J Glob Optim (2010) 46:147–154
DOI 10.1007/s10898-009-9414-2

Convex relaxation for solving posynomial programs

Hao-Chun Lu · Han-Lin Li ·
Chrysanthos E. Gounaris · Christodoulos A. Floudas

Received: 2 February 2008 / Accepted: 3 March 2009 / Published online: 21 March 2009

Abstract Convex underestimation techniques for nonlinear functions are an essential part of global optimization. These techniques usually involve the addition of new variables and constraints. In the case of posynomial functions $x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$, logarithmic transformations (Maranas and Floudas, Comput. Chem. Eng. 21:351–370, 1997) are typically used. This study develops an effective method for finding a tight relaxation of a posynomial function by introducing variables y_j and positive parameters β_j, for all $\alpha_j > 0$, such that $y_j = x_j^{-\beta_j}$. By specifying β_j carefully, we can find a tighter underestimation than the current methods.

Keywords Convex underestimation · Posynomial functions

1 Introduction

Convex underestimation techniques are frequently applied in global optimization algorithms. A good convex underestimator should be as tight as possible and should require a minimal number of additional variables and constraints. Floudas [8,9] present various convex underestimating techniques. Ryoo and Sahinidis [21] studied the use of arithmetic intervals, recursive arithmetic intervals, logarithmic transformations, and exponential transformations for multilinear functions. Liberti and Pantelides [14] proposed a nonlinear continuous and differentiable convex envelope for monomials of odd degree. Pörn et al. [20] presented different convexification strategies for nonconvex optimization problems. Björk et al. [4] studied

H.-C. Lu (✉)
Department of Information Management, College of Management Fu Jen Catholic University, No. 510, Jhongjheng Rd., Sinjhuang, Taipei 242, Taiwan
e-mail: haoclu@gmail.com

H.-L. Li
Institute of Information Management, National Chiao Tung University, Management Building 2, 1001 Ta-Hsueh Road, Hsinchu 300, Taiwan

C. E. Gounaris · C. A. Floudas
Department of Chemical Engineering, Princeton University, Princeton, NJ 08544 ,USA

convexifications for signomial terms, introduced properties of power convex functions and studied quasi-convex convexifications.

Tardella [22] studied the class of functions whose convex envelope on a polyhedron coincides. Meyer and Floudas [19] described the structure of the polyhedral convex envelopes of edge-concave functions over polyhedral domains using geometric arguments and showed the improvements over the classical αBB convex underestimators for box-constrained optimization problems. Caratzoulas and Floudas [5] developed convex underestimators for trigonometric functions. Akrotirianakis and Floudas [1,2] introduced a new class of convex underestimators for twice continuously differentiable nonlinear programs, studied their theoretical properties, and proved that the resulting convex relaxation is improved compared to the α BB one. More recently, Gounaris and Floudas [10,11] presented a piecewise application of the α BB method that produces tighter convex underestimators than the original variants.

Three popular convex underestimation methods include arithmetic intervals (AI) [12], recursive arithmetic intervals (rAI) [12,15,21], and explicit facets (EF) for convex envelopes of trilinear monomials [17,18]. However, these current methods have difficulty in treating a posynomial function. Since the number of linear constraints of convex envelopes for a multilinear function with n variables grows doubly exponentially in n, it is more difficult for AI to treat a posynomial function for $n > 3$ cases. Moreover, applying the rAI scheme to underestimate a multilinear function $x_1x_2 \ldots x_n$ requires use of exponentially many 2^{n-1} linear inequalities. Therefore, the rAI bounding scheme also has difficulty in treating posynomial functions.

EF [17,18] provide the explicit facets of the convex and concave envelopes of trilinear monomials. The explicit facets of the convex envelope are effective in treating general trilinear monomials but the derivation of explicit facets for the convex envelope of general multilinear monomials and signomials remains an open problem. Li et al. [13] proposed a new method for the convex relaxation of posynomial functions. $x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$ via the reciprocal transformation and linear underestimation of the concave terms.

