J Glob Optim (2008) 42:51–67
DOI 10.1007/s10898-008-9287-9

Tight convex underestimators for $\mathcal{C}^2$-continuous problems: I. univariate functions

Chrysanthos E. Gounaris · Christodoulos A. Floudas

Received: 14 December 2006 / Accepted: 6 February 2008 / Published online: 13 March 2008

Abstract A novel method for the convex underestimation of univariate functions is presented in this paper. The method is based on a piecewise application of the well-known αBB underestimator, which produces an overall underestimator that is piecewise convex. Subsequently, two algorithms are used to identify the linear segments needed for the construction of its $\mathcal{C}^1$-continuous convex envelope, which is itself a valid convex underestimator of the original function. The resulting convex underestimators are very tight, and their tightness benefits from finer partitioning of the initial domain. It is theoretically proven that there is always some finite level of partitioning for which the method yields the convex envelope of the function of interest. The method was applied on a set of univariate test functions previously presented in the literature, and the results indicate that the method produces convex underestimators of high quality in terms of both lower bound and tightness over the whole domain under consideration.

Keywords Global optimization · Convex underestimation · αBB · Convex envelopes · Univariate functions

1 Introduction

Due to recent theoretical and algorithmic advances, *global optimization* has found an increased number of applications across many branches of engineering and science. For instance, complex problems, like the ones arising in refinery pooling (Meyer and Floudas 2006), azeotropic distillation (Maranas et al. 1996; Harding et al. 1997) and phase and chemical equilibrium (McDonald and Floudas 1994, 1995, 1997), have all been tackled by global optimization approaches. Furthermore, many interesting mathematical problems (e.g., enclosure of all solutions of systems of nonlinear equations (Maranas and Floudas 1995), parameter estimation in nonlinear algebraic models (Esposito and Floudas 1998), bilevel programming problems (Gümüş and Floudas 2001)) can be expressed with global

C. E. Gounaris · C. A. Floudas (✉)
Department of Chemical Engineering, Princeton University, Princeton, NJ 08544-5263, USA
e-mail: floudas@titan.princeton.edu

optimization formulations, something that expands the applicability of the relevant results. The publications by Sherali and Adams (1999), Floudas (2000), Horst and Tuy (2003), Horst et al. (2000), Tawarmalani and Sahinidis (2002a) and Floudas and Pardalos (1995, 2003), as well as the recent review papers by Floudas (2005) and Floudas et al. (2005), provide thorough insight on the current status of the field from both the theoretical and application perspective.

In their effort to locate the global solution, deterministic global optimization algorithms, like the αBB (Maranas and Floudas 1994; Androulakis et al. 1995; Adjiman et al. 1998a,b), employ a branch and bound framework. During this process, convex underestimation techniques are used to formulate relaxed convex problems that can be solved to optimality with the use of local solvers, thus providing valid lower bounds for the original problem. The tightness of the underestimators used is of fundamental importance for the computational performance of these algorithms, since a tighter relaxation can lead to faster fathoming and less nodes of the branch and bound tree to be visited (Floudas 2000).

As a consequence, a lot of research effort has been focused on finding tight convex underestimators, particularly for functions of some special structure. From the pioneering work of McCormick (1976) and Al-Khayyal and Falk (1983) who introduced the convex and concave envelope of the bilinear term, up to more recent results on the trilinear envelope (Meyer and Floudas 2003, 2004), a multitude of underestimators has been proposed in the literature. These include results on univariate monomials of odd degree (Liberti and Pantelides 2003), multilinear functions (Ryoo and Sahinidis 2001), fractional (Maranas and Floudas 1995; Tawarmalani and Sahinidis 2001, 2002b) and trigonometric terms (Caratzoulas and Floudas 2005).

In the case of arbitrary nonconvex functions that do not exhibit an exploitable mathematical structure, the αBB general underestimator (Androulakis et al. 1995; Adjiman and Floudas 1996) can be used:

$$L(x) = f(x) - \sum_{v=1}^{V} \alpha_v (x_v - x_v^L)(x_v^U - x_v) \tag{1}$$

Originally introduced in the work of Maranas and Floudas (1994), this underestimator derives from the function by subtracting a positive quadratic ($\alpha_v \geq 0 \forall v$). Given sufficiently large values of the α_v parameters, all nonconvexities in the original function $f(x)$ can be overpowered, resulting into a convex underestimator $L(x)$ that is valid for the entire domain $[x^L, x^U]$. A number of rigorous methods have been devised in order to select appropriate values for these parameters (Adjiman et al. 1998a; Hertz et al. 1999). Extensive computational testing of the algorithm (Adjiman et al. 1998b) showed that the most efficient of those methods is the one based on the *scaled Gherschgorin* theorem. According to this method, it suffices to select:

$$\alpha_v = \max\left\{0, -\frac{1}{2}\left(\underline{h_{vv}} - \sum_{\substack{u=1\\u\neq v}}^{V} \max\left\{|\underline{h_{vu}}|, |\overline{h_{vu}}|\right\} \frac{(x_u^U - x_u^L)}{(x_v^U - x_v^L)}\right)\right\} \tag{2}$$

where $\underline{h_{vu}}$ and $\overline{h_{vu}}$ are lower and upper bounds of $\partial^2 f/\partial x_v x_u$ that can be calculated by interval analysis.

One could alternatively use a new class of general purpose convex underestimators that has been developed by Akrotirianakis and Floudas (2004a,b). These underestimators are derived in a similar fashion, by subtracting an exponential term from the original function, that is:

$$L_1(x) = f(x) - \sum_{v=1}^{V} \left(1 - e^{\gamma_v (x_v - x_v^L)}\right) \left(1 - e^{\gamma_v (x_v^U - x_v)}\right) \tag{3}$$

An iterative systematic procedure is used to determine the values of the γ_v parameters so as the underestimating function to be convex. The procedure ensures also that the resulting underestimator $L_1(x)$ is tighter than $L(x)$, the one that results from the original method. Floudas and Kreinovich (2007a,b) have in fact shown that these two functional forms (original quadratic and exponential) are the only optimal ones, since they are the only ones to be shift-, sign- and scale-invariant.

Maranas and Floudas (1994) showed that the maximum separation distance between the original function $f(x)$ and the underestimator $L(x)$ of Eq. 1 is a quadratic function of interval length. Because of this, as well as because of potentially less overestimation in the interval extension of the Hessian matrix elements h_{vu}, the underestimator would become tighter with shrinkage of the domain under consideration. This was firstly exploited in the work of Meyer and Floudas (2005), where a piecewise approach was utilized. The method proposed partitioning of the domain into many subdomains and construction of the corresponding αBB underestimator for each one of them. These underestimators, although not valid for the entire domain, are much tighter in their respective subdomains. A linear term is subsequently added to each one of these underestimators and is selected in such a way, so that the combination of all these convex *pieces* results into an overall convex underestimator that is continuous and smooth ($\mathcal{C}^1$-continuity).

In the proposed approach, we construct these αBB underestimators, but, instead of adding linear terms, we identify those supporting line segments that have to be combined with convex parts of the original underestimators so as to form a $\mathcal{C}^1$-continuous convex underestimator that is valid for the overall domain under consideration. One can also consider only the lines corresponding to these linear segments, thus coming up with a piecewise linear underestimator that can easily be incorporated in the NLP relaxation as a set of linear constraints.

The method is presented in detail for the case of univariate functions, where it can be directly applied. Theoretical and algorithmic extensions of the method for application on multivariate functions have also been developed, but these will be discussed in a subsequent paper (Gounaris and Floudas 2008).

