Journal of Global Optimization (2005) 33: 1–13
DOI 10.1007/s10898-005-2098-3

Treating Free Variables in Generalized Geometric Global Optimization Programs

HAN-LIN LI[1] and JUNG-FA TSAI[2]
[1]*Institute of Information Management, National Chiao Tung University, No. 1001, Ta Hsueh Road, Hsinchu 300, Taiwan*
[2]*Department of Business Management, National Taipei University of Technology, No. 1, Sec. 3,Chung Hsiao E. Road, Taipei 10608, Taiwan*
(e-mail: jftsai@ntut.edu.tw)

(Received 14 January 2003; accepted 16 January 2005)

Abstract. Generalized geometric programming (GGP) problems occur frequently in engineering design and management. Recently, some exponential-based decomposition methods [Maranas and Floudas, 1997, *Computers and Chemical Engineering* 21(4), 351–370; Floudas et al., 1999, *Handbook of Test Problems in Local and Global Optimization*, Kluwer Academic Publishers, Boston, pp. 85–105; Floudas, 2000 *Deterministic Global Optimizaion: Theory, Methods and Application*, Kluwer Academic Publishers, Boston, pp. 257–306] have been developed for GGP problems. These methods can only handle problems with positive variables, and are incapable of solving more general GGP problems. This study proposes a technique for treating free (i.e., positive, zero or negative) variables in GGP problems. Computationally effective convexification rules are also provided for signomial terms with three variables.

Key words: Generalized geometric programming, Global optimization

1. Introduction

Generalized Geometric Programming (GGP) Problems are frequent in various fields, such as engineering design, location-allocation, chemical process, and management problems. Many theoretical and algorithmic contributions of GGP have also been proposed. The developments of global optimization methods for GGP problems focus mainly on deterministic and heuristic approaches. Horst and Pardalos (1995) and Pardalos and Romeijn (2002) provided an impressive overview of these approaches. Recently, Maranas and Floudas (1997), Floudas et al. (1999) and Floudas (2000) (these three methods are called Floudas' methods in this study) developed methods for solving GGP problems to obtain a global optimum. However, their methods are only applicable to GGP problems with positive variables. This study proposes a technique for improving these methods such that they can also handle GGP problems containing free (i.e., positive, zero or negative) variables.

The mathematical formulation of a GGP problem with free variables is expressed as follows:

GGP:

$$\begin{aligned}
&\text{Minimize} \quad Z(X)=\sum_{p=1}^{T_0} c_p z_p \\
&\text{subject to} \quad \sum_{q=1}^{T_k} h_{kq} z_{kq} \leqslant l_k, \quad k=1,\ldots,K, \\
&z_p = x_1^{\alpha_{p1}} x_2^{\alpha_{p2}} \cdots x_n^{\alpha_{pn}}, \quad p=1,\ldots,T_0, && (1)\\
&z_{kq} = x_1^{\beta_{kq1}} x_2^{\beta_{kq2}} \cdots x_n^{\beta_{kqn}}, \quad k=1,\ldots,K, \quad q=1,\ldots,T_k, && (2)\\
&X=(x_1,\ldots,x_m,x_{m+1},\ldots,x_n), \quad \underline{x}_i \leqslant x_i \leqslant \overline{x}_i,
\end{aligned}$$

$$x_i \text{ are positive variables}, \alpha_{pi}, \beta_{kqi} \in \Re, \text{ for } 1 \leqslant i \leqslant m,$$

$$x_i \text{ are free variables}, \alpha_{pi} \text{ and } \beta_{kqi} \text{ are integers, for } m+1 \leqslant i \leqslant n,$$

where $c_p, h_{kq}, l_k \in \Re$, $T_k, k=0,\ldots,K$, represent the number of posynomial terms of the objective function and of the constraints, and $\underline{x}_i$ and $\overline{x}_i$ are lower and upper bounds of the continuous variable x_i, respectively. If $\alpha_{pi}, \beta_{kqi} \in \Re$ then x_i should be positive; and if α_{pi} and β_{kqi} are integers then x_i can be free variables. These restrictions are illustrated in Examples of Section 5.

For solving the GGP Problem with x_i, where $\varepsilon \leqslant x_i \leqslant \overline{x}_i$, ε is a small positive number for $i=1,\ldots,n$, Floudas' methods denote $x_i = \mathrm{e}^{y_i}$ and group all monomials with identical sign. They reformulate the GGP problem as the following exponential-based non-linear optimization problem:

P1 (Floudas' Model):

$$\begin{aligned}
&\text{Minimize} \quad G_0(Y)=G_0^+(Y)-G_0^-(Y) \\
&\text{subject to} \quad G_k(Y)=G_k^+(Y)-G_k^-(Y) \leqslant l_k, \quad k=1,\ldots,K, \\
&\ln\varepsilon \leqslant y_i \leqslant \overline{y}_i, \quad \overline{y}_i=\ln\overline{x}_i, \quad i=1,\ldots,n, \\
&G_0^+(Y)=\sum_{C_p>0} c_p \mathrm{e}^{\sum_{i=1}^{n} \alpha_{pi} y_i}, \quad p=1,\ldots,T_0, \\
&G_0^-(Y)=\sum_{C_p<0} c_p e^{\sum_{i=1}^{n} \alpha_{pi} y_i}, \quad p=1,\ldots,T_0, \\
&G_k^+(Y)=\sum_{h_{kq}>0} h_{kq} \mathrm{e}^{\sum_{i=1}^{n} \beta_{kqi} y_i}, \quad k=1,\ldots,K, \quad q=1,\ldots,T_k, \\
&G_k^-(Y)=\sum_{h_{kq}<0} h_{kq} \mathrm{e}^{\sum_{i=1}^{n} \beta_{kqi} y_i}, \quad k=1,\ldots,K, \quad q=1,\ldots,T_k,
\end{aligned}$$

where $Y = (y_1, \ldots, y_n)$ is a decision vector, G_k^+ and G_k^-, $k = 0, \ldots, K$, are positive posynomial functions, and ε is a small positive number.