This study proposes a novel method of convex relaxation for posynomial functions $f(\mathbf{X}) = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$. For $\alpha_j > 0$, we introduce variable y_j and positive parameter β_j such that $y_j = x_j^{-\beta_j}$. We denote y_j^U as the linear function of β_j satisfying $y_j \le x_j^{-\beta_j} \le y_j^U$. By specifying β_j carefully so as to minimize the difference between y_j^U and y_j, we underestimate $f(\mathbf{X})$ tightly. The proposed method can obtain very tight results, and this is demonstrated with a number of examples drawn from the literature.

This paper is organized as follows. Section 2 develops the convex underestimators for posynomial functions. Section 3 specifies the value of β_j, while the numerical examples are given in Sect. 4.

2 Convex underestimator of a posynomial function

This section presents a new method to develop a convex underestimator for a twice-differentiable posynomial function $f(\mathbf{X}) = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$, where $\mathbf{X} = (x_1, \ldots, x_n)$, $0 < \underline{x}_i \le x_i \le \bar{x}_i$, $\alpha_i \in \Re$ for $i = 1, 2, \ldots, n$, and $c \in \Re$. Let $H(\mathbf{X})$ be the Hessian matrix of $f(\mathbf{X})$ and $H_i(\mathbf{X})$ be the ith principal minor of $H(\mathbf{X})$. The determinant of $H(\mathbf{X})$ and $H_i(\mathbf{X})$ can be expressed as

$$\det H(\mathbf{X}) = (-1)^n \left(\prod_{i=1}^{n} \alpha_i x_i^{n\alpha_i - 2}\right)\left(1 - \sum_{i=1}^{n} \alpha_i\right) \tag{1}$$

and

$$\det H_i(\mathbf{X}) = (-1)^k \left(\prod_{i=1}^{k} \alpha_i x_i^{k\alpha_i - 2}\right)\left(\prod_{j=k+1}^{n} x_j^{k\alpha_j}\right)\left(1 - \sum_{i=1}^{k} \alpha_i\right), \quad k = 1, \ldots, n-1. \tag{2}$$

If $x_i \geq 0$ and $\alpha_i < 0 \forall i$, then $\det H(\mathbf{X}) \geq 0$ and $\det H_i(\mathbf{X}) \geq 0$.

The following proposition holds:

Proposition 1 *A twice-differentiable posynomial function*

$$f(X) = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n} \quad \forall x_i > 0 \tag{3}$$

is a convex function when $\alpha_i < 0 \quad \forall i$.

From Proposition 1, we deduce the following rules for a general posynomial function $f(\mathbf{X})$ as in (3).

Rule 1 If $\alpha_i < 0 \forall i$, then $f(\mathbf{X})$ is already a convex function by Proposition 1. No convexification is required.

Rule 2 If $\alpha_j > 0$ for some j, $j \notin I$, $I = \{k | \alpha_k < 0 \, \forall k = 1, \ldots, n\}$, then we convert $f(\mathbf{X})$ into a new function

$$f(\mathbf{X}, \mathbf{Y}) = \prod_{i \in I} x_i^{\alpha_i} \prod_{j \notin I} y_j^{-\frac{\alpha_j}{\beta_j}} \tag{4}$$

where

$$y_j = x_j^{-\beta_j}, \beta_j \quad \text{are constants}, \quad 0 < \beta_j \leq 1. \tag{5}$$

Since $f(\mathbf{X}, \mathbf{Y})$ is a convex function, we only need to relax (5) for all $j \notin I$.