2 Theoretical results

Let $f(x)$ be a univariate function that needs to be underestimated in $D = [x^L, x^U]$. We select an integer $N > 1$ and partition the complete domain in N segments of equal length. Thus, the ith subdomain would be defined as $D_i = [x^{i-1}, x^i]$, where: $x^i = x^L + \frac{i}{N}(x^U - x^L)$, $i = 0, 1, \ldots, N$.

For every subdomain D_i, $i = 1, 2, \ldots, N$, we construct the corresponding αBB underestimator:

$$\begin{aligned} P_i(x) &= f(x) - \alpha^i (x - x^{i-1})(x^i - x) \\ \alpha^i &= \max\left\{0, -\tfrac{1}{2}\underline{f''}_{(D_i)}\right\} \end{aligned} \tag{4}$$

where $\underline{f''}_{(D_i)}$ is a lower bound of the second derivative that is valid for the entire subdomain D_i. Note that although an underestimator $P_i(x)$ can be defined outside its respective subdomain, its convexity is only guaranteed for $x \in [x^{i-1}, x^i]$.

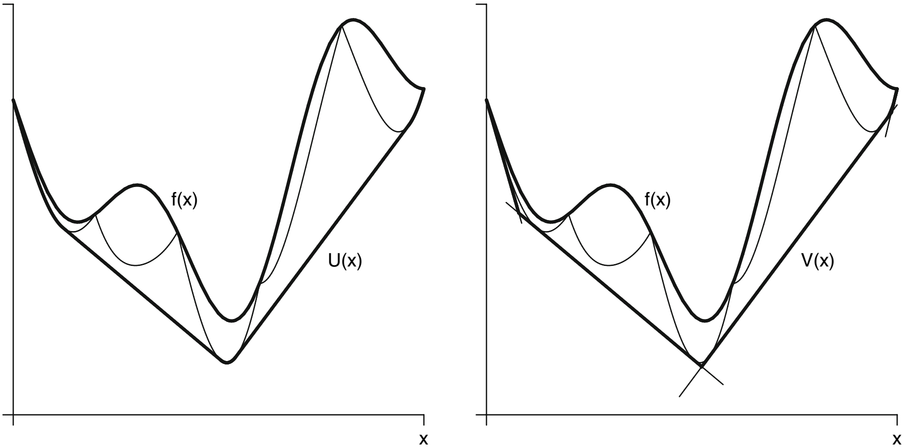

Fig. 1 Function $f(x)$ and underestimators $U(x)$ and $V(x)$

We define $P(x), x \in [x^L, x^U]$ to be the following piecewise function:

$$P(x) = P_i(x), \quad \text{if } x^{i-1} \leq x \leq x^i \tag{5}$$

This function is a piecewise convex valid underestimator of $f(x)$. Since it is not convex, a convexification technique has to be employed. Our proposed technique involves the identification of those supporting line segments that are required for an overall understimator $U(x)$ (depicted in Fig. 1). The technique is based on two algorithms, called *"inner"* and *"outer"*, which are described in detail in the subsequent sections.

The underestimator $U(x)$ consists of the identified linear parts, as well as convex parts of the underestimators $P_i(x)$, therefore it is a $\mathcal{C}^1$-continuous branched function. This might pose some computational complications if the lower bounding (relaxation) problem is to be solved by local optimization solvers that require $\mathcal{C}^2$-continuity. In order to avoid this problem, one can take into account only the lines that correspond to the line segments. According to this alternative, we first identify the linear segments needed for the construction of underestimator $U(x)$, but we consider those as lines defined in $[x^L, x^U]$. Let there be K such lines denoted as $T_k(x), k = 1, 2, \ldots, K$ and arranged in order of ascending slope. If applicable, this set can be augmented with lines that are tangential to P_1 and P_N at the respective domain edges x^L and x^U, according to the following two rules:

$$\text{if } \left.\frac{dP_0}{dx}\right|_{x^L} < slope(T_1) \text{ then add } T_0(x) = P_0(x^L) + \left.\frac{dP_0}{dx}\right|_{x^L}(x - x^L)$$

$$\text{if } \left.\frac{dP_N}{dx}\right|_{x^U} > slope(T_K) \text{ then add } T_{K+1}(x) = P_N(x^U) + \left.\frac{dP_N}{dx}\right|_{x^U}(x - x^U)$$

Note that these two tangents can be included in the set also in the case where $K = 0$, something that would have occurred if function $f(x)$ was already convex. This corresponds to a possible linearization of a convex function.

Each of these lines T_k is a valid underestimator of function $f(x)$ across the whole domain. We define the function $V(x)$ to be the pointwise maximum of all these lines. $V(x)$ is convex, since it is the pointwise maximum of linear functions and it is obviously an underestimator, since it consists of pieces of other underestimators. The underestimator $V(x)$ is also shown in Fig. 1. At the expense of some tightness (in the regions where underestimator $U(x)$ consisted

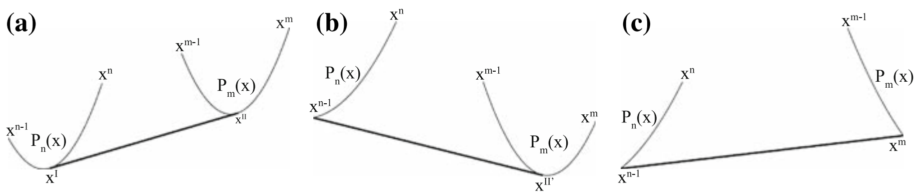

Fig. 2 Examples of supporting line segments

of convex parts), we now have a piecewise linear underestimator $V(x)$ that can be incorporated in the relaxation as a set of linear constraints. The whole lower bounding problem can now be formulated as a linear programming problem (LP).

2.1 Inner algorithm

Given two convex underestimators P_n and P_m, such that $n < m$, the objective of the "inner" algorithm is to identify the supporting line segment that underestimates both pieces in their respective subdomains $[x^{n-1}, x^n]$ and $[x^{m-1}, x^m]$.

This line segment can be completely defined by a point $\{x^{(n)}, P_n(x^{(n)})\}$ on the nth piece, and a point $\{x^{(m)}, P_m(x^{(m)})\}$ on the mth. Note that it can be tangential to both convex pieces, tangential to only one of them, or not tangential to any of the two pieces. Examples of these cases are shown in Fig. 2. The inner algorithm should therefore constitute a procedure that would identify the applicable case and would robustly calculate these two points $x^{(n)}$ and $x^{(m)}$. The line corresponding to the line segment would then be:

$$T(x) = P_n(x^{(n)}) + \frac{P_m(x^{(m)}) - P_n(x^{(n)})}{x^{(m)} - x^{(n)}}(x - x^{(n)}) \tag{6}$$

In the case where $x^{(n)} = x^{(m)}$ (that would simultaneously require $x^{(n)} = x^n,\ x^{(m)} = x^{m-1}$ and $m = n + 1$), the applicable line could be defined as:

$$T(x) = P_n(x^{(n)}) + \left.\frac{dP_n}{dx}\right|_{x^{(n)}} (x - x^{(n)}) \tag{7}$$

that corresponds to the common tangent at the joint of the two consecutive convex pieces. Because of this, the underestimator $V(x)$ of an originally convex function $f(x)$ would correspond to a collection of tangential supports at the points where the domain has been partitioned.