P1 is an exponential-based decomposition programming problem where both the constraints and the objective are decomposed into the difference of two convex functions. A decomposition program has good properties for finding its global optimum (Horst and Tuy, 1996). A convex relaxation of the decomposition can be computed easily based on the linear lower bound of the concave parts of the objective function and constraints.

Floudas' methods can reach finite ε-convergence to the global minimum by successively refining a convex relaxation of a series of non-linear convex optimization problems. However, the usefulness of their methods is limited by two difficulties as detailed below:

(i) Since Floudas' methods require to replace x_i by e^{y_i}, x_i must be strictly positive. However, x_i may be zero or negative values in some applications. For instance, in a portfolio investment problem, $x_i > 0$, $x_i = 0$, and $x_i < 0$ respectively mean that the ith item of an asset is being treated as buying, without buying and selling, and selling. Furthermore, some variables (e.g., temperature, force, acceleration, speed etc.) of many engineering design problems are allowed to be zero or negative values.

(ii) For certain classes of signomial terms z_p in Equation (1) (or z_{kq} in Equation (2)), it is inefficient to replace each x_{pi} in z_p with $e^{y_{pi}}$ to linearize z_p since this requires the new non-convex constraint "$y_{pi} = \ln x_{pi}$" to be added to the constraint set. In fact, for a z_p where α_{pi} satisfies some conditions, some effective techniques can be developed to convexify z_p.

This study proposes some techniques for overcoming the above difficulties in Floudas' methods. The proposed techniques can be applied to free variables as well as positive variables. Additionally, the proposed techniques can effectively convexify signomial terms with three variables (i.e., z_p and z_{kq} in Equations (1) and (2)).

The rest of this paper is organized as follows. Section 2 formulates some propositions for treating free variables. Subsequently, Section 3 proposes a modified Floudas' model based on the propositions discussed in Section 2. The convexification strategies for signomial terms with three variables are analyzed in Section 4. After that, various numerical examples are demonstrated.

2. Propositions

Consider x_i, $\underline{x}_i < x_i \leqslant \overline{x}_i$, $\underline{x}_i$ can be zero or negative. Denote λ_i and θ_i as two 0-1 variables, as defined below:

(i) $x_i = 0$ if and only if $\lambda_i = 0$.
(ii) $x_i < 0$ if and only if $\lambda_i = 1$ and $\theta_i = 1$.
(iii) $x_i > 0$ if and only if $\lambda_i = 1$ and $\theta_i = 0$.

The above conditions can be represented by a set of linear inequalities as described below:

PROPOSITION 1. *Let* $\underline{x}_i,\ \overline{x}_i \in \Re,\ \underline{x}_i \leqslant x_i \leqslant \overline{x}_i,\ \lambda_i, \theta_i \in \{0, 1\}$, $0 < x_i^0 \leqslant \overline{x}_i$ *then:*

$$x_i = x_i^0 \lambda_i (1 - 2\theta_i) \Leftrightarrow \begin{cases} \text{(i)} & \underline{x}_i \lambda_i \leqslant x_i \leqslant \overline{x}_i \lambda_i \\ \text{(ii)} & \overline{x}_i(\lambda_i - 2\theta_i - 1) + x_i^0 \leqslant x_i \leqslant \overline{x}_i(1 - \lambda_i + \theta_i) + x_i^0 \\ \text{(iii)} & \overline{x}_i(\lambda_i + \theta_i - 2) - x_i^0 \leqslant x_i \leqslant \overline{x}_i(3 - \lambda_i - 2\theta_i) - x_i^0 \end{cases}$$

Proof. Let us first prove that constraints (i), (ii), and (iii) imply $x_i = x_i^0 \lambda_i (1 - 2\theta_i)$; as stated below:

If (i) is activated then $\lambda_i = 0$, which results in $x_i = 0$ and $x_i^0 \lambda_i (1 - 2\theta_i) = 0$.
If (ii) is activated then $\lambda_i - 2\theta_i - 1 = 1 - \lambda_i + \theta_i$ (means $\lambda_i = 1$ and $\theta_i = 0$), which implies $x_i = x_i^0$ and $x_i^0 \lambda_i (1 - 2\theta_i) = x_i^0$.
If (iii) is activated then $\lambda_i + \theta_i - 2 = 3 - \lambda_i - 2\theta_i$ (means $\lambda_i = 1$ and $\theta_i = 1$), which implies $x_i = -x_i^0$ and $x_i^0 \lambda_i (1 - 2\theta_i) = -x_i^0$.

