J Glob Optim (2008) 42:69–89
DOI 10.1007/s10898-008-9288-8

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

Chrysanthos E. Gounaris · Christodoulos A. Floudas

Received: 14 December 2006 / Accepted: 6 February 2008 / Published online: 18 March 2008

Abstract In Part I (Gounaris, C.E., Floudas, C.A.: Tight convex understimators for $\mathcal{C}^2$-continuous functions: I: Univariate functions. J. Global Optim. (2008). doi: 10.007/s10898-008-9287-9), we introduced a novel approach for the underestimation of univariate functions which was based on a piecewise application of the well-known αBB underestimator. The resulting underestimators were shown to be very tight and, in fact, can be driven to coincide with the convex envelopes themselves. An approximation by valid linear supports, resulting in piecewise linear underestimators was also presented. In this paper, we demonstrate how one can make use of the high quality results of the approach in the univariate case so as to extend its applicability to functions with a higher number of variables. This is achieved by proper projections of the multivariate αBB underestimators into select two-dimensional planes. Furthermore, since our method utilizes projections into lower-dimensional spaces, we explore ways to recover some of the information lost in this process. In particular, we apply our method after having transformed the original problem in an orthonormal fashion. This leads to the construction of even tighter underestimators, through the accumulation of additional valid linear cuts in the relaxation.

Keywords Global optimization · Convex underestimation · αBB · Piecewise affine underestimators

1 Introduction

Very important and mathematically complex optimization problems arise frequently in process synthesis and design. The synthesis of separation sequences (e.g., Aggarwal and Floudas 1990; Yeomans and Grossmann 1999), reactor networks (e.g., Kokossis and Floudas 1990, 1994; Achenie and Biegler 1990) and heat exchanger networks (e.g., Ciric and Floudas 1989; Yee and Grossmann 1990) have been traditionally receiving a lot of attention. Other problems of chemical engineering nature, such as finding the location of all azeotropes

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

(Maranas et al. 1996; Harding et al. 1997) and phase and chemical equilibrium (McDonald and Floudas 1994, 1995, 1997), as well as problems arising in computational biology (e.g., protein structure prediction (Floudas et al. 2006) have all been addressed with deterministic global optimization algorithms, and more particularly with the αBB (Maranas and Floudas 1994; Androulakis et al. 1995; Adjiman et al. 1998a,b) framework. The latter has also found application in other interesting problems, such as enclosing all solutions of systems of nonlinear equations (Maranas and Floudas 1995), estimating parameters in nonlinear algebraic models (Esposito and Floudas 1998) and solving bilevel programming problems (Gümüş and Floudas 2001).

The αBB method employs a branch and bound scheme, where sequences of monotonically improving lower and upper bounds for the global solution are generated. The lower bounding is achieved through convex underestimation of the functions under question that are of increasing tightness as the branching process evolves. Therefore, the development of tight convex underestimators of nonconvex functions is of fundamental importance for the computational performance of the method.

There have been many results in the literature for the convex relaxation of problems that exhibit some special structure. These include results on bilinear (McCormick 1976; Al-Khayyal and Falk 1983), trilinear (Meyer and Floudas 2003, 2004), multilinear (Ryoo and Sahinidis 2001) and fractional (Maranas and Floudas 1995; Tawarmalani and Sahinidis 2001; Tawarmalani and Sahinidis 2002b) terms, univariate monomials of odd degree (Liberti and Pantelides 2003), as well as results on trigonometric (Caratzoulas and Floudas 2005) and edge-concave (Tardella 2003; Meyer and Floudas 2005) functions. For a comprehensive study, see the textbooks by Sherali and Adams (1999), Floudas (2000), Horst et al. (2000), Tawarmalani and Sahinidis (2002a), Horst and Tuy (2003), Zabinsky (2003) and Floudas and Pardalos (1995, 2003), as well as the recent review papers by Floudas (2005) and Floudas et al. (2005). When the function structure cannot be exploited specifically, one can use the αBB underestimator (Maranas and Floudas 1994; Androulakis et al. 1995; Adjiman and Floudas 1996) for general nonconvex functions. This underestimator derives from the nonconvex function of interest by subtracting a separable positive quadratic function. Given that this separable quadratic function is sufficiently large, all nonconvexities in the original function can be overpowered, resulting into a convex underestimator. 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. These two classes (quadratic and exponential) are in fact the only optimal ones, as has been shown by Floudas and Kreinovich (2007a,b).

2 Synopsis of part I

In Part I (Gounaris and Floudas 2008), we developed a novel method to determine convex underestimators for univariate functions $f(x)$. The method proposed partitioning of the initial domain in $N > 1$ segments of equal length and construction of the corresponding αBB underestimator (Maranas and Floudas 1994; Androulakis et al. 1995; Adjiman and Floudas 1996; Adjiman et al. 1998a,b) for each subdomain. In order to combine the convex pieces of this piecewise underestimator, the method utilizes two algorithms, called *"inner"* and *"outer"*. The "inner" algorithm is applicable to a given pair of pieces and its purpose is to identify reliably the tightest possible supporting line segment that underestimates both pieces in their respective subdomains. The "outer" algorithm performs a set of suitable calls

to the "inner" one in its effort to identify which of the line segments are required for an overall convex understimator $U(x)$. This resulting underestimator is $\mathcal{C}^1$-continuous since it consists of parts of the identified line segmets and parts of the convex pieces themselves. As an extension of this approach, one could consider only the lines defined by the line segments, resulting in a piecewise linear underestimator $V(x)$ of the original function $f(x)$.

The computational results demonstrated that the method yields underestimators of high quality, in terms of lower bound obtained, as well as in tightness of the underestimators across the whole domain under consideration. It was theoretically proven that, for sufficient level of partitioning, the understimator $U(x)$ can be the convex envelope of $f(x)$ itself, while the underestimator $V(x)$ can be ϵ-close to the convex envelope, with $\epsilon > 0$ being arbitrarily small.

Since these univariate underestimators are very tight, the remaining question is whether we can exploit them so as to construct underestimators of functions in higher dimensions. In this paper, we present some extensions of the method for application on multivariate functions that involve dimension reduction of the problem through proper projections into lower-dimensional spaces. For consistency, the notation used is a generalization of that used in Part I.

3 Method description

Let $f(x)$ be a function of V variables that needs to be underestimated in a box domain $D = [x_1^L, x_1^U] \times \ldots \times [x_V^L, x_V^U]$. We choose integers $N_v > 1,\ v = 1, 2, \ldots, V$ and partition each range $[x_v^L, x_v^U]$ in N_v segments of equal length. Thus, the j^{th} segment of the v^{th} set would be defined as $[x_v^{j-1}, x_v^j]$, where: $x_v^j = x_v^L + \frac{j}{N_v}(x_v^U - x_v^L),\ j = 0, 1, \ldots, N_v$. The complete V-dimensional domain D has now been partitioned into $N = \prod_{v=1}^{V} N_v$ box subdomains of equal measures. Let D_i be such a V-dimensional subdomain. It is uniquely defined by a set of indices $i_v,\ 1 \le i_v \le N_v,\ \forall v = 1, 2, \ldots, V$. Thus, the i^{th} subdomain would be defined as $D_i = [x_1^{i_1-1}, x_1^{i_1}] \times \ldots \times [x_V^{i_V-1}, x_V^{i_V}]$.

For every subdomain $D_i, i = 1, 2, \ldots, N$, we construct the corresponding αBB underestimator (Maranas and Floudas 1994; Androulakis et al. 1995; Adjiman and Floudas 1996; Adjiman et al. 1998a,b):

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

where $\underline{h_{vu}^{(i)}}$ and $\overline{h_{vu}^{(i)}}$ are respectively lower and upper bounds of $\partial^2 f/\partial x_v x_u$ that are 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 $x \in D_i$.