Let us define a number of points:

- Points $x^I \in [x^{n-1}, x^n]$ and $x^{II} \in [x^{m-1}, x^m]$: These points correspond to the case where the line T is tangent to both pieces (Fig. 2a). Point x^I corresponds to piece n and point x^{II} corresponds to piece m. They can be obtained by solving the following system of nonlinear equations:

$$\begin{aligned} \left.\frac{dP_n}{dx}\right|_{x^I} (x^{II} - x^I) + P_n(x^I) - P_m(x^{II}) = 0 \\ \left.\frac{dP_m}{dx}\right|_{x^{II}} (x^{II} - x^I) + P_n(x^I) - P_m(x^{II}) = 0 \end{aligned} \tag{8}$$

- Point $x^{I'} \in [x^{n-1}, x^n]$: This point corresponds to the case where the line T is tangent to only piece n (similar to Fig. 2b). Point $x^{(m)}$ is fixed at either of the two subdomain edges of piece m. Point $x^{I'}$ can be obtained from the solution of the following nonlinear equation:

$$\left.\frac{dP_n}{dx}\right|_{x^{I'}} (x^{fix} - x^{I'}) + P_n(x^{I'}) - P_m(x^{fix}) = 0 \tag{9}$$

where: $x^{fix} = x^{m-1}$ or x^m

- Point $x^{II'} \in [x^{m-1}, x^m]$: This point corresponds to the case where the line T is tangent to only piece m (Fig. 2b). Point $x^{(n)}$ is fixed at either of the two subdomain edges of piece n. Point $x^{II'}$ can be obtained from the solution of the following nonlinear equation:

$$\left.\frac{dP_m}{dx}\right|_{x^{II'}} (x^{fix} - x^{II'}) + P_m(x^{II'}) - P_n(x^{fix}) = 0 \tag{10}$$

where: $x^{fix} = x^{n-1}$ or x^n

- Points $x^{o1} \in [x^{m-1}, x^m]$ and $x^{o2} \in [x^{m-1}, x^m]$: These points are points of piece m, where their slope is equal to the slope of piece n at $x = x^{n-1}$ and $x = x^n$ respectively. They can be obtained from the solution of the following nonlinear equations:

$$\left.\frac{dP_m}{dx}\right|_{x^{o1}} - \left.\frac{dP_n}{dx}\right|_{x^{n-1}} = 0 \tag{11}$$

$$\left.\frac{dP_m}{dx}\right|_{x^{o2}} - \left.\frac{dP_n}{dx}\right|_{x^{n}} = 0 \tag{12}$$

- Points $x^{o3} \in [x^{n-1}, x^n]$ and $x^{o4} \in [x^{n-1}, x^n]$: These points are points of piece n, where their slope is equal to the slope of piece m at $x = x^{m-1}$ and $x = x^m$ respectively. They can be obtained from the solution of the following nonlinear equations:

$$\left.\frac{dP_n}{dx}\right|_{x^{o3}} - \left.\frac{dP_m}{dx}\right|_{x^{m-1}} = 0 \tag{13}$$

$$\left.\frac{dP_n}{dx}\right|_{x^{o4}} - \left.\frac{dP_m}{dx}\right|_{x^{m}} = 0 \tag{14}$$

Let us also define the following slopes:

$$(Q_1) = \left.\frac{dP_n}{dx}\right|_{x^{n-1}} \quad (Q_2) = \left.\frac{dP_n}{dx}\right|_{x^{n}} \quad (Q_3) = \left.\frac{dP_m}{dx}\right|_{x^{m-1}} \quad (Q_4) = \left.\frac{dP_m}{dx}\right|_{x^{m}}$$

$$(O_1) = \frac{P_m(x^{o1}) - P_n(x^{n-1})}{x^{o1} - x^{n-1}} \qquad (R) = \frac{P_m(x^m) - P_n(x^n)}{x^m - x^n}$$

$$(O_2) = \frac{P_m(x^{o2}) - P_n(x^{n})}{x^{o2} - x^{n}} \qquad (L) = \frac{P_m(x^{m-1}) - P_n(x^{n-1})}{x^{m-1} - x^{n-1}}$$

$$(O_3) = \frac{P_m(x^{m-1}) - P_n(x^{o3})}{x^{m-1} - x^{o3}} \qquad (E) = \frac{P_m(x^m) - P_n(x^{n-1})}{x^m - x^{n-1}}$$

$$(O_4) = \frac{P_m(x^{m}) - P_n(x^{o4})}{x^{m} - x^{o4}} \qquad (I) = \frac{P_m(x^{m-1}) - P_n(x^{n})}{x^{m-1} - x^{n}} \quad (\text{if } m \neq n+1)$$

$$(I) = \frac{\left.\frac{dP_n}{dx}\right|_{x^n} + \left.\frac{dP_m}{dx}\right|_{x^{m-1}}}{2} \quad (\text{ if } m = n+1)$$

Due to convexity of the pieces, it always holds that $(Q_1) \leq (Q_2)$ and $(Q_3) \leq (Q_4)$. Every possible value combination of these four slopes can be described with one and only one of the following six cases:

$$\begin{array}{ll} \textit{Case A}: (Q_1) \leq (Q_2) \leq (Q_3) \leq (Q_4) & \textit{Case D}: (Q_3) < (Q_1) \leq (Q_2) \leq (Q_4) \\ \textit{Case B}: (Q_1) \leq (Q_3) < (Q_2) \leq (Q_4) & \textit{Case E}: (Q_3) < (Q_1) \leq (Q_4) < (Q_2) \\ \textit{Case C}: (Q_1) \leq (Q_3) \leq (Q_4) < (Q_2) & \textit{Case F}: (Q_3) \leq (Q_4) < (Q_1) \leq (Q_2) \end{array}$$

Depending on which case applies, the procedure would be as follows:

Case A:

if	$(R) > (Q_4)$,	then $return(x^n, x^m)$
elseif	$(I) > (Q_3)$,	then $return(x^n, x^{II'})$
elseif	$(I) \geq (Q_2)$,	then $return(x^n, x^{m-1})$
elseif	$(L) > (Q_1)$,	then $return(x^{I'}, x^{m-1})$
else		$return(x^{n-1}, x^{m-1})$

Case B:

if	$(R) \geq (Q_4)$,	then $return(x^n, x^m)$
elseif	$(O_2) > (Q_2)$,	then $return(x^n, x^{II'})$
elseif	$(O_3) \geq (Q_3)$,	then $return(x^I, x^{II})$
elseif	$(L) > (Q_1)$,	then $return(x^{I'}, x^{m-1})$
else		$return(x^{n-1}, x^{m-1})$

Case C:

if	$(R) \geq (Q_2)$,	then $return(x^n, x^m)$
elseif	$(O_4) > (Q_4)$,	then $return(x^{I'}, x^m)$
elseif	$(O_3) \geq (Q_3)$,	then $return(x^I, x^{II})$
elseif	$(L) > (Q_1)$,	then $return(x^{n-1}, x^{II'})$
else		$return(x^{n-1}, x^{m-1})$

Case D:

if	$(R) \geq (Q_4)$,	then $return(x^n, x^m)$
elseif	$(O_2) > (Q_2)$,	then $return(x^n, x^{II'})$
elseif	$(O_1) \geq (Q_1)$,	then $return(x^I, x^{II})$
elseif	$(L) > (Q_3)$,	then $return(x^{n-1}, x^{II'})$
else		$return(x^{n-1}, x^{m-1})$

Case E:

if	$(R) \geq (Q_2)$,	then $return(x^n, x^m)$
elseif	$(O_4) > (Q_4)$,	then $return(x^{I'}, x^m)$
elseif	$(O_1) \geq (Q_1)$,	then $return(x^I, x^{II})$
elseif	$(L) > (Q_3)$,	then $return(x^{n-1}, x^{II'})$
else		$return(x^{n-1}, x^{m-1})$

Case F:

if	$(R) \geq (Q_2)$,	then $return(x^n, x^m)$
elseif	$(E) > (Q_1)$,	then $return(x^{I'}, x^m)$
elseif	$(E) \geq (Q_4)$,	then $return(x^{n-1}, x^m)$
elseif	$(L) > (Q_3)$,	then $return(x^{n-1}, x^{II'})$
else		$return(x^{n-1}, x^{m-1})$

Note that these sequences of slope comparisons ensure that a solution of Eqs. 8–14 will be required only when its existness and uniqueness within the respective subdomains is guaranteed. Therefore, utilization of local techniques, such as *Newton–Raphson*, would suffice in locating the appropriate solution of these nonlinear equations.