The next step is to prove that the equality $x_i = x_i^0 \lambda_i (1 - 2\theta_i)$ is fully converted into constraints (i), (ii), and (iii). If $\lambda_i = 0$ then the equality means $x_i = 0$, which is the same as in (i), and does not violate with (ii) and (iii). If $\lambda_i = 1$ and $\theta_i = 0$ then the equality is the same as in (ii) and does not violate with (i) and (iii). Similarly, if $\lambda_i = 1$ and $\theta_i = 1$ then the equality is the same as in (iii). The above demonstrates that the equality $x_i = x_i^0 \lambda_i (1 - 2\theta_i)$ is equivalent to constraints (i), (ii), and (iii). □

Remark 1. Consider the variable x_i in Proposition 1, if $\underline{x}_i = 0$ then x_i is expressed as $x_i = x_i^0 \lambda_i$, $0 < x_i^0 \leqslant \overline{x}_i$, and

(i) $0 \leqslant x_i \leqslant \overline{x}_i \lambda_i$.
(ii) $\overline{x}_i(\lambda_i - 1) + x_i^0 \leqslant x_i \leqslant \overline{x}_i(1 - \lambda_i) + x_i^0$.

Now denote z and z^0 as below:

$$z = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n} \quad \text{and} \quad z^0 = (x_1^0)^{\alpha_1} (x_2^0)^{\alpha_2} \cdots (x_n^0)^{\alpha_n},$$

where x_i^0 are positive variables.

From Proposition 1, it is clear that

$$z = z^0 \lambda_1 \lambda_2 \cdots \lambda_n (1 - 2\theta_1)^{\alpha_1} (1 - 2\theta_2)^{\alpha_2} \cdots (1 - 2\theta_n)^{\alpha_n}, \quad \lambda_i, \theta_i \in \{0, 1\}. \quad (3)$$

Now denote J as a set containing all i (where the lower bound of x_i is negative), and denote I as a set of all i (where $i \in J$ and α_i is odd), J and I are expressed as

$$J = \{i \mid i = 1, 2, \ldots, n, \ \text{where } \underline{x}_i \text{ is negative}\}$$

and

$$I = \{i \mid i = 1, 2, \ldots, n, \quad i \in J \text{and } \alpha_i \text{ is odd}\}.$$

Consider the following propositions and remarks.

PROPOSITION 2. *Expression* (3) *can be rewritten as*

$$z = z^0 \lambda_1 \lambda_2 \cdots \lambda_n \prod_{i \in I} (1 - 2\theta_i)^{\alpha_i}, \quad \lambda_i, \theta_i \in \{0, 1\}. \tag{4}$$

Proof.

(i) If $i \notin J$ then $x_i \geqslant 0$, therefore $\theta_i = 0$ and $(1 - 2\theta_i)^{\alpha_i} = 1$.
(ii) If $i \in J$ but $i \notin I$ then α_i is even, therefore $(1 - 2\theta_i)^{\alpha_i} = 1$.

□

Remark 2. If $\sum_{i \in I} \theta_i$ is an odd value, then $\prod_{i \in I} (1 - 2\theta_i)^{\alpha_i} = -1$. If $\sum_{i \in I} \theta_i$ is an even value, then $\prod_{i \in I} (1 - 2\theta_i)^{\alpha_i} = 1$.

PROPOSITION 3. *Let* $\theta \in \{0, 1\}$ *and* t *an integer variable, where* $\theta = \sum_{i \in I} \theta_i - 2t$ *and* $0 \leqslant t \leqslant \frac{1}{2} \left(\sum_{i \in I} \theta_i + 1 \right)$, *then* $\prod_{i \in I} (1 - 2\theta_i)^{\alpha_i} = 1 - 2\theta$.

Proof. If $\sum_{i \in I} \theta_i$ is odd then θ is 1, and if $\sum_{i \in I} \theta_i$ is even then θ is 0. This proves the proposition. □

For instance, if $\sum_{i \in I} \theta_i = 4$ then $\theta = 4 - 2t$ for $0 \leqslant t \leqslant 2$, which forces $t = 2$ and $\theta = 0$. If $\sum_{i \in I} \theta_i = 5$ then $\theta = 5 - 2t$ for $0 \leqslant t \leqslant 3$, which requires $t = 2$ and $\theta = 1$.

Remark 3. The product term $\lambda_1 \lambda_2 \cdots \lambda_n$ can be replaced by a $0-1$ variable λ, where $\lambda \leqslant \lambda_i$ for $i = 1, 2, \ldots, n$, and $\lambda \geqslant \sum_{i=1}^{n} \lambda_i - n + 1$.

By referring to Proposition 3 and Remark 3, Equation (4) becomes

$$z = z^0 \lambda (1 - 2\theta). \tag{5}$$

Now z can be reformulated as the linear function of z^0, λ and θ, as described below:

PROPOSITION 4. *Let* $\lambda, \theta \in \{0, 1\}$, $\bar{z}$ *is the upper bound of* z, *and* $\bar{z} > 0$, *then:*

$$z = z^0\lambda(1-2\theta) \Leftrightarrow \begin{cases} z = w - 2r \\ 0 \leqslant w \leqslant \bar{z}\lambda \\ z^0 + \bar{z}(\lambda - 1) \leqslant w \leqslant z^0 + \bar{z}(1-\lambda) \\ 0 \leqslant r \leqslant \bar{z}\theta \\ r \leqslant \bar{z}\lambda \\ z^0 + \bar{z}(\lambda + \theta - 2) \leqslant r \leqslant z^0 + \bar{z}(2 - \lambda - \theta) \end{cases}$$

Proof.
If $\lambda = 0$ then $w = 0$ and $r = 0$ based on (i) and (iv), which results in $z = w - 2r = z^0\lambda(1-2\theta) = 0$.
If $\lambda = 1$ and $\theta = 0$ then $w = z^0$ (from (ii)) and $r = 0$ (from (iii)), which results in $z = z^0 = z^0\lambda(1-2\theta)$.
If $\lambda = 1$ and $\theta = 1$ then $w = z^0$ (from (ii)) and $r = z^0$ (from (v)), which results in $z = -z^0 = z^0\lambda(1-2\theta)$.
The above cases support that $z = z^0\lambda(1-2\theta) = w - 2r$. □

Remark 4. For $z = x_1^{\alpha_1}x_2^{\alpha_2}\cdots x_n^{\alpha_n}$, where $x_i \geqslant 0$, Equation (5) becomes $z = z^0\lambda$. Following Proposition 4, z is simplified as $z = w$ where (i) $0 \leqslant w \leqslant \bar{z}\lambda$ and (ii) $z^0 + \bar{z}(\lambda - 1) \leqslant w \leqslant z^0 + \bar{z}(1-\lambda)$.