We select variable $w,\ 1 \le w \le V$, which we designate to be the *active* variable, and enumerate all $M_w = N/N_w$ permutations of indices $i_v,\ v \neq w$. Every such permutation $m,\ 1 \le m \le M_w$, corresponds to a subdomain $D_{wm} = \left[x_w^L, x_w^U\right] \times \prod_{\substack{v=1 \\ v\neq w}}^{V} \left[x_v^{i_v-1}, x_v^{i_v}\right]$, which

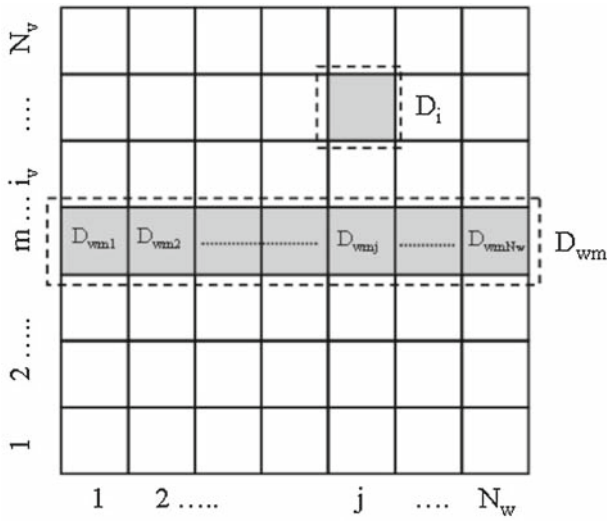

Fig. 1 Partitioning of the multidimensional domain D

can be further divided into N_w subdomains $D_{wmj} = \left[x_w^{j-1}, x_w^j\right] \times \prod_{\substack{v=1 \\ v\neq w}}^{V} \left[x_v^{i_v-1}, x_v^{i_v}\right]$, $j = 1, 2, \ldots, N_w$. These subdomains, depicted in Figure 1, belong to the set of the original subdomains D_i (for $i_w = j$) and therefore each one has an underestimator $P_{wmj}(x)$ associated with it, that is:

$$P_{wmj}(x) = f(x) - \alpha_w^i \left(x_w - x_w^{j-1}\right)\left(x_w^j - x_w\right) - \sum_{\substack{v=1 \\ v\neq w}}^{V} \alpha_v^i (x_v - x_v^{i_v-1})(x_v^{i_v} - x_v) \quad (2)$$

where index i satisfies $D_i = D_{wmj}$ and parameters $\alpha_v^i, v = 1, 2, \ldots, V$ are calculated according to Eq. 1.

For every such subdomain D_{wmj}, $j = 1, 2, \ldots, N_w$, we define the following univariate function:

$$G_{wmj}(x_w) = \min_{\substack{x_v \\ \forall v\neq w}} P_{wmj}(x), \quad x_w^{j-1} \leq x_w \leq x_w^j \quad (3)$$

Since they correspond to the minimum of a convex function over a subset of its variables, these functions are convex. Furthermore, each one is defined over a different segment of $\left[x_w^L, x_w^U\right]$. Therefore, each one can be considered as a convex *piece* of an overall *piecewise* convex underestimator. The latter is fully suitable for application of the convex underestimation method for univariate functions which was described, in detail, in Part I (Gounaris and Floudas 2008). Figure 2 depicts how three functions $G_{wmj}(x_w)$, $j = 1, 2, 3$ are extracted from three adjacent underestimators $P_{wmj}(x)$, $j = 1, 2, 3$.

Note that continuity between pieces, $G_{wmj}(x_w^j) = G_{wm(j+1)}(x_w^j)$, which holds in the case of univariate functions, might in general not hold in the multivariate case. However, this does not limit the applicability of the univariate method, since its "outer" algorithm does not pose such a requirement.

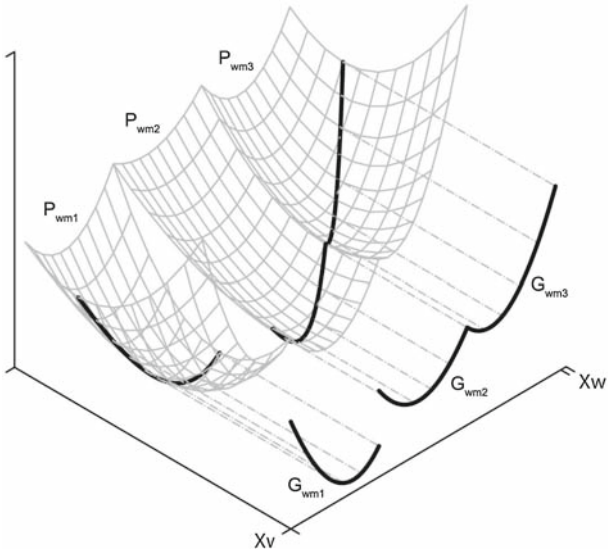

Fig. 2 Functions $P_{wmj}(x)$ and corresponding functions $G_{wmj}(x_w)$

Also note that there is no need to know these functions explicitly. The value of these functions at any x_w can be obtained reliably with the use of a standard local optimization solver, from the direct solution of the $(V-1)$-dimensional convex minimization problem of Eq. 3. A numerical approximation with the use of finite differences can be used for the various derivatives required.

Let $V_{wm}(x_w)$ be the piecewise linear underestimator obtained by the univariate method, and let it be the pointwise maximum of K_{wm} associated lines, that is:

$$V_{wm}(x_w) = \max\{T_{wmk}(x_w),\ \forall k = 1, 2, \ldots, K_{wm}\},\quad x_w^L \le x_w \le x_w^U \tag{4}$$

Without loss of generality, let us assume that the lines T_{wmk} are arranged in order of ascending slope, that is, $slope(T_{wm(k-1)}) < slope(T_{wmk})$, $k = 2, 3, \ldots, K_{wm}$, and that the set already includes the potential augmented tangents at the domain edges, designated as T_0 and T_{K+1} in Part I (Gounaris and Floudas 2008).

Univariate underestimator $V_{wm}(x_w)$ could, in principle, be considered as a multivariate function that is dependent to only one variable, x_w, and defined over the whole multidimensional (dimension V) subdomain D_{wm}. That is:

$$V_{wm}(x_w) \to V_{wm}(x),\quad x \in D_{wm} \tag{5}$$

Function $V_{wm}(x)$ is piecewise affine and consists of segments of V-dimensional hyperplanes. Since these hyperplanes depend only on the w^{th} variable, they are parallel to all standard basis vectors e_v with the exception of e_w (to which they are parallel only if the slope of the corresponding line T_{wmk} is zero). This function is a valid underestimator for the original function $f(x)$ across the whole subdomain D_{wm}.

Applying the aforementioned procedure for every permutation $m = 1, 2, \ldots, M_w$, we come up with a collection of such underestimating segments, each of which is a valid underestimator for the function $f(x)$ across a subset of its original domain D. Figure 3 depicts underestimators $V_{wm}(x)$ for eight permutations.

