leq x_5 \leq 60\}.
 \end{aligned} \tag{4.4}$$

Utilizing the method in this paper, we find the optimal value 1.1935 after 15 iterations at an optimal solution  $x^* = (1.0, 30.2734, 1.0, 17.4640, 1.0)$ .

In the future work, we will do more numerical experiments to test the performance of the proposed algorithm.

## Acknowledgment

This paper is supported by the National Natural Science Foundation of China (60974082), and the Fundamental Research Funds for the Central Universities (K50510700004).

## References

- [1] R. Horst and H. Tuy, *Global Optimization*, Springer, Berlin, Germany, 2nd edition, 1993.
- [2] C. A. Floudas and V. Visweswaran, "Quadratic optimization," in *Handbook of Global Optimization*, vol. 2 of *Nonconvex Optimization and Its Applications*, pp. 217–269, Kluwer Academic, Dodrecht, The Netherlands, 1995.
- [3] 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.
- [4] 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.
- [5] 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, Ser. B, pp. 155–181, 1997.
- [6] Y. Wang, K. Zhang, and P. Shen, "A new type of condensation curvilinear path algorithm for unconstrained generalized geometric programming," *Mathematical and Computer Modelling*, vol. 35, no. 11-12, pp. 1209–1219, 2002.
- [7] 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.
- [8] 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.
- [9] P. Shen, "Linearization method of global optimization for generalized geometric programming," *Applied Mathematics and Computation*, vol. 162, no. 1, pp. 353–370, 2005.
- [10] 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.