3. Modified Floudas' Model

This section merges the technique in Section 2 into the Floudas' Model of P1. Denote $y_{pi} = \ln x_{pi}^0$ and $y_{kqi} = \ln x_{kqi}^0$, where $x_{pi}^0 \geqslant \varepsilon$ and $x_{kqi}^0 \geqslant \varepsilon$ for ε to be a value indicating machine precision (as $\varepsilon = 10^{-8}$). The product terms $z_p^0 = x_1^{\alpha_{p1}}x_2^{\alpha_{p2}}\cdots x_n^{\alpha_{pn}}$ and $z_{kq}^0 = x_1^{\beta_{kq1}}x_2^{\beta_{kq2}}\cdots x_n^{\beta_{kqn}}$ are expressed as $z_p^0 = \mathrm{e}^{\sum_{i=1}^n \alpha_{pi}y_i}$ and $z_{kq}^0 = \mathrm{e}^{\sum_{i=1}^n \alpha_{kqi}y_i}$. Based on the above discussion, the GGP model can be converted into the following linear mixed 0–1 program:
P2 (Modified Floudas' Model):

$$\text{Minimize} \quad \sum_{p=1}^{T_0} c_p(w_p - 2r_p)$$

$$\text{subject to} \quad \sum_{q=1}^{T_k} h_{kq}(w_{kq} - 2r_{kq}) \leqslant l_k \quad \text{for } k = 1, \ldots, K,$$

$$
\begin{array}{lll}
\text{(a1)} & 0 \leqslant w_p \leqslant \bar{z}\lambda_p, & p=1,\ldots,T_0,\\
\text{(a2)} & w_p - \mathrm{e}^{\sum \alpha_{pi} y_{pi}} + \bar{z}(1-\lambda_p) \leqslant 0, & p=1,\ldots,T_0,\\
\text{(a3)} & -w_p + \mathrm{e}^{\sum \alpha_{pi} y_{pi}} + \bar{z}(\lambda_p - 1) \leqslant 0, & p=1,\ldots,T_0,\\
\text{(a4)} & 0 \leqslant r_p \leqslant \bar{z}\theta_p, & p=1,\ldots,T_0,\\
\text{(a5)} & r_p \leqslant \bar{z}\lambda_p, & p=1,\ldots,T_0,\\
\text{(a6)} & r_p - \mathrm{e}^{\sum \alpha_{pi} y_{pi}} + \bar{z}(2-\lambda_p+\theta_p) \leqslant 0, & p=1,\ldots,T_0,\\
\text{(a7)} & -r_p + \mathrm{e}^{\sum \alpha_{pi} y_{pi}} + \bar{z}(\lambda_p+\theta_p-2) \leqslant 0, & \forall p \in J, \quad p=1,\ldots,T_0,\\
\text{(a8)} & \sum_{i=1}^{n} \lambda_{pi} - n + 1 \leqslant \lambda_p, & p=1,\ldots,T_0,\\
\text{(a9)} & \lambda_p \leqslant \lambda_{pi}, & p=1,\ldots,T_0, \quad i=1,\ldots,n,\\
\text{(a10)} & \theta_p = \sum_{i \in I} \theta_{pi} - 2t_p, & p=1,\ldots,T_0,\\
\text{(a11)} & 0 \leqslant t_p \leqslant \frac{1}{2}(\sum_{i \in I} \theta_{pi} + 1), & p=1,\ldots,T_0,\\
\text{(a12)} & \underline{x}_{pi}\lambda_{pi} \leqslant x_{pi} \leqslant \bar{x}_{pi}\lambda_{pi}, & p=1,\ldots,T_0, \quad i=1,\ldots,n,\\
\text{(a13)} & \bar{x}_{pi}(\lambda_{pi} - 2\theta_{pi} - 1) + \mathrm{e}^{y_{pi}} & \\
 & \quad \leqslant x_{pi} \leqslant (1-\lambda_{pi}+\theta_{pi})\bar{x}_{pi} + \mathrm{e}^{y_{pi}}, & \forall pi \in J, \quad p=1,\ldots,T_0,\\
 & & \quad i=1,\ldots,n,\\
\text{(a14)} & \bar{x}_{pi}(\lambda_{pi} + \theta_{pi} - 2) - \mathrm{e}^{y_{pi}} & \\
 & \quad \leqslant x_{pi} \leqslant \bar{x}_{pi}(3-\lambda_{pi}-2\theta_{pi}) - \mathrm{e}^{y_{pi}}, & \forall pi \in J, \quad p=1,\ldots,T_0,\\
 & & \quad i=1,\ldots,n,\\
\text{(a15)} & \bar{x}_{pi}(\lambda_{pi} - 1) + \mathrm{e}^{y_{pi}} & \\
 & \quad \leqslant x_{pi} \leqslant \bar{x}_{pi}(1-\lambda_{pi}) + \mathrm{e}^{y_{pi}}, & \forall pi \notin J, \quad p=1,\ldots,T_0,\\
 & & \quad i=1,\ldots,n,\\
\text{(a16)} & \ln \varepsilon \leqslant y_{pi} \leqslant \ln \bar{x}_{pi}, & p=1,\ldots,T_0, \quad i=1,\ldots,n,
\end{array}
$$

where λ_p, λ_{pi}, θ_p, θ_{pi} are 0–1 variables, and t_p is an integer variable, for $p=1,\ldots,T_0$. $\bar{z} = \mathrm{Max}\{0, z_p, z_{kq},$ for all p and $kq\}$ is a constant.