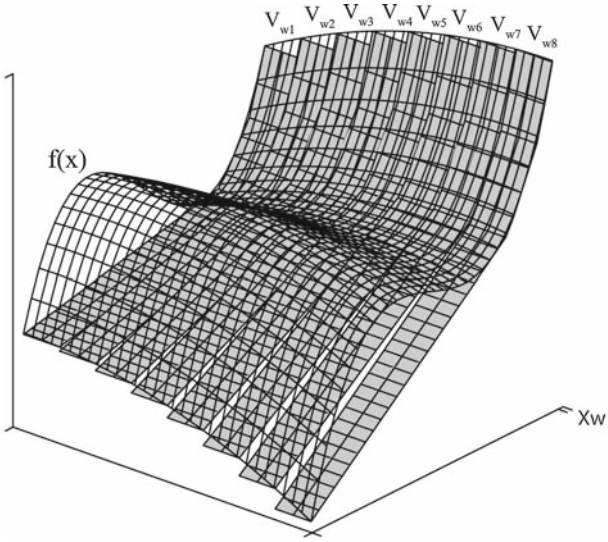

Fig. 3 Function $f(x)$ and underestimators $V_{wm}(x)$ for every permutation $m = 1, 2, \ldots, 8$

Let us use index $m = 0$ to refer to a piecewise affine underestimator, $V_{w0}(x)$, that would be valid for the whole domain of interest, D. In order to develop such an underestimator, we have to combine information from all the underestimating segments that correspond to permutations $m = 1, 2, \ldots, M_w$. This combination can be achieved back in the projection space, by computing the lower hull of the set of all underestimators $V_{wm}(x_w)$. In fact, one needs to consider only the vertex points of each underestimator $V_{wm}(x_k)$ (that is the points of intersection between two lines $T_{wm(k-1)}$ and T_{wmk}), as well as their end points $\left(x_w^L, T_{wm1}(x_w^L)\right)$ and $\left(x_w^U, T_{wm(K_{wm})}(x_w^U)\right)$. Any standard 2d convex hull algorithm (e.g., *Graham-Scan*) can be used for this purpose. The lower hull is a convex piecewise linear function $V_{w0}(x_w)$, and it is the pointwise maximum of K_{w0} lines, that is:

$$V_{w0}(x_w) = \max\left\{T_{w0k}(x_w),\ \forall k = 1, 2, \ldots, K_{w0}\right\},\quad x_w^L \le x_w \le x_w^U \tag{6}$$

By construction, this function is a convex underestimator of all pieces $G_{wmj}(x_w)$ for all permutations, that is:

$$V_{w0}(x_w) \le G_{wmj}(x_w),\quad x_w \in [x_w^{j-1}, x_w^j],\ \forall j = 1, 2, \ldots, N_w,\ \forall m = 1, 2, \ldots, M_w \tag{7}$$

Therefore, function $V_{w0}(x_w)$, if considered as $V_{w0}(x)$, is a valid underestimator for function $f(x)$ across its whole original domain D. Figure 4 depicts the different segments and illustrates how these can be combined into an overall underestimator. The side view corresponds to the two-dimensional projection and depicts the underestimators $V_{wm}(x_w)$ for all permutations $m = 1, 2, \ldots, M_w$, superimposed with their convex lower hull $V_{w0}(x_w)$.

For any selection of the active variable w, the method will yield a convex (piecewise affine) underestimator which would be valid for the whole domain of interest, D. However, the method can be independently applied for every variable being active (one at a time), leading to a collection of valid underestimators. The pointwise maximum of all these is itself a valid convex underestimator, and is tighter (or equally tight) to the original function than any of its predecessors. Thus, the resulting underestimator one could use is:

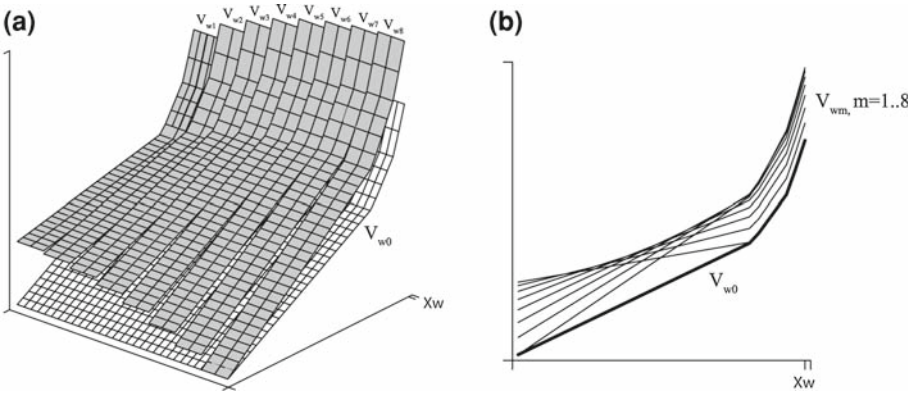

Fig. 4 Combination of underestimators $V_{wm}(x),\ m = 1, 2, \ldots, M_w$ into an overall underestimator $V_{w0}(x)$

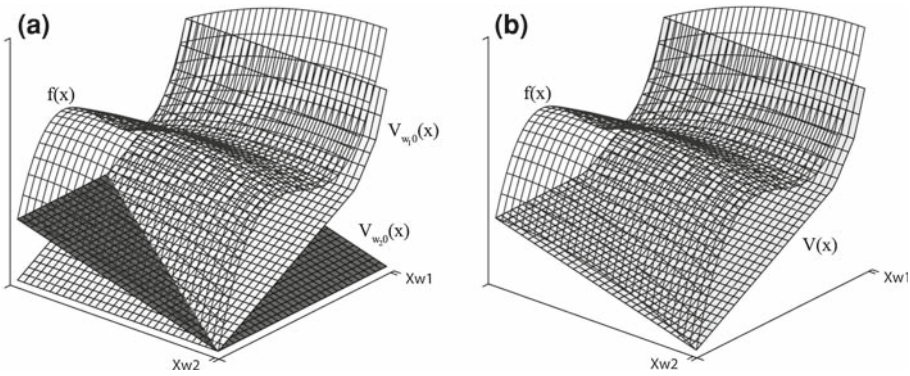

Fig. 5 Underestimators $V_{w0}(x),\ w = 1, 2, \ldots, V$ and their pointwise maximum, undererestimator $V(x)$

$$V(x) = \max\{V_{w0}(x),\ \forall w = 1, 2, \ldots, V\},\quad x \in D \tag{8}$$

Figure 5 depicts the different underestimators $V_{w0}(x)$ along with their pointwise maximum, underestimator $V(x)$.

Note that the underestimator $V(x)$ is also piecewise affine, and can be represented in the problem relaxation as a set of linear constraints. Since we do not know explicitly which hyperplanes $T_{w0k}(x_w) \rightarrow T_{w0k}(x),\ k = 1, 2, \ldots, K_{w0},\ w = 1, 2, \ldots, V$ contribute some part of theirs to the overall underestimator $V(x)$, all of them should be included in this relaxation, despite the fact that some may end up being redundant.