Let us now focus on relaxing the equality (5). Since $\underline{x}_j^{\beta_j} \leq x_j^{\beta_j} \leq \bar{x}_j^{\beta_j}$ and $\frac{1}{\bar{x}_j^{\beta_j}} = \underline{y}_j \leq y_j \leq \bar{y}_j = \frac{1}{\underline{x}_j^{\beta_j}}$, we have:

$$\left(x_j^{\beta_j} - \underline{x}_j^{\beta_j}\right)\left(y_j - \frac{1}{\bar{x}_j^{\beta_j}}\right) \geq 0. \tag{6}$$

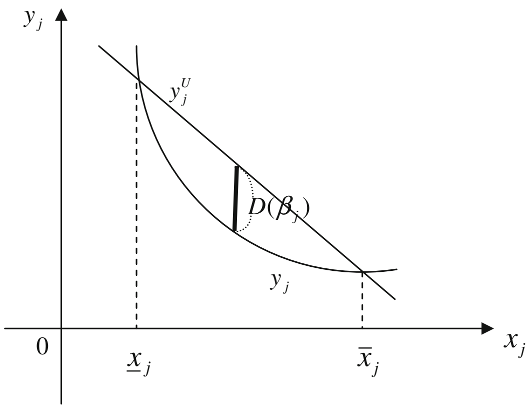
Owing to

$$x_j^{\beta_j} y_j = 1 \quad \text{and} \quad -x_j^{\beta_j} \leq -\left(\underline{x}_j^{\beta_j} + \frac{\bar{x}_j^{\beta_j} - \underline{x}_j^{\beta_j}}{\bar{x}_j - \underline{x}_j}(x_j - \underline{x}_j)\right). \tag{7}$$

It is obvious

$$\begin{aligned} y_j &\leq \frac{1}{\bar{x}_j^{\beta_j}} + \frac{1}{\underline{x}_j^{\beta_j}} - \frac{1}{\bar{x}_j^{\beta_j} \underline{x}_j^{\beta_j}}\left(\underline{x}_j^{\beta_j} + \frac{\bar{x}_j^{\beta_j} - \underline{x}_j^{\beta_j}}{\bar{x}_j - \underline{x}_j}(x_j - \underline{x}_j)\right) \\ &= \underline{x}_j^{-\beta_j} + \frac{\bar{x}_j^{-\beta_j} - \underline{x}_j^{-\beta_j}}{\bar{x}_j - \underline{x}_j}(x_j - \underline{x}_j). \end{aligned} \tag{8}$$

Fig. 1 $D\beta_j$



We then deduce the following proposition:

Proposition 2 *Consider a twice-differentiable non-convex function* $f(\boldsymbol{X}) = x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n}$, $0 < \underline{x}_i \le x_i \le \bar{x}_i \,\forall i$, *where* $\underline{x}_i$ *and* $\bar{x}_i$ *are respectively the lower and upper bounds of* x_i. *Let* $I = \{k|\alpha_k < 0\,\forall k = 1, \ldots, n\}$. *The lower bound of* $f(\boldsymbol{X})$ *can be obtained by solving the following convex problem.*

$$\begin{aligned} &\mathop{\mathbf{Min}}_{\mathrm{x,y}} \prod_{i\in I} x_i^{\alpha_i} \prod_{j\notin I} y_j^{-\frac{\alpha_j}{\beta_j}} \\ &\mathbf{s.t.}\;\; y_j \le y_j^U, \quad j \notin I, \end{aligned} \tag{9}$$

$$\begin{aligned} &y_j^U = \underline{x}_j^{-\beta_j} + \frac{\bar{x}_j^{-\beta_j} - \underline{x}_j^{-\beta_j}}{\bar{x}_j - \underline{x}_j}(x_j - \underline{x}_j), \\ &\textit{where}\; \bar{x}_j^{-\beta_j} \le y_j \le \underline{x}_j^{-\beta_j}, \quad \textit{and} \quad 0 < \beta_j \le 1. \end{aligned} \tag{10}$$

Remark 1 A smaller β_j in Proposition 2 means that y_j^U is tighter to y_j, as presented in Fig. 1.