(b1)–(b16): the same as in Equations (a1)–(a16) where all subscripts p and pi are changed to kq and kqi, respectively. All λ_{kpi}, θ_{kqi}, λ_{kq}, θ_{kq} are 0-1 variables, and t_{kq} is an integer variable.

Each of the constraints (a2), (a3), (a6), (a7), (a13), (a14), (a15) is the difference between two convex functions. All other constraints in P2 are linear inequalities composed of 0–1 variables and continuous variables. Using the branch-bound algorithm, the P2 program can be computed to find a point sufficiently closed to the global optimum based on the pre-specified precision through the linear lower bounding of the concave parts.

Remark 5. For a general signomial form $z = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ where the number of variables with free lower bounds is $k (k < n)$, the number of extra binary variables used to transform z into some proper forms solvable using Floudas' methods is $2 + 2n$. Moreover, the number of additional constraints in the transformation is $13 + 7n + 2k$.

4. Convexification Strategies for Signomial Terms with Three Variables

P2 provides a model for convexifying general signomial terms in a GGP problem based on an exponential-based decomposition techniques. However, there are more computationally efficient convexification strategies for signomial terms with specific features. To simplify the expression, this study takes a signomial term with three variables as an example illustrating the convexification techniques. Consider the following propositions:

PROPOSITION 5. *A twice-differentiable function* $f(X) = cx_1^{\alpha_1} x_2^{\alpha_2} x_3^{\alpha_3}$ *is a convex function in one of the following conditions.*

(i) $c \geqslant 0$, $\alpha_1, \alpha_2, \alpha_3 \leqslant 0$, *and* α_i *is even for corresponding* $x_i, x_i < 0$, $i = 1, 2, 3$. *(i.e., if* $x_i < 0$, *then* α_i *must be even. Otherwise,* α_i *can be odd or even.)*
(ii) $c < 0$, $0 \leqslant \alpha_1, \alpha_2, \alpha_3 < 1$, $\sum_{i=1}^{3} \alpha_i \leqslant 1$, *and* $x_1, x_2, x_3 \geqslant 0$.
(iii) $c < 0$, $\alpha_1, \alpha_2, \alpha_3 \leqslant 0$, *and odd number of all* α_i *are odd for corresponding* $x_i, x_i < 0, i = 1, 2, 3$. (Referring to Tsai et al. (2002))

Proof. Denote $H(X)$ as the Hessian matrix of $f(X)$, and denote H_i as the ith principal minor of a Hessian matrix $H(X)$ of $f(X)$. The determinant of H_i can be expressed as $\det H_i = (-1)^i (\prod_{j=1}^{i} c\alpha_j x_j^{i\alpha_j - 2})(1 - \sum_{j=1}^{i} \alpha_j)$ for $i = 1, 2, 3$.

(i) Since $c \geqslant 0$, $\alpha_1, \alpha_2, \alpha_3 \leqslant 0$, and α_i is even for corresponding $x_i, x_i < 0 (i = 1, 2, 3)$, then $\det H_1 \geqslant 0$, $\det H_2 \geqslant 0$, and $\det H_3 \geqslant 0$. Hence, $f(X)$ is convex.
(ii) Since $c < 0$, $0 \leqslant \alpha_1, \alpha_2, \alpha_3 < 1$, $\sum_{i=1}^{3} \alpha_i \leqslant 1$, and $x_1, x_2, x_3 \geqslant 0$, then $\det H_1 \geqslant 0$, $\det H_2 \geqslant 0$, and $\det H_3 \geqslant 0$. Hence, $f(X)$ is convex.
(iii) If odd number of α_i are odd for corresponding $x_i, x_i < 0 (i = 1, 2, 3)$, then $x_1^{\alpha_1} x_2^{\alpha_2} x_3^{\alpha_3} < 0$. Since $c < 0$, $\alpha_1, \alpha_2, \alpha_3 \leqslant 0$, and $x_1^{\alpha_1} x_2^{\alpha_2} x_3^{\alpha_3} < 0$, then $\det H_1 \geqslant 0$, $\det H_2 \geqslant 0$, and $\det H_3 \geqslant 0$. Therefore, $f(X)$ is convex. □

For a given signomial term z, if z can be converted into a set of convex terms satisfying Proposition 5, then the whole solution process is more computationally efficient. Under this condition, z does not require exponential-based decomposition.

Remark 6. For a signomial term $z=cx_1^{\alpha_1}x_2^{\alpha_2}x_3^{\alpha_3}$, $c\geqslant 0$, $\alpha_1,\alpha_2,\alpha_3\leqslant 0$, $x_1<0, x_2, x_3>0$, if α_1 is even, then z is convex. Otherwise, z can be expressed as $z=cx_{11}^{\alpha_1}x_2^{\alpha_2}x_3^{\alpha_3}+cx_{12}^{\alpha_1}x_2^{\alpha_2}x_3^{\alpha_3}(\underline{x_1}\leqslant x_{11}\leqslant 0, 0<x_{12}\leqslant\overline{x_1})$ where $cx_{12}^{\alpha_1}x_2^{\alpha_2}x_3^{\alpha_3}$ is convex.