Let $L(x)$ be a convex underestimator of function $f(x)$ in domain D. Equation (9) shows the general relaxation of a nonlinear programming problem (NLP) into a convex nonlinear programming problem (CNLP).

$$\left\{\begin{array}{ll}\min\limits_{x} & f^{(0)}(x)\\ s.t. & f^{(q)}(x) \le 0\ ,\ \forall q = 1, 2, \ldots, Q\end{array}\right\} \rightarrow \left\{\begin{array}{ll}\min\limits_{x} & L^{(0)}(x)\\ s.t. & L^{(q)}(x) \le 0\ ,\ \forall q = 1, 2, \ldots, Q\end{array}\right\} \tag{9}$$

Since our method produces piecewise affine underestimators $L \equiv V$, the resulting convex relaxation is just a linear programming problem (LP), which takes the form of Formulation (10).

$$\left\{\begin{array}{ll}\min\limits_{x} & V^{(0)}(x)\\ s.t. & V^{(q)}(x)\leq 0\ ,\ \forall q=1,2,\ldots,Q\end{array}\right\}$$
$$\Leftrightarrow\left\{\begin{array}{lll}\min\limits_{\mu,x} & \mu & \\ s.t. & \mu\geq T^{(0)}_{w0k}(x_w), & \left\{\begin{array}{l}\forall k=1,2,\ldots,K^{(0)}_{w0}\\ \forall w=1,2,\ldots,V\end{array}\right\}\\ & T^{(q)}_{w0k}(x_w)\leq 0, & \left\{\begin{array}{l}\forall k=1,2,\ldots,K^{(q)}_{w0}\\ \forall w=1,2,\ldots,V\\ \forall q=1,2,\ldots,Q\end{array}\right\}\end{array}\right\} \tag{10}$$

4 Domain rotation

The methodology presented in Sect. 3 involves the minimization of underestimators $P_{wmj}(x)$, over all their variables with the exception of one, variable x_w, which is designated as "active". This results into univariate functions $G_{wmj}(x_w)$, whose two-dimensional epigraphs coincide with the projections of the $(V+1)$-dimensional epigraphs of the V-variate functions $P_{wmj}(x)$, that is:

$$\operatorname*{epi}_{x_w\in\left[x^L_w,x^U_w\right]}\left\{G_{wmj}(x_w)\right\}=\operatorname*{Pr}_{\{e_w,e_{V+1}\}}\operatorname*{epi}_{x\in D_i}\left\{P_{wmj}(x)\right\} \tag{11}$$

Whenever such a projection into spaces of lower dimensionality is involved, there is the possibility that some useful information is lost. Some of this lost information will be recovered if we opt to apply the methodology for every variable being "active", one at a time, which basically calls for projecting into V different two-dimensional planes, each one being parallel to a different basis vector e_v, $v=1,2,\ldots,V$. However, since there is a finite number of variables in our problem, there is a limited number of planes to which we can project. If we want to enhance further the collection of underestimators that we will eventually accumulate in the relaxation (thus improve our chances for better tightness/lower bound), we will have to project into additional planes, that do not correspond to some variable that is "natural" to the problem, rather than to some linear combination of theirs.

This can be achieved by applying an orthonormal transformation to the problem's variable space, that is:

$$x\rightarrow x'=R\cdot x \tag{12}$$

This transformation has to be orthogonal, which means that it should preserve the lengths of vectors and the angles between vectors. Furthermore, it should be an orientation-preserving transformation. A $V\times V$ matrix R that could provide such a transformation is called a *rotation matrix* and has to be a member of the special orthogonal group, that is:

$$R\in SO(V)\Leftrightarrow\left\{\begin{array}{ll}R^{-1}=R^T\\ |R|=+1\end{array}\right. \tag{13}$$

Section 4.1 discusses the selection of a suitable such matrix.

Let $f^{(R)}(x')$, $x'\in D'$ be the rotated counterpart of function $f(x)$, $x\in D$, such that domain D' fully encloses domain D, that is: $D'\supseteq D$. Let us also assume temporarily that function $f(x)$ can be defined in the expanded domain D' as well.

Lemma 1 *The following equalities hold:*

$$f^{(R)}(x') = f(R^T \cdot x') \tag{14}$$
$$\nabla f^{(R)}(x') = R \cdot \nabla f(R^T \cdot x') \tag{15}$$
$$\nabla^2 f^{(R)}(x') = R \cdot \nabla^2 f(R^T \cdot x') \cdot R^T \tag{16}$$

Applying the method for function $f^{(R)}(x')$, $x' \in D'$ leads to an underestimator $V^{(R)}(x')$ which is valid for the whole domain D'. Since $D' \supseteq D$, this underestimator will be valid for the original domain D as well. $V^{(R)}(x') = V^{(R)}(R \cdot x)$ is piecewise affine as it is the pointwise maximum of hyperplanes, which are not necessarily parallel to some basis vectors e_v anymore. However, all these hyperplanes are valid linear cuts that can be accumulated in the relaxation along with the ones that correspond to the original underestimator $V(x)$.

Depending on the availability of computational resources, many different rotation matrices can be used as each attempt will have a chance to improve the overall tightness through the accumulation of additional valid cuts. The final overall underestimator would then be the pointwise maximum of all these cuts:

$$V(x) = \max\left\{V^{(R_r)}(x) \;\; \forall r\right\}, \quad x \in D \tag{17}$$

Figure 6 depicts underestimators $V^{(R)}(x)$ for different matrices R along with their pointwise maximum, underestimator $V(x)$.

4.1 Matrix selection

As it has already been mentioned, a suitable matrix R has to be a member of the special orthogonal group, $SO(V)$. These matrices can be parameterized by $\Phi = V(V-1)/2$ parameters, called *Euler angles*. We shall denote these as φ_{pq}, $p = 1, 2, \ldots, V-1$, $q = p+1, p+2, \ldots, V$.

Let $R^{(pq)}$ be the $V \times V$ rotation matrix that corresponds to the rotation of the V-dimensional space as the (x_p, x_q)-plane (two-dimensional) rotates counter-clockwise (vector e_p towards vector e_q) by angle φ_{pq}.

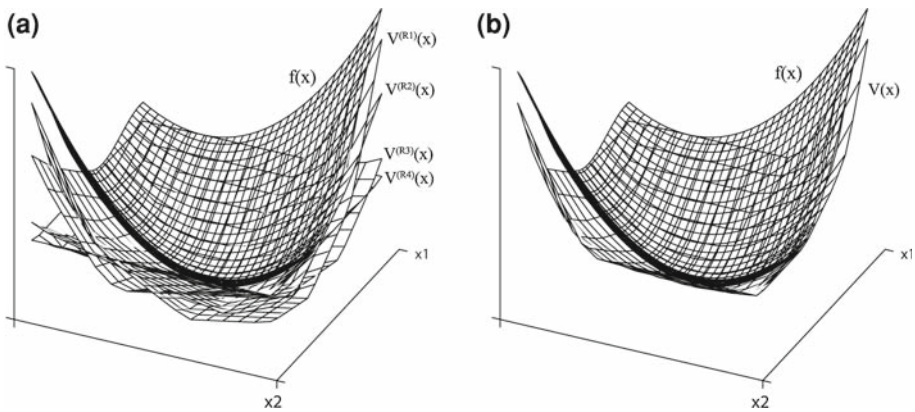

Fig. 6 Underestimators $V^{(R)}(x)$ for various matrices R and their pointwise maximum

The elements of this matrix, $R_{vu}^{(pq)}$, are as follows:

$$R_{vu}^{(pq)} = \begin{cases} cos\left(\varphi_{pq}\right), & \text{if } v = u = p, q \\ sin\left(\varphi_{pq}\right), & \text{if } v = p \text{ and } u = q \\ -sin\left(\varphi_{pq}\right), & \text{if } v = q \text{ and } u = p \\ 1 \quad , & \text{if } v = u \text{ and } v \neq p, q \\ 0 \quad , & \text{otherwise} \end{cases} \tag{18}$$

Each rotation in the V-dimensional space is equivalent to Φ sequential rotations of linearly independent two-dimensional planes around axes of rotation that are perpendicular to them. We make the convention that these planes would be the ones defined by pairs of our basis vectors (e_v, e_u), such that $1 \leq v < u \leq V$. The overall $V \times V$ rotation matrix would then be:

$$R = \prod_{p=1}^{V-1} \prod_{q=p+1}^{V} R^{(pq)} \tag{19}$$

Note that choosing $\varphi_{pq} = 0, \ \forall p, q$ leads to the identity matrix. Thus, it can be infered that the concept of domain rotation is a generalization of the original methodology (no rotation), which corresponded to the case where $R = I$.