The value of β_j affects the tightness of y_j^U to y_j. However, given the restriction of computational accuracy, β_j cannot be chosen arbitrarily close to zero. The next selection computes the lowest possible value of β_j so that there exists a computationally distinguishable distance between the two functions, thus preventing the round-off error in the computer's floating computation.

3 Selection of $\boldsymbol{\beta_j}$

The selection of β_j is the same as solving the following optimization program:

Min β_j.
s.t. $y_j^U - y_j \ge \varepsilon$, where ε is the accuracy of the computer.

The maximal value of $\frac{1}{\bar{x}_j^{\beta_j}} + \frac{1}{\underline{x}_j^{\beta_j}} - \frac{1}{\bar{x}_j^{\beta_j}\underline{x}_j^{\beta_j}} y_j^{-1} - y_j$ is $\left(\underline{x}_j^{-0.5\beta_j} - \bar{x}_j^{-0.5\beta_j}\right)^2$, appearing at $y_j = \underline{x}_j^{-0.5\beta_j}\bar{x}_j^{-0.5\beta_j}$ where $\frac{\partial\left(\frac{1}{\bar{x}_j^{\beta_j}} + \frac{1}{\underline{x}_j^{\beta_j}} - \frac{1}{\bar{x}_j^{\beta_j}\underline{x}_j^{\beta_j}} y_j^{-1} - y_j\right)}{\partial y_j} = 0$. Therefore, β_j needs to satisfy:

$$(\underline{x}_j^{-0.5\beta_j} - \bar{x}_j^{-0.5\beta_j})^2 \geq \varepsilon. \tag{11}$$

Let $g_1(\beta_j) = \underline{x}_j^{-0.5\beta_j}$ as a continuous function of β_j. The Taylor expansion of $g_1(\beta_j)$ is $g_1(\beta_j) = \sum_{n=0}^{\infty} \frac{g_1^{(n)}(\beta_j^0)}{n!}(\beta_j - \beta_j^0)^n$. Choosing $\beta_j^0 = 0$, we have

$$g_1(\beta_j) = g_1(0) + g_1'(0)\beta_j + \frac{g_1''(0)}{2!}\beta_j^2 + \cdots + \frac{g_1^{(n)}(0)}{n!}\beta_j^n + \cdots = 1 + \sum_{n=1}^{\infty} \frac{\beta_j^n}{n!}(-0.5\ln \underline{x}_j)^n$$

Similarly, let $g_2(\beta_j) = \bar{x}_j^{-0.5\beta_j}$ that leads to $g_2(\beta_j) = 1 + \sum_{n=1}^{\infty} \frac{\beta_j^n}{n!}(-0.5\ln \bar{x}_j)^n$.

We then have $\left(\underline{x}_j^{-0.5\beta_j} - \bar{x}_j^{-0.5\beta_j}\right)^2 = \left(g_1(\beta_j) - g_2(\beta_j)\right)^2$

$$= \left(\sum_{n=1}^{\infty} \frac{\beta_j^n}{n!}\left((-0.5\ln \underline{x}_j)^n - (-0.5\ln \bar{x}_j)^n\right)\right)^2 = \left(\sum_{n=1}^{\infty} A_n\right)^2, \tag{12}$$

where we have defined $A_n = \frac{\beta_j^n}{n!}((-0.5\ln \underline{x}_j)^n - (-0.5\ln \bar{x}_j)^n)$ for all $n = 1, 2, \ldots, \infty$.

It is clear that $A_1, A_3, A_5, \ldots$ are always positive. If $\ln(\bar{x}_j\underline{x}_j) \leq 0$ then $A_2, A_4, A_6, \ldots$ are also non-negative, otherwise $A_2, A_4, A_6, \ldots$ are negative.