Illustrative example
As an illustration, let us consider the example of Fig. 3. For this particular case, we have: $(Q_1) = -2.00$, $(Q_2) = +0.20$, $(Q_3) = +0.25$ and $(Q_4) = +1.00$. Therefore, *Case A* applies. We compute: $(R) = +0.14$. The first comparison is false so we move on to compute: $(I) = -0.34$. The second and third comparisons also fail, and we compute $(L) = -0.67$. The fourth comparison holds true, that is: $(L) > (Q_1)$, therefore the algorithm returns the result that the supporting line segment should be tangential to the nth piece at $x^{(n)} = x^{I'}$, to be identified by solving Eq. 9, while its other end should be fixed at $x^{(m)} = x^{m-1}$. The resulting line is shown in Fig. 3. Note that it is guaranteed that point $x^{I'}$ exists and that it is unique.

2.2 Outer algorithm

Given a piecewise convex underestimator $P(x)$ consisting of a set of sequential convex pieces $P_i(x)$, $i = 1, 2, \ldots, N$, the objective of the "outer" algorithm is to identify those supporting line segments (or lines) that participate in the overall underestimator $U(x)$ (or $V(x)$).

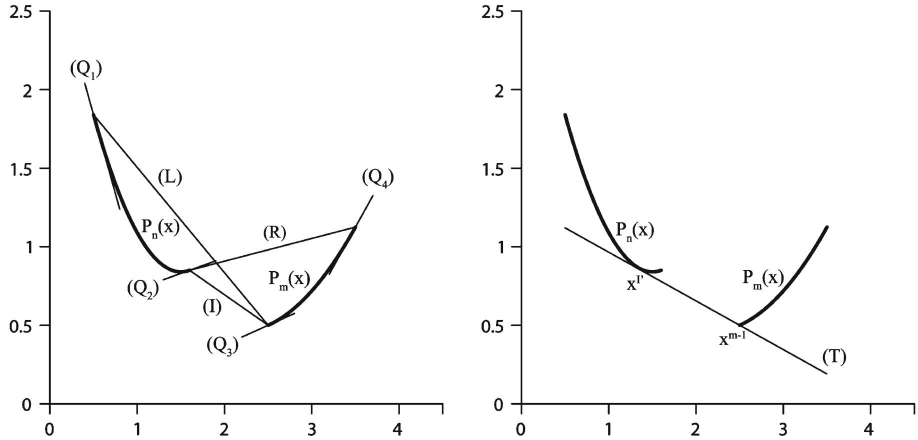

Fig. 3 Illustrative example for "inner" algorithm

Having established the capability of identifying supporting line segments (or lines) for a given pair of convex pieces n and m ("inner" algorithm), the "outer" algorithm will make appropriate calls to this procedure and will use the returning results to identify which of those line segments should be taken into account when constructing an overall convex underestimator.

The algorithm is an inductive incremental procedure that mimics the *Graham–Scan* algorithm for the computation of the convex hull of a set of points (O'Rourke 1998). It can be summarized with the following pseudo-code:

$$
\begin{array}{ll}
 & K = 0,\ n_1 = 1,\ m = 2 \\
(\delta_1): & \text{while } (n_{K+1} \neq N)\ \{ \\
(\delta_2): & \quad T_c = INNER(n_{K+1}, m) \\
 & \quad \text{if } (K = 0) \text{ or } (slope(T_c) > slope(T_K))\ \{ \\
 & \qquad K = K + 1,\ T_K = T_c,\ n_{K+1} = m,\ m = m + 1,\ \text{goto } (\delta_1) \\
 & \quad \text{else} \\
 & \qquad T_K = void,\ K = K - 1,\ \text{goto } (\delta_2) \\
 & \quad \} \\
 & \}
\end{array}
$$

Note that the algorithm does not require that the pieces are connected, that is: $P_{i-1}(x^i) = P_i(x^i)$. However, this holds true for every $i = 1, 2, \ldots, N$ since the underestimator $P(x)$ is continuous (for univariate functions).

Illustrative example

Consider the example of Fig. 4. In this particular example, we have partitioned the domain $[x^L, x^U] = [x^0, x^5]$ into $N = 5$ subdomains, each with length $\Delta x = \frac{x^U - x^L}{5}$, and we have constructed the depicted underestimators $P_i, i \in \{1, 2, 3, 4, 5\}$ according to Eq. 4. The algorithm will examine the set of these underestimators in increasing index i, and will perform a sequence of calls to the "inner" algorithm, each time providing a different pair of these pieces as arguments. We start with an empty stack of linear segments. In the first iteration, the algorithm constructs the linear segment $T_c = T_{12} = INNER(1, 2)$, which is accepted

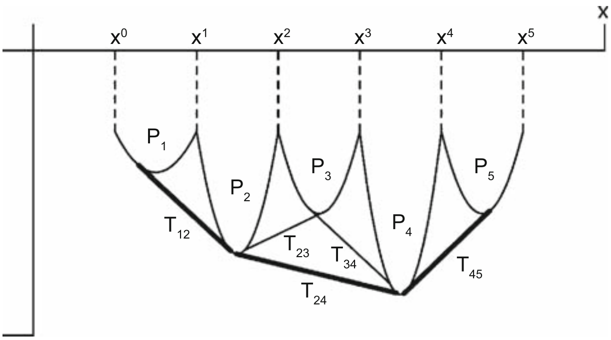

Fig. 4 Illustrative example for "outer" algorithm

in the stack since it is the only one so far ($K = 0$). In the second iteration, the algorithm constructs the linear segment $T_c = T_{23} = INNER(2, 3)$, which is also accepted because $slope(T_{23}) > slope(T_{12})$. In the next iteration, linear segment $T_c = T_{34}$ is constructed, but not accepted because the slope criterion fails, that is $slope(T_{34}) \leq slope(T_{23})$. Furthermore, T_{23} is removed from the stack and the algorithm takes a step backwards to consider piece P_2 once more. $T_c = T_{24}$ is constructed and its slope compared with the slope of the top (and currently only one) member of the stack, T_{12}. The slope criterion, $slope(T_{24}) > slope(T_{12})$, holds true and T_{24} is accepted in the stack. In the next (and last) iteration, T_{45} is constructed and accepted in the stack as well, since $slope(T_{45}) > slope(T_{24})$. Now that the last piece, P_5, has been visited, the loop exits and the algorithm terminates. The final instance of the stack is $T = \{T_{12}, T_{24}, T_{45}\}$, which constitutes the set of linear segments that are required for the construction of an overall convex underestimator $U(x)$.

3 Tightness of proposed underestimator

It is apparent that as the level of partitioning increases, the underestimator $P(x)$ comes closer to the function, and therefore convex underestimator $U(x)$ approaches the convex envelope of $f(x)$. In fact, for any function, there will be a *finite* level of partitioning that suffices for this to happen. Before we provide a rigorous proof to this, we will first present—without proof—a number of relevant lemmas. Their proofs can be found online at http://titan.princeton.edu/.