4.2 Domain selection

Since we use an orthonormally transformed variable space, we have to make sure that the domain under consideration, $D' = [x_1'^L, x_1'^U] \times \ldots \times [x_V'^L, x_V'^U]$, completely includes the original domain of interest $D = [x_1^L, x_1^U] \times \ldots \times [x_V^L, x_V^U]$. A proper selection of this domain D' can be done according to Lemma 2.

Lemma 2 *Let $D = [x_v^L, x_v^U]$ be the original box domain of our problem, and let $D' = [x_r'^L, x_r'^U]$ be the "rotated" box domain that results after an orthonormal transformation. Let also y_{rv} be a binary parameter associated with the sign of R_{rv}, that is:*

$$y_{rv} = \begin{cases} 1, & \text{if } R_{rv} \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{20}$$

If the following hold for every $r = 1, 2, \ldots, V$:

$$\begin{aligned} x_r'^L &= \sum_{v=1}^{V} R_{rv} \left\{ y_{rv} x_v^L + (1 - y_{rv}) x_v^U \right\} \\ x_r'^U &= \sum_{v=1}^{V} R_{rv} \left\{ (1 - y_{rv}) x_v^L + y_{rv} x_v^U \right\} \end{aligned} \tag{21}$$

then D is a subset of D', that is: $D \subseteq D'$.

However, since the method will involve partitioning of this domain, it may not be necessary to consider all the subdomains thus produced. We can safely neglect all the subdomains that do not overlap at all with the original domain D, which is the domain of interest. This can be done easily by comparing the location of the corner points of each subdomain with the actual bounds x_v^L and x_v^U of domain D. Note that in the general case, underestimators $V_{wm}(x_w)$ for all permutations $m, \ 1 \leq m \leq M_w$ will no longer have the same end points, x_w^L and x_w^U, rather than each will have its own pair $x_w^{j_m^L}$ and $x_w^{j_m^U}$. However, this does not pose any complication since the combination of all permutations (through the two-dimensional

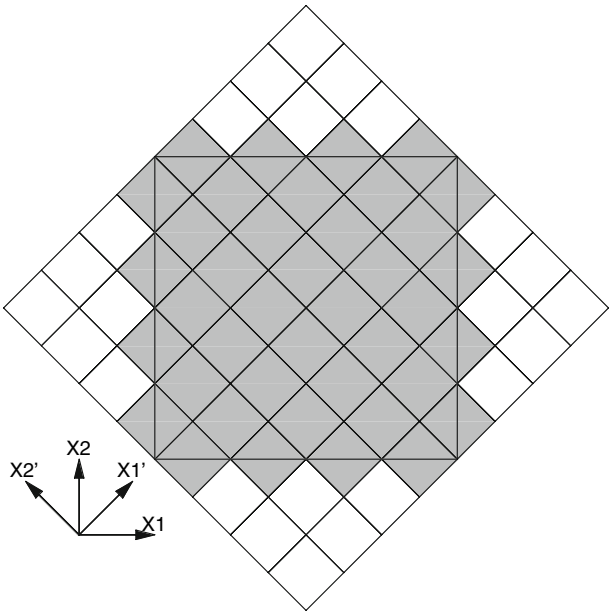

Fig. 7 Original domain D, rotated domain D', and identification of subdomains worth considering

convex hull of vertex and end points) can be trivially adapted to take this under consideration. Figure 7 depicts illustratively a "rotated" domain D' and which of its subdomains are worth considering after a given level of partitioning.

Also note that for the methodology to be applicable, function $f(x)$ has to be defined and be $\mathcal{C}^2$-continuous in the domain D' or at least in all its subdomains that we consider. This is usually the case, since it is common that the functions of our model preserve their continuity over a larger range than the one we select by imposing bounds on the various variables. However, discontinuities might occur outside domain D and we have to be cautious so as to exclude them from domain D'. This can be done with a sufficient increase of the level of partitioning, since such an action would monotonically reduce domain D'. Note that: $N \to +\infty \Rightarrow D' \to D$.

4.3 Level of partitioning selection

A final decision that has to be made when we opt to work with a rotated domain, is the level of partitioning that we will apply. This partitioning $N'_v,\ v = 1, 2, \ldots, V$ will correspond to the new variables x'_v which are linear combinations of the problem's natural variables x_v. This partitioning can be arbitrary selected, and the results are again expected to improve with higher partitioning. However, it is more natural that we define the level of partitioning only on the original domain D, since this is the one that has some natural meaning for our problem. Given a selected level of partitioning $N = \prod_{v=1}^{V} N_v$, we propose partitioning of domain D' in $N' = \prod_{v=1}^{V} N'_v$ subdomains such that:

$$N'_v = \lceil \sum_{u=1}^{V} |R_{vu}|\, N_u \rceil,\ \forall v = 1, 2, \ldots, V$$

This partitioning is suitable because it provides subdomains D'_i that are of comparable measures with the original subdomains D_i. As a consequence, underestimators $P'_{wmj}(x')$ and subsequently $V'(x')$ are expected to be of comparable tightness with their original counterparts. This is highly welcome because the additional linear cuts that will be accumulated in the relaxation have a good chance of not being redundant.

4.4 Computation of α parameters

As it is has been shown in Eq. 1, the selection of the α_v^i parameters involves computing bounds on the elements of the Hessian matrix. In the case of an orthonormally transformed problem, the Hessian matrix of the "rotated" function, $f^{(R)}$, is given by Eq. 16. For the calculation of new $\alpha_r^{i'}$ values (that correspond to the new "rotated" function and subdomains), bounds for the elements of the Hessian matrix of $f^{(R)}$ can be derived from the bounds of the original Hessian of f by performing standard interval arithmetic on this equation. The following two Lemmas are relevant with this computation.

Lemma 3 *Let $\underline{h_{vu}^{(i)}}$ and $\overline{h_{vu}^{(i)}}$ be respectively lower and upper bounds of $\partial^2 f/\partial x_v x_u$ that are valid for the entire subdomain D_i, and let $\underline{h_{rs}^{(R)(i')}}$ and $\overline{h_{rs}^{(R)(i')}}$ be respectively lower and upper bounds of $\partial^2 f^{(R)}/\partial x'_r x'_s$ that are valid for the entire "rotated" subdomain D'_i, such that $D'_i \subseteq D_i$. The following expressions provide valid such bounds:*

$$\begin{aligned}
\underline{h_{rs}^{(R)(i')}} &= \sum_{v=1}^{V}\sum_{u=1}^{V}\left\{R_{rv}\left[y_{rv,su}\left(\underline{h_{vu}^{(i)}}\right) + (1-y_{rv,su})\left(\overline{h_{vu}^{(i)}}\right)\right]R_{us}^T\right\} \\
\overline{h_{rs}^{(R)(i')}} &= \sum_{v=1}^{V}\sum_{u=1}^{V}\left\{R_{rv}\left[(1-y_{rv,su})\left(\underline{h_{vu}^{(i)}}\right) + \left(\overline{h_{vu}^{(i)}}\right)y_{rv,su}\right]R_{us}^T\right\}
\end{aligned} \tag{22}$$

where $y_{rv,su}$ is a binary parameter associated with the sign of $R_{rv}R_{su}$, that is:

$$y_{rv,su} = \begin{cases} 1, & \text{if } R_{rv}R_{su} \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

Lemma 3 allows us to compute intervals for the elements of the new Hessian, given intervals of the respective elements of the original one. However, for these intervals to be valid for the complete subdomain of interest, D'_i, we need to use subdomains D_i that completely enclose the former, that is: $D'_i \subseteq D_i$. Furthermore, these box subdomains should be as small as possible, so as to avoid unnecessary overestimation of the $\alpha_r^{i'}$ values later on. Lemma 4 dictates a proper selection of such intervals D_i.