For instance, consider A_2:

$$A_2 = \frac{\beta_j^2}{2}\left(0.25(\ln \underline{x}_j)^2 - 0.25(\ln \bar{x}_j)^2\right) = -0.125\beta_j^2\left(\ln(\bar{x}_j\underline{x}_j) \cdot \ln \frac{\bar{x}_j}{\underline{x}_j}\right) \tag{13}$$

It is obvious that $A_2 \geq 0$ if and only if $\ln(\bar{x}_j\underline{x}_j) \leq 0$.

Proposition 3 *The β_j parameter in Proposition 2 can be selected as follows.*

(i) *If* $\ln(\bar{x}_j\underline{x}_j) \leq 0$, *then*

$$\beta_j \geq \frac{2\sqrt{\varepsilon}}{\ln \bar{x}_j/\underline{x}_j}, \tag{14}$$

where: ε is the accuracy of the computer.

(ii) *If* $\ln(\bar{x}_j\underline{x}_j) > 0$, *then*

$$\beta_j \geq \frac{4\ln \bar{x}_j/\underline{x}_j - \sqrt{G}}{2\ln \bar{x}_j/\underline{x}_j \cdot \ln(\bar{x}_j\underline{x}_j)}, \tag{15}$$

where:

$$G = 16\ln \bar{x}_j/\underline{x}_j\left(\ln \bar{x}_j/\underline{x}_j - 2\sqrt{\varepsilon}\ln(\bar{x}_j\underline{x}_j)\right) \tag{16}$$

and ε should satisfy:

$$\text{“computer accuracy”} \leq \varepsilon \leq \left(\frac{\ln \bar{x}_j/\underline{x}_j}{2\ln(\bar{x}_j\underline{x}_j)}\right)^2. \tag{17}$$

Proof Case 1 $\ln(\bar{x}_j \underline{x}_j) \le 0$

In this case, $A_k > 0\, \forall k$ and $\left(\sum_{k=1}^{\infty} A_k\right)^2 \ge A_1^2 = \left(0.5\beta_j \ln \bar{x}_j/\underline{x}_j\right)^2$. For (11) to hold, it suffices to have $\left(0.5\beta_j \ln \bar{x}_j/\underline{x}_j\right)^2 \ge \varepsilon$, which results in (14).

Case 2 $\ln(\bar{x}_j \underline{x}_j) > 0$

In this case, $A_2, A_4, A_6, \ldots$ are negative. If we choose β_j so as to satisfy

$$A_1 + A_2 \ge \sqrt{\varepsilon}, \tag{18}$$

then it will also hold that $A_{2k-1} + A_{2k} \ge 0$, $k = 2, 3, \ldots \infty$. Therefore, in such a case we would have

$$\left(\underline{x}_j^{-0.5\beta_j} - \bar{x}_j^{-0.5\beta_j}\right)^2 = \left(A_1 + A_2 + \sum_{k=2}^{\infty}(A_{2k-1} + A_{2k})\right)^2 \ge \varepsilon. \tag{19}$$

We now focus on examining the required conditions for the β_j parameters, so that Eq. 18 holds:

$$\begin{aligned}
(19) \Leftrightarrow\ & \beta_j(0.5\ln\bar{x}_j - 0.5\ln\underline{x}_j) + 0.5\beta_j^2(0.25\ln^2\underline{x}_j - 0.25\ln^2\bar{x}_j) \ge \sqrt{\varepsilon} \\
\Leftrightarrow\ & -(\ln\bar{x}_j - \ln\underline{x}_j)(\ln\bar{x}_j + \ln\underline{x}_j)\beta_j^2 + 4\beta_j(\ln\bar{x}_j - \ln\underline{x}_j) - 8\sqrt{\varepsilon} \ge 0 \\
\Leftrightarrow\ & \ln\bar{x}_j/\underline{x}_j \cdot \ln(\bar{x}_j\underline{x}_j)\beta_j^2 - 4\ln\bar{x}_j/\underline{x}_j\beta_j + 8\sqrt{\varepsilon} \le 0.
\end{aligned}$$