Lemma 1 *If domain D_j is a subset of domain D_i, its corresponding parameter α^j is less or equal than α^i, that is:*

$$D_j \subseteq D_i \Rightarrow \alpha^j \leq \alpha^i \tag{15}$$

Lemma 2 *Let $x \in [x^L, x^U]$ be a domain of interest and D_i, $i = 1, 2, \ldots, N$ be the subdomains that result from its partition into N pieces. Every subdomain D'_j, $j = 1, 2, \ldots, N'$, that results from a partitioning of $[x^L, x^U]$ in $N' = sN$, $s \in \{1, 2, \ldots\}$ pieces, is a subset of some original subdomain D_i, that is:*

$$N' = sN \Rightarrow \ni i_j : D'_j \subseteq D_{i_j} \quad \forall j \tag{16}$$

Lemma 3 *Let $\tilde{x_k}$, $k = 1, 2, \ldots, K$ be a set of regular numbers, such that $x^L < \tilde{x_k} < x^U$ $\forall k$, that constitute the coordinates of a set of points in the domain (x^L, x^U). There is some level* *of uniform partitioning, N_c, of domain $[x^L, x^U]$, for which every such point coincides with a border between two adjacent subdomains.*

It is important to point out that, in principle, the coordinates of the points of interest, $\tilde{x_k}$, may not be regular numbers. However, the regularity assumption made in Lemma 3 should not be considered limiting in their selection. Every non-regular number is always ϵ-close to a regular number, with this ϵ being arbitrarily small. So, instead of the points of interest, one can use their closest regular numbers without any significant errors in the computations. In any case, these computations are typically taking place under a precision limitation that is imposed by a floating point implementation.

Lemma 4 *Let $\phi^{(f)}(x)$ denote the convex envelope of function $f(x)$, that is the tightest possible convex underestimator of this function across a given domain D, and let $f_1(x)$ and $f_2(x)$ be two functions defined in this domain. The following holds:*

$$\phi^{(f_1)}(x) \leq f_2(x) \leq f_1(x) \quad \forall x \in D \;\Rightarrow\; \phi^{(f_1)}(x) = \phi^{(f_2)}(x) \quad \forall x \in D \tag{17}$$

Lemma 5 *Let $f(x)$ be a strictly concave function defined in $[x^\alpha, x^\beta]$ and let $D_i, i = 1, 2, \ldots, sN$ be the subdomains that result from its partition into sN pieces. There is some finite integer s, $s \geq 1$, for which the underestimator $P(x)$ is above the convex envelope of $f(x)$, that is:*

$$\begin{gathered}\phi^{(f)}(x) \leq P(x) \quad \forall x \in [x^\alpha, x^\beta] \Leftrightarrow \phi^{(f)}(x) \leq P_i(x) \quad \forall x \in D_i \quad \forall i \\ \text{where}: P_i(x) = f(x) - \alpha^i(x - x^{i-1})(x^i - x)\,,\, i = 1, 2, \ldots, sN\end{gathered} \tag{18}$$

Let us now apply the preceding lemmas in order to prove the following theorem:

Theorem 1 *There is some finite partitioning level N, for which the convex underestimator $U(x)$ is the convex envelope of function $f(x)$.*

Proof Every $\mathcal{C}^2$-continuous univariate function consists of *convex* (possibly containing linear) and *strictly concave* segments in an alternating sequence. Let the points of interest $\tilde{x_k}$ be the points at which the second derivative switches between non-negative (convex) and strictly negative (strictly concave) values. According to Lemma 3, there is a partitioning level N_c for which we have a domain change at every one of these points. Thus, the subdomains produced in this manner belong fully to either a convex or a strictly concave part of function f and, because of Lemma 2, the same will hold if we partition into $N_c' = sN_c, s \in \mathbb{N}^*$ subdomains as well.

For each subdomain D_i that corresponds to a convex part, we have $\alpha^i = 0$ and $P_i(x) = f(x)\,\forall x \in D_i$. Therefore, since (by definition) $\phi^{(f)} \leq f$, we also have $\phi^{(f)}(x) \leq P_i(x)\,\forall x \in D_i$. For each subdomain D_i that corresponds to a strictly concave part, Lemma 5 dictates that there is some finite partitioning $s_i N_c$, $s_i \in \mathbb{N}^*$ for which the same holds. Choosing the least common multiple of all these will lead to a partitioning for which $\phi^{(f)}(x) \leq P_i(x)\,\forall x \in D_i\,\forall i$. Thus:

$$\phi^{(f)}(x) \leq P(x) \quad \forall x \in [x^L, x^U] \tag{19}$$

Equation 19 along with the fact that function P is an underestimator of f, $P(x) \leq f(x)\,\forall x \in [x^L, x^U]$, allows for the application of Lemma 4 (with $f_1 \equiv f$ and $f_2 \equiv P$), which leads to $\phi^{(f)}(x) = \phi^{(P)}(x)\ \forall x \in [x^L, x^U]$. But the convex underestimator $U(x)$ is the convex envelope of underestimator $P(x)$ (by construction), that is $U(x) = \phi^{(P)}(x)$. We finally come to the conclusion that:

$$U(x) = \phi^{(f)}(x) \quad \forall x \in [x^L, x^U] \tag{20}$$

and thus Theorem 1 has been proven. □

It has already been mentioned that the underestimator $V(x)$, which is a piecewise linear approximation of underestimator $U(x)$, comes with some expense in tightness. However, as the level of partitioning increases, underestimator $V(x)$ comes closer to underestimator $U(x)$ and their maximum difference can be driven to be arbitrarily small. This is summarized in the following theorem:

Theorem 2 *There is some finite partitioning level N, for which underestimator V is ϵ-close to underestimator U, that is:*

$$\max_{x \in D} \{U(x) - V(x)\} < \epsilon \tag{21}$$

where: $\epsilon > 0$ is an arbitrarily small constant.

Proof The convex underestimator $U(x)$ consists of linear parts (identified by the two algorithms) as well as convex parts of some pieces of the piecewise convex underestimator $P(x)$. By construction, underestimator $V(x)$ coincides with $U(x)$ at these linear parts, which are practically extended so as the overall underestimator to be piecewise linear. The underestimator $V(x)$ might also be augmented with tangents of the first and last piece at $x = x^L$ and $x = x^U$ respectively, as well as with common tangents at the joint of two consecutive pieces, $x = x^i$, in the case where function $f(x)$ is convex along $[x^{i-1}, x^{i+1}]$ and thus coincides with pieces $P_i(x)$ and $P_{i+1}(x)$.

For the parts where the $\mathcal{C}^1$-continuous $U(x)$ is linear, $U(x) - V(x) = 0$ and the theorem holds. We will focus on proving the theorem for the regions where underestimator $U(x)$ is strictly convex.

Note that for every two consecutive lines T_k and T_{k+1} of the underestimator $V(x)$, there will exist a common piece P_{q_k} to which both are tangential. This derives from the way that $V(x)$ has been constructed. Let I_Q be the set of indices of those strictly convex pieces that contribute some part of theirs to the overall convex underestimator $U(x)$, and let $P_q(x)$, $q \in I_Q$ be such a piece. Let also T_1 and T_2 ($slope(T_1) < slope(T_2)$) be the corresponding lines that are tangential to piece $P_q(x)$ at $x = \xi_1$ and $x = \xi_2$ respectively.