Lemma 4 *Let $D'_i = [x_r'^L, x_r'^U]$ be the "rotated" box subdomain of interest that results after an orthonormal transformation, and let $D_i = [x_v^L, x_v^U]$ be a box subdomain of the original variable space D. Let also y_{rv} be a binary parameter associated with the sign of $R_{rv} = R_{vr}^T$, as defined in Lemma 2.*

If the following hold for every $v = 1, 2, \ldots, V$:

$$\begin{aligned}
x_v^L &= \sum_{r=1}^{V} R_{rv}^T\left\{y_{rv}x_r'^L + (1-y_{rv})x_r'^U\right\} \\
x_v^U &= \sum_{r=1}^{V} R_{rv}^T\left\{(1-y_{rv})x_r'^L + y_{rv}x_r'^U\right\}
\end{aligned} \tag{24}$$

then D_i is the smallest suitable superset of domain D'_i.

Lemma 4 provides the tightest possible box subdomains that are orthogonal to the original coordinate system, and completely enclose the corresponding "tilted" subdomains that are orthogonal to the new –rotated– coordinate system. Note the inverse relationship between Lemmas 2 and 4.

5 Complexity analysis

As described in the previous sections, the overall methodology involves three nested loops at the rotation-, active variable-, and permutation-level. In fact, this inherent structure of the algorithm makes it suitable for implementation in a distributed computing environment, where satisfactory scaling is expected. The distribution of computing work can occur at either one of these three levels, and investigating the different potential strategies actually poses a very interesting theme for further research.

Let us now discuss the overall complexity of the algorithm. We denote with R the number of different rotation matrices to be used, with V the number of variables, and with N the level of partitioning we choose for the range of each variable. The complexity, C, is given by Eq. 25.

$$C = \mathcal{O}\left(R \times V \times N^{V-1} \times N\right) \tag{25}$$

As it has already been mentioned in the first part of this series (Gounaris and Floudas 2008), the univariate algorithm inherits the linear time complexity of the Graham-Scan convex hull algorithm for sorted entries, that is $\mathcal{O}(N)$.

Assuming that the partitioning is uniform (that is we select to partition the range of every variable into an equal number of subdomains, N), the total number of permutations that we need to consider is N^{V-1}. We should keep in mind that this assumption results in an exhaustive domain partitioning that is not always necessary. For example, one could choose a subset of the variables whose range would be partitioned into a smaller number of subdomains, or even not partitioned at all (i.e., $N_v = 1$). Good candidates for such an action are those variables v that do not participate strongly in the non-convexities and correspond to small (or even zero) α_v values.

Finally, the quantity $R \times V$ (number of matrices used times number of possible active variables) basically accounts for all two-dimensional planes that we opt to project to. As a general rule, the larger the number of different projection planes used, the more the cuts that are accumulated and the tighter the relaxation that ends up being formulated. The best possible results will be obtained if one samples the whole Euler angle space at a very fine resolution $\Delta\varphi$. Note that for our application, it suffices to select Euler angles from the range $[0, \frac{\pi}{2})$, therefore the total number of rotation matrices used would be: $R = \left(\lceil \frac{\pi/2}{\Delta\varphi} \rceil\right)^{\Phi}$. The latter assumes equally fine sampling of all $\Phi = V(V-1)/2$ Euler angles of the V-dimensional space. Again, this is an exhaustive strategy and does not necessarily ought to be followed in the actual implementation. In fact, a good lower bound can usually be obtained without the need for rotation. Furthermore, considering only one active variable will probably be sufficient, as it is illustrated by the first part of Fig. 5 where each of the two depicted underestimators provides a tight lower bound. However, the accumulation of many affine cuts that result from domain rotations will contribute towards obtaining good tightness across the whole domain under consideration. Therefore, whenever just a good lower bound is required, like in the case of an objective function, one need not perform rotation, while rotation, which adds com-

Table 1 Lower bound results for a collection of bivariate functions

#	Function $f(x)$	$[x_1^L, x_1^U] \times [x_2^L, x_2^U]$	αBB	$N_1 = N_2$						GO f^*
				2	4	8	16	32	64	
1	$\left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cos(x_1) + 10$	$[-5, 10] \times [0, 15]$	−884	−165	−23.1	−1.39	0.073	0.316	0.359	0.398
2	$x_1^4 + 4x_1^3 + 4x_1^2 + x_2^2$	$[-3, 3] \times [-3, 3]$	−127	−26	−3.16	−0.27	−0.008	0	0	0
3	$100\left(x_1^2 - x_2\right)^2 + (x_1 - 1)^2$	$[-2, 2] \times [-2, 2]$	−4399	−1101	−122.2	−20.7	−3.05	−0.754	−0.115	0
4	$4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-3, 3] \times [-1.5, 1.5]$	−998	−250	−42.9	−4.98	−1.032	−1.032	−1.032	−1.032
5	$\prod_{j=1}^{2}\left\{\sum_{i=1}^{5} icos\left[(i+1)x_j + 1\right]\right\}$	$[-1, 1] \times [-1, 1]$	−1 E+4	−2530	−593	−207	−189.0	−188.1	−187.2	−186.7
6	$\frac{1}{4}x_1^4 - \frac{1}{2}x_1^2 + \frac{1}{10}x_1 + \frac{1}{2}x_2^2$	$[-1, 1] \times [-1, 1]$	−0.53	−0.38	−0.35	−0.35	−0.35	−0.35	−0.35	−0.35
7	$10^6x_1^2 + x_2^2 - \left(x_1^2 + x_2^2\right)^2 + 10^{-6}\left(x_1^2 + x_2^2\right)^4$	$[-0.01, 0.01] \times [-2, 2]$	−92	−23	−12	−12	−12	−12	−12	−12
8	$(-13 + x_1 + ((5 - x_2)x_2 - 2)x_2)^2$ $+ (-29 + x_1 + ((x_2 + 1)x_2 - 14)x_2)^2$	$[0, 20] \times [-5, 10]$	−5 E+6	−1 E+6	−7 E+4	−5448	−750	−24.9	−2.83	0
9	$\sum_{i=1}^{10}\{2 + 2i - [exp(ix_1) + exp(ix_2)]\}^2$	$[0.1, 0.4] \times [0.1, 0.4]$	−3 E+4	−4944	−58.8	103.6	121.5	124.0	124.3	124.4
10	$x_1^2 + \left(x_2 - x_1^2 - 1\right)^2 + \sum_{i=1}^{29}\left(x_2 - \left(x_1 + \frac{i}{29}x_2\right)^2 - 1\right)^2$	$[-1, 1] \times [-1, 1]$	−784	−99.3	−6.89	0.35	0.634	0.663	0.663	0.663
11	$(x_1 - 0.2)^2 + \left(2x_1^2 + x_2^2 - 1\right)^2$ $+10^{-5}\left(exp\left(\frac{x_2}{10}\right) - exp(0.2) + exp\left(\frac{x_1}{10}\right) - exp(0.1)\right)^2$ $+10^{-5}\left(exp\left(\frac{x_2}{10}\right) - exp(0.1)\right)^2$	$[0, 0.5] \times [0, 0.5]$	−0.06	0.13	0.153	0.153	0.153	0.153	0.153	0.153
12	$(x_1 - 1)^2 + (x_2 - 1)^2 + (x_1 + 2x_2 - 3)^2 + (x_1 + 2x_2 - 3)^4$	$[0, 2] \times [0, 2]$	−213	−54.4	−2.53	−0.097	−0.004	−1E-4	−1 E-5	0
13	$(2x_1 + x_2 - 3)^2 + (x_1x_2 - 1)^2$	$[0, 4] \times [0, 4]$	−45.8	−4.21	−0.34	−0.05	−0.02	−0.009	−0.005	0