For instance, $z=x_1^{-1}x_2^{-2}x_3^{-1}$ with $-5\leqslant x_1\leqslant 5, 0<x_2,x_3\leqslant 5$ can be expressed as $z=x_{11}^{-1}x_2^{-2}x_3^{-1}+x_{12}^{-1}x_2^{-2}x_3^{-1}(-5\leqslant x_{11}\leqslant 0, 0<x_{12}\leqslant 5)$ where the term $x_{12}^{-1}x_2^{-2}x_3^{-1}$ is a convex term requiring no transformation, and the term $x_{11}^{-1}x_2^{-2}x_3^{-1}$ can be transformed into some proper forms solvable by Floudas' methods.

Remark 7. For a signomial term $z=cx_1^{\alpha_1}x_2^{\alpha_2}x_3^{\alpha_3}$, $c<0$, $\alpha_1,\alpha_2,\alpha_3\leqslant 0$, $x_1<0, x_2, x_3>0$, if α_1 is odd, then z is convex. Otherwise, $z=cx_{11}^{\alpha_1}x_2^{\alpha_2}x_3^{\alpha_3}+cx_{12}^{\alpha_1}x_2^{\alpha_2}x_3^{\alpha_3}(\underline{x_1}\leqslant x_{11}\leqslant 0, 0<x_{12}\leqslant\overline{x_1})$ where $cx_{11}^{\alpha_1}x_2^{\alpha_2}x_3^{\alpha_3}$ is a convex term.

For instance, $z=-x_1^{-1}x_2^{-2}x_3^{-1}$ with $-5\leqslant x_1\leqslant 5, 0<x_2,x_3\leqslant 5$ can be expressed as $z=-x_{11}^{-1}x_2^{-2}x_3^{-1}-x_{12}^{-1}x_2^{-2}x_3^{-1}(-5\leqslant x_{11}\leqslant 0, 0<x_{12}\leqslant 5)$ where $-x_{12}^{-1}x_2^{-2}x_3^{-1}$ is a convex term and needs no transformation, and $-x_{11}^{-1}x_2^{-2}x_3^{-1}$ can be transformed into some proper forms solvable by Floudas' methods.

5. Examples

EXAMPLE 1.

$$\begin{aligned}
&\text{Minimize} && Z(X)=x_1^{2.1}x_2x_3^3+x_1\\
&\text{subject to} && \\
& && -x_1-x_2^2\leqslant -5,\\
& && x_2-x_1+x_3\leqslant 13,\\
& && 0\leqslant x_1\leqslant 3, -2\leqslant x_2\leqslant 3, -2\leqslant x_3\leqslant 3.
\end{aligned}$$

This example contains free variables unable being treated by Floudas' methods. This study introduces non-negative continuous variables w_1, w_2, w_3, w_4, w_5, r_1, r_2, and r_3 as follows:

$$w_1-2r_1=x_1^{2.1}x_2x_3^3,\quad w_2=x_1,\quad w_3=x_2^2,\quad w_4-2r_2=x_2,\quad w_5-2r_3=x_3.$$

Introducing positive variables z_i^0, for $i=1,\ldots,5$, and x_j^0, for $j=1,2,3$, as follows:

$$z_1^0 = e^{2.1y_1+y_{2+}+3y_3}, \quad z_2^0 = e^{y_1}, \quad z_3^0 = e^{2y_2}, \quad z_4^0 = e^{y_2},$$
$$z_5^0 = e^{y_3}, \text{ and } x_j^0 = e^{y_j}, \quad \text{for } j = 1, 2, 3.$$

The upper bounds of z_i and x_j are specified as $\bar{z} = Max\{z_1, z_2, z_3, z_4, z_5\} = 814$ and $\bar{x} = \{\bar{x}_1, \bar{x}_2, \bar{x}_3\} = 3$, respectively.

According to (a1)–(a16) and (b1)–(b16), this example is converted into the following program:

Minimize $(w_1 - 2r_1) + w_2$

subject to

$-w_2 - w_3 \leqslant -5,$

$(w_4 - 2r_2) - w_2 + (w_5 - 2r_3) \leqslant 10,$

(a1) $0 \leqslant w_1 \leqslant \bar{z}\lambda_1, 0 \leqslant w_2 \leqslant \bar{z}\lambda_2, 0 \leqslant w_3 \leqslant \bar{z}\lambda_3, 0 \leqslant w_4 \leqslant \bar{z}\lambda_4,$
$0 \leqslant w_5 \leqslant \bar{z}\lambda_5,$

(a2), (a3) $e^{2.1y_1+y_2+3y_3} + \bar{z}(\lambda_1 - 1) \leqslant w_1 \leqslant e^{2.1y_1+y_2+3y_3} + \bar{z}(1 - \lambda_1),$
$e^{y_1} + \bar{z}(\lambda_2 - 1) \leqslant w_2 \leqslant e^{y_1} + \bar{z}(1 - \lambda_2),$
$e^{2y_2} + \bar{z}(\lambda_3 - 1) \leqslant w_3 \leqslant e^{2y_2} + \bar{z}(1 - \lambda_3),$
$e^{y_2} + \bar{z}(\lambda_4 - 1) \leqslant w_4 \leqslant e^{y_2} + \bar{z}(1 - \lambda_4),$
$e^{y_3} + \bar{z}(\lambda_5 - 1) \leqslant w_5 \leqslant e^{y_3} + \bar{z}(1 - \lambda_5),$