The following hold:

$$T_1(x) = P_q(\xi_1) + \left.\frac{dP_q}{dx}\right|_{\xi_1} (x - \xi_1) \tag{22}$$

$$T_2(x) = P_q(\xi_2) + \left.\frac{dP_q}{dx}\right|_{\xi_2} (x - \xi_2) \tag{23}$$

$$x^{q-1} \le \xi_1 < \xi_2 \le x^q$$

For underestimators $V(x)$ and $U(x)$ we have:

$$V(x) = \max\{T_1(x), T_2(x)\} \quad \forall x \in D_q \tag{24}$$

$$U(x) = \begin{cases} T_1(x), & \text{if } x \in [x^{q-1}, \xi_1) \\ P_q(x), & \text{if } x \in [\xi_1, \xi_2] \\ T_2(x), & \text{if } x \in (\xi_2, x^q] \end{cases} = \begin{cases} P_q(x), & \text{if } x \in [\xi_1, \xi_2] \\ V(x), & \text{otherwise} \end{cases} \tag{25}$$

Let δ_q be the *maximum* difference between U and V along the whole subdomain D_q, that is:

$$\delta_q = \max_{x \in D_q} \{U(x) - V(x)\} = \max_{x \in [\xi_1, \xi_2]} \left\{P_q(x) - V(x)\right\} \tag{26}$$

Note that Eq. 24 implies that $V(x) \geq T_1(x)\ \forall x \in D_q$ (and the same holds for T_2 as well). Therefore, Eq. 26 yields:

$$\delta_q \leq \max_{x \in [\xi_1, \xi_2]} \left\{P_q(x) - T_1(x)\right\} \tag{27}$$

Also, note that function $P_q - T_1$ is convex and that $\left.\frac{d(P_q - T_1)}{dx}\right|_{\xi_1} = 0$, therefore:

$$\max_{x \in [\xi_1, \xi_2]} \left\{P_q(x) - T_1(x)\right\} = P_q(\xi_2) - T_1(\xi_2) \tag{28}$$

Combining the above result with Eq. 27, we have:

$$\begin{aligned}
\delta_q &\leq P_q(\xi_2) - T_1(\xi_2) \\
&\overset{\text{Eq. 22}}{=} P_q(\xi_2) - P_q(\xi_1) - \left.\frac{dP_q}{dx}\right|_{\xi_1} (\xi_2 - \xi_1) \\
&\overset{\xi_1 \neq \xi_2}{=} (\xi_2 - \xi_1) \left\{ \frac{P_q(\xi_2) - P_q(\xi_1)}{\xi_2 - \xi_1} - \left.\frac{dP_q}{dx}\right|_{\xi_1} \right\} \\
&\overset{\frac{d^2 P_q}{dx^2} > 0}{<} (\xi_2 - \xi_1) \left\{ \left.\frac{dP_q}{dx}\right|_{\xi_2} - \left.\frac{dP_q}{dx}\right|_{\xi_1} \right\} \\
&\overset{[\xi_1, \xi_2] \subseteq D_q}{\leq} (x^q - x^{q-1}) \left\{ \left.\frac{dP_q}{dx}\right|_{x^q} - \left.\frac{dP_q}{dx}\right|_{x^{q-1}} \right\} \\
&\overset{\text{Eq. 4}}{=} (x^q - x^{q-1}) \left\{ f'(x^q) - f'(x^{q-1}) + 2\alpha^q (x^q - x^{q-1}) \right\} \\
&\overset{D_q \neq \emptyset}{=} (x^q - x^{q-1})^2 \left\{ \frac{f'(x^q) - f'(x^{q-1})}{x^q - x^{q-1}} + 2\alpha^q \right\} \\
&\overset{\text{note}^1}{\leq} (x^q - x^{q-1})^2 \left\{ \overline{f''}^{(D_q)} + 2\alpha^q \right\} \\
&\overset{\text{Eq. 4}}{=} (x^q - x^{q-1})^2 \left\{ \overline{f''}^{(D_q)} + 2 \max\left\{0, -\frac{1}{2} \underline{f''}_{(D_q)}\right\} \right\} \\
&\overset{D_q \subseteq D}{\leq} (x^q - x^{q-1})^2 \left\{ \overline{f''}^{(D)} + 2 \max\left\{0, -\frac{1}{2} \underline{f''}_{(D)}\right\} \right\} \\
&= \left(\frac{x^U - x^L}{N}\right)^2 C
\end{aligned}$$

where: C is constant.

This constant is *strictly positive* because $\overline{f''}^{(D)}$ is strictly positive (otherwise $f(x)$ would be concave and $U(x)$ would be strictly linear, a case that we need not consider in this analysis). Furthermore, it is *finite* since $f(x)$ has to be $\mathcal{C}^2$-continuous defined on a bounded (box)

[1] In any given domain $D = [\alpha, \beta] \neq \emptyset$, the average slope of a function $\psi(x)$, that is $\frac{\psi(\beta) - \psi(\alpha)}{\beta - \alpha}$, is less or equal than the maximum slope, denoted as $\overline{\psi'}^{(D)}$. Here: $\psi \equiv f'$.

domain. Finally, since it is independent of q, we have:

$$\max_{x \in D} \{U(x) - V(x)\} = \max \left\{0, \ \delta_q \quad \forall q \in I_Q\right\} < \left(\frac{x^U - x^L}{N}\right)^2 C \tag{29}$$

We have established a conservative, yet rigorous, upper bound for the maximum separation distance between underestimators $U(x)$ and $V(x)$. Clearly, we can select a finite level of partitioning N for which the right hand side of Eq. 29 is as small as required (arbitrary $\epsilon > 0$) and thus Theorem 2 has been proven. □

4 Computational results

The method was implemented in C and applied on a set of 40 test functions that have been presented in the literature (Casado et al. 2003). Table 1 presents the results for various levels of partitioning. The original αBB method corresponds to no partitioning ($N = 1$).

As it has been described in Sect. 3, the underestimator should become tighter with doubling of the number of subdomains used and all the results are indeed consistent with this. Also note that functions 3,17,20 and 36 are convex in the domain under consideration and therefore underestimator $U(x)$ will be the function $f(x)$ itself for any partitioning level N.

To illustrate the tightness of the underestimators across the whole domain under consideration, we present Fig. 5 that depicts the plots for functions 4,19 and 33. The underestimators presented correspond to partitioning into $N = 24$, 36 and 48 subdomains (increasing tightness).

The “inner” algorithm always involves a pair of pieces and therefore its average run time, $\bar{t}$, would be independent of their total number, N. It would depend on function-related factors, such as the performance of the numerical methods employed to solve Eqs. 8–14. The “outer” algorithm, which specifies how many times the “inner” one will have to be executed, mimics the well-known Graham–Scan algorithm. The latter is an $\mathcal{O}(N log N)$ algorithm that reduces to $\mathcal{O}(N)$ when the entries are presorted, like in our case where the pieces are ordered inherently. Combining the above observations, we expect that the whole method will exhibit linear computational performance, particularly for large N, at which point the average run time is converged around a constant value, that is: $\overline{t_{N+1}} \simeq \overline{t_N}$. All runs were performed on a 3.20 GHz Intel$^{(R)}$ Pentium$^{(R)}$ 4 processor with 1 Gb of RAM. Computations were very fast, in the order of a few hundredths of a second. Runs with partitioning level $N = 512$ averaged a CPU time of 0.05 s, while the more tedious runs with $N = 1{,}024$ averaged 0.09 s, an indication that the computational complexity is indeed linear in the level of partitioning. None of the runs of Table 1 exceeded a CPU time of 0.12 s.