Table 1 continued

#	Function $f(x)$	$[x_1^L, x_1^U] \times [x_2^L, x_2^U]$	αBB	$N_1 = N_2$						GO f^*
				2	4	8	16	32	64	
14	$\left(2x_1 - x_2 + \left(x_1 + \frac{4}{3}\right)^3/18\right)^2 + \left(2x_2 - x_1 + \left(x_2 + \frac{5}{3}\right)^3/18\right)^2$	$[-1, 1] \times [-1, 1]$	−6.74	−0.31	−0.13	−0.044	−0.008	−0.002	−2 E-4	0
15	$\left(x_1 + 2\left(x_1 + \frac{4}{3}\right)^3/54 + \left(x_2 + \frac{5}{3}\right)^3/54\right)^2 + \left(x_2 + 2\left(x_2 + \frac{5}{3}\right)^3/54 + \left(x_1 + \frac{4}{3}\right)^3/54\right)^2$	$[-1, 1] \times [-1, 1]$	−1.38	−0.14	−0.062	−0.021	−5 E−4	−2 E−4	−1 E−4	0
16	$((3 - 2x_1)\, x_1 - 2x_2 + 1)^2 + ((3 - 2x_2)\, x_2 - x_1 + 1)^2$	$[-2, 2] \times [-2, 2]$	−562	−89.2	−12.0	−1.18	−0.19	−0.04	−0.009	0
17	$\left(x_1\left(2 + 5x_1^2\right) + 1 - x_2\,(1 + x_2)\right)^2 + \left(x_2\left(2 + 5x_2^2\right) + 1 - x_1\,(1 + x_1)\right)^2$	$[-2, 2] \times [-2, 2]$	-3 E+4	−1829	−61.9	−3.66	−0.43	−0.07	−0.01	0
18	$\left(\frac{x_1 + x_2}{2}\right)^2 + \left(x_1^2 + x_2^2 - \frac{2}{3}\right)^2$	$[-2, 2] \times [-2, 2]$	−138	−33.7	−2.67	−0.35	−0.052	−0.009	−0.001	0
19	$2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$	$[-5, 5] \times [-5, 5]$	−3900	−967	−58.4	−12.3	−1.043	−2 E−5	−8 E−6	0
20	$\sum_{i=1}^{2}\left\{x_i^2 - 0.1cos(5\pi x_i)\right\}$	$[-1, 1] \times [-1, 1]$	−22.9	−5.37	−1.182	−0.369	−0.206	−0.2	−0.2	−0.2
21	$-cos(x_1)cos(x_2)exp\left(-(x_1 - \pi)^2 - (x_2 - \pi)^2\right)$	$[-10, 10] \times [-10, 10]$	−1 E+5	−9861	−533	−53.1	−4.69	−1.20	−1.001	−1
22	$\left(1 - 8x_1 + 7x_1^2 - \frac{7}{3}x_1^3 + \frac{1}{4}x_1^4\right)x_2^2exp(-x_2)$	$[0, 5] \times [0, 6]$	−1 E+5	−7041	−374	−29.4	−2.92	−2.375	−2.348	−2.346
23	$sin(x_1 + x_2) + (x_1 - x_2)^2 - 1.5x_1 + 2.5x_2 + 1$	$[-1.5, 4] \times [-3, 3]$	−15.7	−7.35	−3.24	−1.97	−1.930	−1.921	−1.914	−1.913
24	$1 + sin^2(x_1) + sin^2(x_2) - 0.1exp\left(-x_1^2 - x_2^2\right)$	$[-10, 10] \times [-10, 10]$	−8179	−2042	−134	−7.90	0.32	0.9	0.9	0.9

Both variables active. No rotation

putational expense, should be performed only in special cases like in problem constraints that are expected to be active.

6 Computational results

The methodology was implemented in *C* and applied on various test functions. These were drawn from suitable collections that have been presented in the literature (Moré et al. 1981; Ge and Qin 1990; Ali et al. 2005).

Table 1 presents the results for a set of two-variable functions. It can be seen that the lower bounds obtained improve considerably the ones obtained from the original αBB method, and that the results benefit from finer partitioning of the domain. All runs were performed on a 3.20 GHz Intel$^{(R)}$ Pentium$^{(R)}$ 4 processor with 1 Gb of RAM. Runs with partitioning level 32×32 averaged a CPU time of 0.40 s, while the more tedious runs with partitioning level 64×64 averaged 1.86 s, a result that is in good agreement with Eq. 25 (for bivariate functions, doubling the level of partitioning should result in a four-fold increase in CPU time). Maximum run times for these two cases were 1.16 s and 6.41 s, respectively. It should be noted that our implementation uses NPSOL (Gill et al. 1998) for performing function evaluations of functions $G_{wmj}(x_w)$. However, since this corresponds to minimizations of convex objective functions subject to only bound constraints, the CPU times are expected to be lower if we use a local solver that makes use of this information and utilizes simpler data structures.

To illustrate the tightness of the underestimators across the whole domain under consideration, we present Fig. 8 that depicts the plots for two highly nonlinear functions. The level of partitioning, $N_1 \times N_2$, and the resolution of Euler angle sampling, $\Delta\varphi_{12}$, used in each case is mentioned, as well as some additional information regarding the improvement of lower bound, and the number of linear cuts that have to be accumulated in the relaxation.

Table 2 provides results for various nonconvex functions with a higher number of variables. The underestimators produced are generally very tight. Note the particularly good behavior