Denoting G as in (16), we get $\left(\beta_j - \frac{4\ln\bar{x}_j/\underline{x}_j - \sqrt{G}}{2\ln\bar{x}_j/\underline{x}_j\cdot\ln(\bar{x}_j\underline{x}_j)}\right)\left(\beta_j - \frac{4\ln\bar{x}_j/\underline{x}_j + \sqrt{G}}{2\ln\bar{x}_j/\underline{x}_j\cdot\ln(\bar{x}_j\underline{x}_j)}\right) \le 0$, which results to $\frac{4\ln\bar{x}_j/\underline{x}_j - \sqrt{G}}{2\ln\bar{x}_j/\underline{x}_j\cdot\ln(\bar{x}_j\underline{x}_j)} \le \beta_j \le \frac{4\ln\bar{x}_j/\underline{x}_j + \sqrt{G}}{2\ln\bar{x}_j/\underline{x}_j\cdot\ln(\bar{x}_j\underline{x}_j)}$.

Since G need to be non-negative, the computation accuracy ε we use should satisfy (17). □

4 Numerical examples

Two examples are presented so as to demonstrate the tightness of the proposed convex underestimation technique and compare it to other methods such as the root node lower bound of BARON [23], exponential transformations (ET) [16,20], and reciprocal transformations (RT) [13]. The numerical examples were coded in the GAMS v21.7 [3] environment. The ET, RT and proposed method relaxations were solved using the CONOPT3 solver [7], while the BARON (root node) results were obtained both with BARON version 7.2.5, as well as through the version maintained in the NEOS Server for Optimization [6].

Example 1 Find the underestimation of the following function: $f(\mathbf{X}) = x_1x_2x_3x_4x_5 - x_2^{0.5}x_4^{0.5} - 3x_1 - x_5$, $1 \le x_1, x_2, x_3, x_4, x_5 \le 100$

Let $x_i = y_i^{-\frac{1}{\beta}}$ for $i = 1, 2, \ldots, 5$. A convex relaxation is formulated as follows:

$$\begin{aligned}
&\textbf{Min}\ f(\mathbf{X}, Y) = y_1^{-\frac{1}{\beta}} y_2^{-\frac{1}{\beta}} y_3^{-\frac{1}{\beta}} y_4^{-\frac{1}{\beta}} y_5^{-\frac{1}{\beta}} - x_2^{0.5}x_4^{0.5} - 3x_1 - x_5. \\
&\textbf{s.t.}\quad y_i \le 1 + \frac{100^{-\beta}-1}{100-1}(x_j - 1), \quad i = 1, 2, \ldots, 5,
\end{aligned}$$

where $1 \le x_i \le 100$, $100^{-\beta} \le y_i \le 1$, for $i = 1, 2, \ldots, 5$.

The β parameter is selected as follows:

Table 1 Comparisons of lower bound results for examples

Scheme	BARON (root node)	ET	RT	Proposed method	Global Optimum
Example 1	−499	−209.220	−317.076	−209.477	−202
Example 2	−49.4716	−43.9794	−48.8309	−43.9812	−39.7601

We have $\ln(1 \cdot 100) > 0$, therefore, case (ii) of Proposition 3 applies.
We select $\varepsilon = 10^{-6} < \left(\frac{\ln \bar{x}_j/\underline{x}_j}{2\ln(\bar{x}_j\underline{x}_j)}\right)^2 = 0.25$.
From Eqs. 20 and 21, we calculate

$$\beta_j \geq \frac{4\ln 100/1 - \sqrt{G}}{2\ln 100/1 \cdot \ln(100 \cdot 1)} \approx \frac{4\ln 100 - \sqrt{338.6428361}}{2\ln 100 \cdot \ln 100} \approx 0.00434512$$

Solving this convex program with GAMS/CONOPT, we obtain the solution $(x_1, x_2, x_4, x_5, y_1, y_2, y_3, y_4, y_5) = (90.483591, 1, 1, 1, 0.98209321, 1, 1, 1, 1)$, which corresponds to a lower bound of −209.477. The maximal number of required additional linear constraints is 5. Table 1 lists the results from BARON root node, ET, RT (corresponding to $\beta = 1$), proposed method, and known solution. It shows that the lower bound of the proposed method is comparable to ET and much tighter than the other two approaches.