5 Conclusions

In this paper, we presented a convex underestimation method for functions of a single variable. It utilizes the well known αBB underestimator (Maranas and Floudas 1994; Androulakis et al. 1995; Adjiman and Floudas 1996), which is applied in a piecewise fashion. The method was tested on a collection of highly nonlinear problems and the computational results demonstrated that it provides very tight underestimators. Theoretical results on the tightness of these underestimators are presented, along with a proof that there is always some

Table 1 Lower bound results and other information for the complete suite of test functions

#	Function $f(x)$	$[x^L, x^U]$	LM	GM	αBB	$N=2$	4	8	16	32	64	128	256	512	1024	GO f^*
1	$e^{-3x} - \sin^3 x$	[0, 20]	4	1	−450	−113	−28.4	−6.46	−1.549	*	*	*	*	*	*	−1
2	$\sum_{k=1}^{5} -\cos[(k+1)x] + 4$	[0.2, 7.0]	7	1	−516	−126	−32.2	−6.52	*	*	*	*	*	*	*	−1
3	$(x - x^2)^2 + (x-1)^2$	[−10, 10]	1	1	*	*	*	*	*	*	*	*	*	*	*	0
4	$(3x - 1.4)\sin(18x) + 1.7$	[0.2, 7.0]	21	1	−37320	−9320	−2345	−598	−146	−43.0	−18.392	*	*	*	*	−17.58287
5	$2x^2 - \frac{3}{100}e^{-(200(x-0.0675))^2}$	[−10, 10]	1	1	−1E + 12	−8E + 10	−8E + 9	−6E + 8	−5E + 7	−3E + 6	−1E + 5	−4608	−665	−29.2	−0.94	−0.020903
6	$\cos x - \sin(5x) + 1$	[0.2, 7.0]	6	1	−150	−37.1	−9.82	−2.45	−1.004	*	*	*	*	*	*	−0.952897
7	$-x - \sin(3x) + 1.6$	[0.2, 7.0]	4	1	−53.2	−16.7	−8.21	−6.430	*	*	*	*	*	*	*	−6.262872
8	$x + \sin(5x)$	[0.2, 7.0]	7	1	−142	−34.6	−8.97	−1.88	−0.0861	*	*	*	*	*	*	0.077590
9	$-e^{-x}\sin(2\pi x) + 1$	[0.2, 7.0]	7	1	−241	−59.3	−14.2	−2.48	−0.25	*	*	*	*	*	*	0.211315
10	$e^{-x}\sin(2\pi x)$	[0.2, 7.0]	7	1	−242	−60.5	−15.1	−4.19	−0.73	*	*	*	*	*	*	−0.478362
11	$-x + \sin(3x) + 1$	[0.2, 7.0]	5	1	−55.7	−18.1	−8.91	*	*	*	*	*	*	*	*	−5.815675
12	$x\sin x + \sin\left(\frac{10x}{3}\right) + ln x$ $-0.84x + 1.3$	[0.2, 7.0]	4	1	−263	−58.2	−13.0	−7.398	*	*	*	*	*	*	*	−7.047444
13	$\sin x + \sin\left(\frac{10x}{3}\right) + ln x - 0.84x$	[2.7, 7.5]	3	1	−39.7	−11.5	−6.05	−4.632	*	*	*	*	*	*	*	−4.601308
14	$ln(3x) ln(2x) - 0.1$	[0.2, 7.0]	1	1	−528	−82.8	−9.02	*	*	*	*	*	*	*	*	−0.141100
15	$\sum_{k=0}^{5} k\cos[(k+1)x + k] + 12$	[0.2, 7.0]	8	1	−2013	−496	−123	−25.7	−2.86	*	*	*	*	*	*	−0.870885
16	$-\sum_{k=1}^{5} k\sin[(k+1)x + k] + 3$	[0.2, 7.0]	7	1	−2023	−501	−132	−35.1	−11.19	*	*	*	*	*	*	−9.031249
17	$\sin^2(1 + \frac{x-1}{4}) + (\frac{x-1}{4})^2$	[−10, 10]	1	1	*	*	*	*	*	*	*	*	*	*	*	0.475689
18	$\sqrt{x}\sin^2 x$	[0.2, 7.0]	3	2	−75.5	−15.6	−1.93	−0.122	*	*	*	*	*	*	*	0
19	$x^2 - \cos(18x)$	[−5, 5]	29	1	−4026	−1001	−249	−63.0	−16.5	−3.84	−1.459	*	*	*	*	−1
20	e^{x^2}	[−10, 10]	1	1	*	*	*	*	*	*	*	*	*	*	*	1

Table 1 continued

#	Function $f(x)$	$[x^L, x^U]$	LM	GM	αBB	$N = 2$	4	8	16	32	64	128	256	512	1024	GO f^*
21	$\frac{x^2}{20} - \cos x + 2$	[−20, 20]	7	1	−199	−42.7	−9.60	−0.59	0.733	*	*	*	*	*	*	1
22	$\cos x + 2\cos(2x)e^{-x}$	[0.2, 7.0]	2	1	−72.9	−18.6	−3.41	*	*	*	*	*	*	*	*	−0.918397
23	$(x + \sin x)e^{-x^2}$	[−10, 10]	1	1	−2E+5	−55163	−1900	−70.4	−3.50	−1.022	−0.8255	*	*	*	*	−0.824239
24	$2\sin x e^{-x}$	[0.2, 7.0]	2	1	−19.0	−4.38	−0.57	*	*	*	*	*	*	*	*	−0.027864
25	$2\cos x + \cos(2x) + 5$	[0.2, 7.0]	3	2	−30.9	−5.07	2.28	3.335	*	*	*	*	*	*	*	3.5
26	$e^{\sin(3x)}$	[0.2, 7.0]	5	3	−141	−34.8	−8.22	−1.03	*	*	*	*	*	*	*	0.367879
27	$\sin x \cos x - 1.5\sin^2 x + 1.2$	[0.2, 7.0]	3	2	−39.2	−9.87	−2.08	−0.4528	*	*	*	*	*	*	*	−0.451388
28	$\sin x$	[0, 20]	4	3	−50.8	−13.5	−3.84	−1.301	*	*	*	*	*	*	*	−1
29	$2(x-3)^2 - e^{\frac{x}{2}} + 5$	[0.2, 7.0]	1	1	−25.1	−5.21	*	*	*	*	*	*	*	*	*	−0.410315
30	$-e^{\sin(3x)} + 2$	[0.2, 7.0]	4	4	−281	−69.6	−18.1	−3.55	−0.767	*	*	*	*	*	*	−0.718282
31	$-\sum_{i=1}^{10} \frac{1}{[k_i(x-a_i)]^2 + c_i}$	[0, 10]	8	1	−5E+8	−2E+7	−6E+5	−37943	−3838	−211	−25.4	−14.689	*	*	*	−14.59265
32	$\sin(\frac{1}{x})$	[0.02, 1]	6	6	−30312	−7579	−1894	−474	−118	−28.0	−5.83	−1.220	−1.119	−1.094	−1.024	−1
33	$-\sum_{k=1}^{5} k\sin[(k+1)x + k]$	[−10, 10]	20	3	−17496	−4374	−1096	−276	−79.5	−22.0	*	*	*	*	*	−12.03125
34	$\frac{x^2-5x+6}{x^2+1} - 0.5$	[0.2, 7.0]	1	1	−24139	−406	−1.34	*	*	*	*	*	*	*	*	−0.535534
35	$-\sum_{i=1}^{10} \frac{1}{[k_i(x-a_i)]^2 + c_i}$	[0, 10]	7	1	−8E+8	−5E+7	−2E+6	−72451	−3154	−439	−39.2	−15.41	−13.993	−13.923	*	−13.92245
36	$\frac{(x+1)^3}{x^2} - 1.7$	[0.2, 7.0]	1	1	*	*	*	*	*	*	*	*	*	*	*	−0.35
37	$x^4 - 12x^3 + 47x^2 - 60x - 20e^{-x}$	[−1, 7]	1	1	−3716	−606	−79.9	*	*	*	*	*	*	*	*	−32.78126
38	$x^6 - 15x^4 + 27x^2 + 250$	[−4, 4]	2	2	−1478	−342	−86.0	2.60	*	*	*	*	*	*	*	7
39	$x^4 - 10x^3 + 35x^2 - 50x + 24$	[−10, 20]	2	2	−560	−112	−36.1	−9.41	−2.25	−1.268	*	*	*	*	*	−1
40	$24x^4 - 142x^3 + 303x^2 - 276x + 3$	[0, 3]	2	1	−114	−95.2	−90.6	*	*	*	*	*	*	*	*	−89

An asterisk denotes that the bound is equal to the known global optimum, f^*, within six decimal digits of accuracy

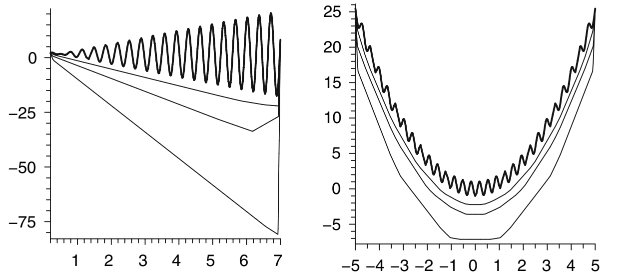

Fig. 5 Functions 4, 19 and 33 with underestimators $V(x)$ for three different partitioning levels ($N = 24$, 36 and 48)

finite level of partitioning for which the method yields the theoretical (yet a priori unknown) convex envelope of the function across the domain of interest.