(a4) $0 \leqslant r_1 \leqslant \bar{z}\theta, \quad 0 \leqslant r_2 \leqslant \bar{z}\theta_1, \quad 0 \leqslant r_3 \leqslant \bar{z}\theta_2,$

(a5) $r_1 \leqslant \bar{z}\lambda_1, \quad r_2 \leqslant \bar{z}\lambda_4, \quad r_3 \leqslant \bar{z}\lambda_5,$

(a6), (a7) $e^{2.1y_1+y_2+3y_3} + \bar{z}(\lambda_1 + \theta - 2) \leqslant r_1 \leqslant e^{2.1y_1+y_2+3y_3} + \bar{z}(2 - \lambda_1 - \theta),$
$e^{y_2} + \bar{z}(\lambda_4 + \theta_1 - 2) \leqslant r_2 \leqslant e^{y_2} + \bar{z}(2 - \lambda_4 - \theta_1),$
$e^{y_3} + \bar{z}(\lambda_5 + \theta_2 - 2) \leqslant r_3 \leqslant e^{y_3} + \bar{z}(2 - \lambda_5 - \theta_2),$

(a8) $\lambda_2 + \lambda_4 + \lambda_5 - 3 + 1 \leqslant \lambda_1,$

(a9) $\lambda_1 \leqslant \lambda_2, \lambda_1 \leqslant \lambda_3$ and $\lambda_1 \leqslant \lambda_5,$

(a10) $\theta = \theta_1 + \theta_2 - 2t,$

(a11) $t \leqslant \frac{1}{2}(\theta_1 + \theta_2 + 1),$

(a12) $0 \leqslant w_2 \leqslant 3\lambda_2, \quad -2\lambda_4 \leqslant w_4 - 2r_2 \leqslant 3\lambda_4,$
$-2\lambda_5 \leqslant w_5 - 2r_3 \leqslant 3\lambda_5,$

(a13) $3(\lambda_4 - \theta_1 - 1) + e^{y_2} \leqslant w_4 - 2r_2 \leqslant 3(1 - \lambda_4 + \theta_1) + e^{y_2},$
$3(\lambda_5 - \theta_2 - 1) + e^{y_3} \leqslant w_5 - 2r_3 \leqslant 3(1 - \lambda_5 + \theta_2) + e^{y_3}$

(a14) $3(\lambda_4 - \theta_1 - 2) + e^{y_2} \leqslant w_4 - 2r_2 \leqslant 3(3 - \lambda_4 - 2\theta_1) + e^{y_2},$
$3(\lambda_5 - \theta_2 - 2) + e^{y_3} \leqslant w_5 - 2r_3 \leqslant 3(3 - \lambda_5 - 2\theta_2) + e^{y_3},$

(a15) $3(\lambda_2 - 1) + e^{y_1} \leqslant w_2 \leqslant 3(1 - \lambda_2) + e^{y_1},$
$\ln\varepsilon \leqslant y_1 \leqslant \ln 3, \quad \ln\varepsilon \leqslant y_2 \leqslant \ln 3, \quad \ln\varepsilon \leqslant y_2 \leqslant \ln 3,$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \theta, \theta_1, \theta_2$ are 0–1 variables, $\bar{z} = 814$, and y_1, y_2, y_3 are unrestricted in sign.

By adding eight more binary variables, the above program is solved by Floudas' methods with the global optimal solution $(x_1, x_2, x_3) = (3, -2, 3)$, $(w_1, w_2, w_3, w_4, w_5) = (542.4359, 3, 4, 2, 3)$, $t = 0$, $(r_1, r_2, r_3) = (542.3459, 2, 3)$, $(y_1, y_2, y_3) = (1.0986, 0.6931, 1.0986)$, $(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5) = (1, 1, 1, 1, 1)$, and $(\theta, \theta_1, \theta_2) = (1, 1, 0)$. The objective value is –539.4359.

EXAMPLE 2. Insulated steel tank design (Ryoo and Sahinidis, 1995)

$$\begin{aligned}
\text{Minimize}\quad & 400x_1^{0.9} + 1000 + 22(x_2 - 14.7)^{1.2} + x_4 \\
\text{subject to}\quad & \\
& x_2 = \exp(-3950/(x_3 + 460) + 11.86), \\
& 144(80 - x_3) = x_1 x_4, \\
& 0 \leqslant x_1 \leqslant 15.1, 14.7 \leqslant x_2 \leqslant 94.2, -459.67 \leqslant x_3 \leqslant 80, \quad 0 \leqslant x_4.
\end{aligned}$$

The decision variable x_3, which denotes the temperature of the ammonia inside the tank, may be a negative value. This problem cannot be treated directly by Floudas' methods. After converting the program with the proposed techniques, the program is solved to obtain the global solution $(x_1, x_2, x_3, x_4) = (0, 94.1779, 80, 0)$ with the objective value 5194.87.