Example 2 $f(\mathbf{X}) = x_1^{-2}x_2^{-1.5}x_3^{1.2}x_4^{3} - 3x_3^{0.5} + x_2 - 4x_4, \quad 0.01 \leq x_1, \quad x_2, x_3, x_4 \leq 10.$
Let $x_i = y_i^{-\frac{1}{\beta}}, i = 3, 4$. Solving following program:

$$\begin{aligned} &\textbf{Min} \quad f(\mathbf{X},\mathbf{Y}) = x_1^{-2}x_2^{-1.5}y_3^{-\frac{1.2}{\beta}}y_4^{-\frac{3}{\beta}} - 3x_3^{0.5} + x_2 - 4x_4 \\ &\textbf{s.t.} \quad y_j \leq 0.01^{-\beta} + \tfrac{10^{-\beta}-0.01^{-\beta}}{10-0.01}(x_j - 0.01), \quad j = 3, 4 \\ &\text{where} \quad 10^{-\beta} \leq y_j \leq 0.01^{-\beta}, \quad j = 3, 4, \\ &\qquad 0.01 \leq x_i \leq 10, \quad i = 1, 2, 3, 4. \end{aligned}$$

The β parameter is selected as follows:
We have $\ln(0.01 \cdot 10) \leq 0$, therefore case (i) of proposition 3 applies.
We select $\varepsilon = 10^{-6}$ and use Eq. 14 to calculate $\beta_j \geq \frac{2\sqrt{\varepsilon}}{\ln 10/0.01} \approx 0.00028953$.
Solving the convex relaxation with GAMS/CONOPT, we obtain $(x_1, x_2, x_3, x_4, y_3, y_4) = (10, 1.31724, 4.23897, 10, 1.00049, 0.99934)$, which corresponds to a lower bound of −43.9812. The comparison results are shown in Table 1.

5 Conclusion

This study integrates the convexification techniques and the bounding schemes to construct a convex lower bound for a posynomial program. By specifying properly the value of the β_j parameters, the convex relaxations produced provide tight bounds to the global optimum of the posynomial program. Comparing with other underestimation/relaxation techniques, the proposed method produces underestimators of comparable or much better tightness.

Acknowledgments The authors thank the area editor, the anonymous associate editor, and anonymous referees for providing insightful comments that significantly improved this paper. This research has been supported by Taiwan NSC 97-2218-E-030-005-.