This observation is very important, since it leads to the conclusion that application of the method with a sufficiently large level of partitioning will close the gap at the root node of the branch and bound tree, thus eliminating the need for any branching at all.

Although the method can be directly applied to univariate functions, efforts were made to extend it into multivariate ones, thus taking advantage of its high quality results. In a subsequent paper (Gounaris and Floudas 2008), we will present how one can utilize the proposed univariate method to construct convex underestimators of functions with a higher number of variables.

Acknowledgment The authors gratefully acknowledge support from the National Science Foundation.

References

Adjiman, C.S., Floudas, C.A.: Rigorous convex underestimators for general twice-differentiable problems. J. Glob. Optim. **9**, 23–40 (1996)

Adjiman, C.S., Dallwig, S., Floudas, C.A., Neumaier, A.: A global optimization method, αBB, for general twice-differentiable constrained NLPs I. Theoretical Advances. Comput. Chem. Eng. **22**, 1137–1158 (1998a)

Adjiman, C.S., Androulakis, I.P., Floudas, C.A.: A global optimization method, αBB, for general twice-differentiable constrained NLPs II. Implementation and Computational Results. Comput. Chem. Eng. **22**, 1159–1179 (1998b)

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 (2004a)

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 (2004b)

Al-Khayyal, F.A., Falk, J.E.: Jointly constrained biconvex programming. Math. Oper. Res. **8**, 273–286 (1983)

Androulakis, I.P., Maranas, C.D., Floudas, C.A.: αBB: a global optimization method for general constrained nonconvex problems. J. Glob. Optim. **7**, 337–363 (1995)

Caratzoulas, S., Floudas, C.A.: Trigonometric convex underestimator for the base functions in Fourier space. J. Optim. Theory Appl. **124**, 339–362 (2005)

Casado, L.G., Martinez, J.A., Garcia, I., Sergeyev, Y.D.: New interval analysis support functions using gradient information in a global minimization algorithm. J. Glob. Optim. **25**, 345–362 (2003)

Esposito, W.R., Floudas, C.A.: Global optimization in parameter estimation of nonlinear algebraic models via the error-in-variables approach. Indus. Eng. Chem. Res. **35**, 1841–1858 (1998)

Floudas, C.A.: Deterministic Global Optimization: Theory, Algorithms and Applications. Kluwer Academic Publishers (2000)

Floudas, C.A., Pardalos, P.M.: Preface. J. Glob. Optim. **1** 113 (1995)

Floudas, C.A., Pardalos, P.M.: Frontiers in Global Optimization. Kluwer Academic Publishers (2003)

Floudas, C.A.: Research challenges, opportunities and synergism in systems engineering and computational biology. AIChE J. **51**, 1872–1884 (2005)
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)
Floudas, C.A., Kreinovich, V.: Towards optimal techniques for solving global optimization problems: symmetry-based approach. In: Torn A., Zilinskas J. (eds.) Models and Algorithms for Global Optimization, pp. 21–42 Springer (2007a)
Floudas, C.A., Kreinovich, V.: On the functional form of convex understimators for twice continuously differentiable functions. Optim. Lett. **1**, 187–192 (2007b)
Gounaris, C.E., Floudas, C.A.: Tight convex underestimators for $\mathcal{C}^2$-continuous functions: II. Multivariate functions. J. Glob. Optim. DOI: 10.1007/s10898-008-9288-8 (2008)
Gümüş, Z.H., Floudas, C.A.: Global optimization of nonlinear bilevel programming problems. J. Glob. Optim. **20**, 1–31 (2001)
Harding, S.T., Maranas, C.D., McDonald, C.M., Floudas, C.A.: Locating all homogeneous azeotropes in multicomponent mixtures. Ind. Eng. Chem. Res. **36**, 160–178 (1997)
Hertz, D., Adjiman, C.S., Floudas, C.A.: Two results on bounding the roots of interval polynomials. Comput. Chem. Eng. **23**, 1333–1339 (1999)
Horst, R., Pardalos, P.M., Thoai, N.V.: Introduction to Global Optimization, 2nd edn. Kluwer Academic Publishers (2000)
Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches, 3rd edn. Springer (2003)
Liberti, L., Pantelides, C.C.: Convex envelopes of monomials of odd degree. J. Glob. Optim. **25**, 157–168 (2003)
Maranas, C.D., Floudas, C.A.: Global minimum potential energy conformations of small molecules. J. Glob. Optim. **4**, 135–170 (1994)
Maranas, C.D., Floudas, C.A.: Finding all solutions of nonlinearly constrained systems of equations. J. Glob. Optim. **7**, 143–182 (1995)
Maranas, C.D., McDonald, C.M., Harding, S.T., Floudas, C.A.: Locating all azeotropes in homogeneous azeotropic systems. Comput. Chem. Eng. **20**, S413–S418 (1996)
McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: Part I—Convex underestimating problems. Math. Program. **10**, 147–175 (1976)
McDonald, C.M., Floudas, C.A.: Decomposition based and branch and bound global optimization approaches for the phase equilibrium problem. J. Glob. Optim. **5**, 205–251 (1994)
McDonald, C.M., Floudas, C.A.: Global optimization for the phase and chemical equilibrium problem: application to the NRTL equation. Comput. Chem. Eng. **19**, 1111–1141 (1995)
McDonald, C.M., Floudas, C.A.: GLOPEQ: a new computational tool for the phase and chemical equilibrium problem. Comput. Chem. Eng. **21**, 1–23 (1997)
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 (2003)
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)
Meyer, C.A., Floudas, C.A.: Convex underestimation of twice continuously differentiable functions by piecewise quadratic perturbation: spline αBB underestimators. J. Glob. Optim. **32**, 221–258 (2005)
Meyer, C.A., Floudas, C.A.: Global optimization of a combinatorially complex generalized pooling problem.. AIChE J. **52**, 1027–1037 (2006)
O'Rourke J.: Computational Geometry in C 2nd edn. Cambridge University Press (1998)
Ryoo, H.S., Sahinidis, N.V.: Analysis of bounds for multilinear functions. J. Glob. Optim. **19**, 403–424 (2001)
Sherali, H.D., Adams, W.P.: Reformulation-Linearization Techniques in Discrete and Continuous Optimization. Kluwer Academic Publishers (1999)
Tawarmalani, M., Sahinidis, N.V.: Semidefinite relaxations of fractional programs via novel convexification techniques. J. Glob. Optim. **20**, 137–158 (2001)
Tawarmalani, M., Sahinidis, N.V.: Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming. Kluwer Academic Publishers (2002a)
Tawarmalani, M., Sahinidis, N.V.: Convex extensions and envelopes of lower semi-continuous functions. Math. Program. **93**, 247–263 (2002b)
Zabinsky, Z.B.: Stochastic Adaptive Search for Global Optimization. Kluwer Academic Publishers (2003)