EXAMPLE 3.

$$\begin{aligned}
\text{Minimize}\quad & Z(X) = x_1^{-2} x_2^{-0.5} x_3^{-1} + 8x_1^{-1} x_4^2 - 8x_4 \\
\text{subject to}\quad & \\
& x_1 - x_2^{0.5} x_3^{0.5} \leqslant 3, \\
& 2x_1 + x_2 - x_3 + x_4 \leqslant 6, \\
& 1 \leqslant x_1 \leqslant 5, \quad 3 \leqslant x_2 \leqslant 7, \quad 1 \leqslant x_3 \leqslant 10, \quad 1 \leqslant x_4 \leqslant 5.
\end{aligned}$$

This program is a non-convex program with four positive variables. By referring to the proposed convexification rules in Proposition 5, the non-linear terms $x_1^{-2} x_2^{-0.5} x_3^{-1}$ and $-x_2^{0.5} x_3^{0.5}$ are convex, and $8x_1^{-1} x_4^2$ can be transformed into a convex term $8x_1^{-1} z^{-1}$ where $z = x_4^{-2}$. By piecewisely linearizing a single term z, the whole program can be reformulated as a convex program solvable to obtain a global optimum. However, solving this program by Floudas' methods requires piecewisely linearizing four logarithmic terms $\ln x_i (i = 1, 2, 3, 4)$ and a non-convex term $-x_2^{0.5} x_3^{0.5}$. Table 1 lists the computational results of solving the program by Floudas' methods and the proposed method on the same computer with LINGO (2001). Table 1 demonstrates that the proposed method is more computationally efficient than Floudas' methods.

Table 1. Computational comparison of Example 3

	Floudas' methods ($\varepsilon \leqslant 5\%$)	The proposed method
(x_1, x_2, x_3, x_4)	(5, 3.5325, 10, 2.4675)	(5, 3.5, 10, 2.5)
Objective value	−9.9962	−9.9979
CPU time (mm:ss)	02:15	00:03

Table 2. Computational comparison of Example 4

	Floudas' methods ($\varepsilon \leqslant 5\%$)	The proposed method
(x_1, x_2, x_3)	(0.2, 0.8, 1.9)	(0.2, 0.8, 1.9079)
(y_1, y_2, y_3, y_4)	(1, 1, 0, 1)	(1, 1, 0, 1)
Objective value	4.5969	4.5796
CPU time (mm:ss)	07:12	00:02

EXAMPLE 4. Process synthesis MINLP (Ryoo and Sahinidis, 1995)

$$\begin{aligned}
\text{Minimize}\quad & (y_1-1)^2+(y_2-2)^2+(y_3-1)^2 \\
& -\log(y_4+1)+(x_1-1)^2+(x_2-2)^2+(x_3-3)^2 \\
\text{subject to}\quad & \\
& y_1+y_2+y_3+x_1+x_2+x_3 \leqslant 5, \\
& y_3^2+x_1^2+x_2^2+x_3^2 \leqslant 5.5, \\
& y_1+x_1 \leqslant 1.2, \quad y_2+x_2 \leqslant 1.8, \quad y_3+x_3 \leqslant 2.5, \\
& y_4+x_1 \leqslant 1.2, \quad y_2^2+x_2^2 \leqslant 1.64, \\
& y_3^2+x_3^2 \leqslant 4.25, \quad y_2^2+x_3^2 \leqslant 4.64, \\
& 0 \leqslant x_1 \leqslant 1.2, \quad 0 \leqslant x_2 \leqslant 1.8, \quad 0 \leqslant x_3 \leqslant 2.5, \\
& y_i \in \{0, 1\}, i = 1, 2, 3, 4.
\end{aligned}$$

Notably, $-\log(y_4+1)$ of the objective function is a convex term and this program is a convex program solvable by the proposed method directly without any transformation. Table 2 compares the computational time between Floudas' methods and the proposed method.

6. Conclusions

This study proposes a method for improving Floudas' methods to treat free variables in GGP programs. The improvement is achieved by converting the logical relationship among the variables in a product term into a set of linear mixed 0-1 inequalities, which can be merged conveniently into Floudas' methods. This study also develops some useful rules to effectively convexify more general signomial terms in GGP programs.

Acknowledgements

The authors would like to thank the anonymous referees for contributing their valuable comments regarding this paper, and thus significantly improving the quality of this paper.

References

Floudas, C.A., Pardalos, P.M., Adjiman, C.S., Esposito, W.R., Gumus, Z.H., Harding S.T., Klepeis, J.L., Meyer, C.A. and Schweiger, C.A. (1999), *Handbook of Test Problems in Local and Global Optimization*, Kluwer Academic Publisherers, Boston, pp. 85–105.

Floudas, C.A. (2000), *Deterministic Global Optimization: Theory, Methods and Application*, Kluwer Academic Publishers, Boston, pp. 257–306.

Horst, R. and Pardalos, P.M., ed. (1995), *Handbook of Global Optimization*, Kluwer Academic Publishers, Boston.

Horst, R. and Tuy, H. (1996), *Global Optimization: Deterministic Approaches*, Springer-Verlag, Berlin, Germany, pp. 505–543.

LINGO Release 7.0. (2001), LINDO System Inc., Chicago.

Maranas, C.D. and Floudas, C.A. (1997), Global optimization in generalized geometric programming, *Computers and Chemical Engineering* 21(4), 351–370.

Pardalos, P.M. and Romeijn, H.E., ed. (2002), *Handbook of Global Optimization-Volumn 2: Heuristic Approaches*, Kluwer Academic Publishers, Boston.

Ryoo, H.S. and Sahinidis, N.V. (1995), Global optimization of Nonconvex NLPs and MINLPs with applications in process design, *Computers and Chemical Engineering* 19(5), 551–566.

Tsai, J.F., Li, H.L. and Hu, N.Z. (2002), Global optimization for signomial discrete programming problems in engineering design, *Engineering Optimization* 34, 613–622.