References

1. Akrotirianakis, I.G., Floudas, C.A.: Computational experience with a new class of convex underestimators: box constrained NLP problems. J. Glob. Optim. **29**, 249–264 (2004a). doi:10.1023/B:JOGO.0000044768.75992.10
2. Akrotirianakis, I.G., Floudas, C.A.: A new class of improved convex underestimators for twice continuously differentiable constrained NLPs. J. Glob. Optim. **30**, 367–390 (2004b). doi:10.1007/s10898-004-6455-4
3. Brooke, A., Kendrick, D., Meeraus, A., Raman, R.: GAMS: a users guide. GAMS Development Corporation, Washington, DC (2005)
4. Björk, K.J., Lindberg, P.O., Westerlund, T.: Some convexifications in global optimization of problems containing signomial terms. Comput. Chem. Eng. **27**, 669–679 (2003). doi:10.1016/S0098-1354(02)00254-5
5. Caratzoulas, S., Floudas, C.A.: A trigonometric convex underestimator for the base functions in Fourier space. J. Optim. Theory Appl. **124**, 339–362 (2005). doi:10.1007/s10957-004-0940-2
6. Czyzyk, J., Mesnier, M., More, J.: The NEOS server. IEEE J. Comput. Sci. Eng. **5**, 68–75 (1998). doi:10.1109/99.714603
7. Drud, A.S.: CONOPT: a system for large-scale nonlinear optimization. Reference manual for CONOPT subroutine library. ARKI Consulting and Development A/S, Bagsvaerd, Denmark (1996)
8. Floudas, C.A.: Global optimization in design and control of chemical process systems. J. Process Control **20**, 125–134 (2000). doi:10.1016/S0959-1524(99)00019-0
9. Floudas, C.A., Akrotirianakis, I.G., Caratzoulas, S., Meyer, C.A.: Kallrath, J.: Global optimization in the 21st century: advances and challenges. Comput. Chem. Eng. **29**, 1185–1202 (2005). doi:10.1016/j.compchemeng.2005.02.006
10. Gounaris, C.E., Floudas, C.A.: Tight convex underestimators for C2-continuous problems: I. Univariate functions. J. Glob. Optim. **42**, 51–67 (2008)
11. Gounaris, C.E., Floudas, C.A.: Tight convex underestimators for C2-continuous problems: II. Multivariate functions. J. Glob. Optim. **42**, 69–89 (2008)
12. Hamed, A.S.E.: Calculation of bounds on variables and underestimating convex functions for nonconvex functions. Ph.D thesis. The George Washington University (1991)
13. Li, H.L., Tsai, J.F., Floudas, C.A.: Convex Underestimation for Posynomial Functions of Positive Variables. Optim. Lett. **2**, 333–340 (2008)
14. Liberti, L., Pantelides, C.C.: Convex envelops of monomials of odd degree. J. Glob. Optim. **25**, 157–168 (2003). doi:10.1023/A:1021924706467
15. Maranas, C.D., Floudas, C.A.: Finding all solutions of nonlinearly constrained systems of equations. J. Glob. Optim. **7**, 143–182 (1995). doi:10.1007/BF01097059
16. Maranas, C.D., Floudas, C.A.: Global optimization in generalized geometric programming. Comput. Chem. Eng. **21**, 351–370 (1997). doi:10.1016/S0098-1354(96)00282-7
17. Meyer, C.A., Floudas, C.A.: Trilinear monomials with positive or negative domains: facets of the convex and concave envelopes. In : Floudas, C.A., Pardalos, P.M. (eds.) Frontiers in Global Optimization. Kluwer Academic Publishers, Santorini, Greece (2003)
18. Meyer, C.A., Floudas, C.A.: Trilinear monomials with mixed sign domains: facets of the convex and concave envelopes. J. Glob. Optim. **29**, 125–155 (2004). doi:10.1023/B:JOGO.0000042112.72379.e6
19. Meyer, C.A., Floudas, C.A.: Convex envelopes for edge concave functions. Math. Program. Ser. B. **103**, 207–224 (2005). doi:10.1007/s10107-005-0580-9
20. Pörn, R., Harjunkoski, I., Westerlund, T.: Convexification of different classes of non-convex MINLP problems. Comput. Chem. Eng. **23**, 439–448 (1999). doi:10.1016/S0098-1354(98)00305-6
21. Ryoo, H.S., Sahinidis, N.V.: Analysis of bounds for multilinear functions. J. Glob. Optim. **19**, 403–424 (2001). doi:10.1023/A:1011295715398
22. Tardella, F.: On the existence of polyhedral convex envelopes. In: Floudas, C.A., Pardalos, P.M. (eds.) Frontiers in Global Optimization, pp. 563–573. Kluwer Academic Publishers, Santorini, Greece (2003)
23. Tawarmalani, M., Sahinidis, N.V.: BARON 7.2.5: Global optimization of mixed-integer nonlinear programs, User's manual (2005)