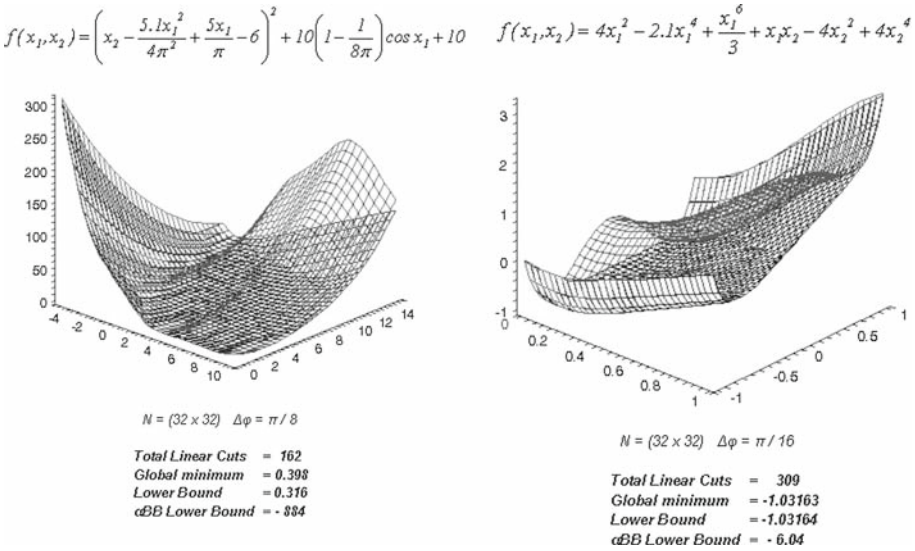

Fig. 8 Piecewise planar underestimators of some nonconvex functions

Table 2 Lower bound results for a collection of multivariate functions

#	Function $f(x)$	$[x_v^L, x_v^U]\forall v$	V	αBB	$N_v\ \forall v$				GO f^*
					2	4	8	16	
1	$x_1x_2 + x_1x_2x_3$	$[-1, 1]$	(3)	-4	-2	-2	-2	-2	-2
2	$x_1x_2 - x_2x_3 - x_3x_4 + x_1x_2x_3 - x_1 + x_4$	$[0, 1]$	(4)	-1.54	-1.062	-1.012	-1.002	-1.001	-1
3	$x_1x_2 + x_2x_3x_4 + x_2x_4 - x_1x_3x_4x_5 + x_2x_3x_5 - x_1x_5$	$[-1, 1]$	(5)	-15	-6.023	-6	-6	-6	-6
		$[1, 3]$		-73.45	-66	-66	-66	-66	-66
4	$\sum_{i=0}^{5}\sum_{j=1}^{6-i}\prod_{k=j}^{j+i} x_k$	$[-1, 1]$	(6)	-60.15	-10.21	-4.13	-3.175	-3.072	-3
5	$\sum_{i=1}^{2}\left\{4x_i^2 - 2.1x_i^4 + x_i^6/6 + x_ix_{i+1} - 4x_{i+1}^2 + 4x_{i+1}^4\right\}$	$[-2, 2]$	(3)	-411.2	-101.6	-14.78	-1.941	-1.291	-1
6	$\sum_{i=1}^{V}\left\{-0.1cos(5\pi x_i) + x_i^2\right\}$	$[-1, 1]$	(3)	-34.31	-8.05	-1.772	-0.554	-0.3095	-0.3
			(4)	-45.74	-10.73	-2.363	-0.738	-0.4127	-0.4
7	$20 - 20exp\left(-0.02\sqrt{V^{-1}\sum_{i=1}^{V} x_i^2}\right) + e - exp\left(V^{-1}\sum_{i=1}^{V} cos(2\pi x_i)\right)$	$[1, 3]$	(3)	-109.9	-24.32	-4.684	-0.019	0.396	0.396
			(4)	-109.9	-24.31	-4.679	-0.019	0.396	0.396
8	$\frac{\pi}{V}\left[10sin^2(\pi x_1) + \sum_{i=1}^{V-1}(x_i - 1)^2\left(10sin^2(\pi x_{i+1})\right) + (x_V - 1)^2\right]$	$[-10, 10]$	(3)	-3 E+6	-7 E+5	-2 E+5	-4 E+4	-9 E+3	0
			(4)	-3 E+6	-7 E+5	-2 E+5	-5 E+4	-1 E+4	0
9	$\sum_{i=1}^{V-1}\prod_{j=i}^{i+1}\sum_{k=1}^{5} kcos\left[(k+1)x_j + k\right]$	$[-10, 10]$	(3)	-2 E+6	-5 E+5	-1 E+5	-3 E+4	-8 E+3	-4 E+2
10	$100\left(x_2 - x_1^2\right)^2 + (1 - x_1)^2 + 90\left(x_4 - x_3^2\right)^2 + (1 - x_3)^2$ $+10.1\left[(1 - x_2)^2 + (1 - x_4)^2\right] + 19.8\left[(1 - x_2) + (1 - x_4)\right]$	$[0, 1]$	(4)	-217.9	-40.1	-4.405	-0.670	-0.175	0

Table 2 continued

#	Function $f(x)$	$[x_v^L, x_v^U]\forall v$	V	αBB	$N_v\ \forall v$				GO f^*
					2	4	8	16	
11	$(x_1+10x_2)^2+5(x_3-x_4)^2+(x_2-2x_3)^4+10(x_1-x_4)^4$	[0, 1]	(4)	−57.23	−12.39	−0.639	−0.064	−0.010	0
12	$x_1^{-2}x_2^{-1.5}x_3^{1.2}x_4^3-3x_3^{0.5}+x_2-4x_4$	[1, 10]	(4)	−1 E+6	−4 E+5	−1 E+5	−2 E+4	−509	−38
13	$0.4x_1^{2/3}x_3^{-2/3}+0.4x_2^{2/3}x_4^{2/3}+10-x_1-x_2$	[0.1, 10]	(4)	−1088	−287	−61.6	−9.2	−9.2	−9.2
14	$-\frac{1}{2}\sum_{i=1}^{V}\left(x_i^4-16x_i^2+5x_i\right)$	[−5, 2]	(3)	−2409	−526.3	−300	−300	−300	−300
			(4)	−3212	−695.9	−400	−400	−400	−400
			(5)	−4015	−865.4	−500	−400	−500	−500

All variables active. No rotation

in the case of multilinear functions (# 1-4), where even a small level of partitioning is usually sufficient to considerably reduce, or even completely close, the gap between lower bound and actual global optimum.

7 Conclusions

In Part I, we presented a novel convex underestimation method for general nonconvex $\mathcal{C}^2$-continuous functions that is based on a piecewise application of the αBB underestimator (Maranas and Floudas 1994; Androulakis et al. 1995; Adjiman and Floudas 1996; Adjiman et al. 1998a,b). In this paper, we presented an extension of the method in order to address multivariate functions. We proposed projections into spaces of lower dimensionality so as to take advantage of the high quality results of the univariate case as well as optional orthonormal tranformations of the problem, in an attempt to improve the results already obtained. The multivariate underestimators are piecewise affine, therefore the whole relaxation problem can be formulated as an LP. Extensive computational testing demonstrated that the method yields very tight underestimators, and the improvement over the original αBB underestimator is significant. The level of partitioning is crucial to the quality of the results, both in terms of lower bound and tightness over the whole domain under consideration. The orthonormal tranformations improve the overall tightness, but have little effect on the value of the lower bound. Since they add to the computational expense, their use is recommended only in cases where they are expected to have a significant impact. Which rotation matrices to use is also crucial for obtaining good results and computational performance. Their optimum selection, possibly problem-specific, is an interesting open question.

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

References

Achenie, L.K.E., Biegler, L.T.: A superstructure based approach to chemical reactor network synthesis. Comput. Chem. Eng. **14**, 23–40 (1990)

Adjiman, C.S., Floudas, C.A.: Rigorous convex underestimators for general twice-differentiable problems. J. Global 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)

Aggarwal, A., Floudas, C.A.: Synthesis of general separation sequences — nonsharp separations. Comput. Chem. Eng. **14**, 631–653 (1990)

Akrotirianakis, I.G., Floudas, C.A.: A new class of improved convex underestimators for twice continuously differentiable constrained NLPs. J. Global Optim. **30**, 367–390 (2004a)

Akrotirianakis, I.G., Floudas, C.A.: Computational experience with an new class of convex underestimators: box-constrained NLP problems. J. Global Optim. **29**, 249–264 (2004b)

Ali, M.M., Khompatraporn, C., Zabinsky, Z.B.: A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. J. Global Optim. **31**, 635–672 (2005)

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. Global Optim. **7**, 337–363 (1995)

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

Ciric, A.R., Floudas, C.A.: A retrofit approach of heat exchanger networks. Comp. Chem. Eng. **13**, 703–715 (1989)
Esposito, W.R., Floudas, C.A.: Global optimization in parameter estimation of nonlinear algebraic models via the error-in-variables approach. Ind. 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. 7, 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. Comp. Chem. Eng. **29**, 1185–1202 (2005)
Floudas, C.A., Fung, H.K., McAllister, S.R., Monnigmann, M., Rajgaria, R.: Advances in protein structure prediction and de novo protein design: a review. Chem. Eng. Sci. **61**, 966–988 (2006)
Floudas, C.A., Kreinovich, V.: Towards optimal techniques for solving global optimization problems: symmetry-based approach. In: Torn, A., Zilinskas, J. (ed.) 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)
Ge, R.P., Qin, Y.F.: Globally convexized functions. Appl. Math Comput. **35**, 131–158 (1990)
Gill, P.E., Murray, W., Saunders M.A., Wright, M.H.: User's guide for NPSOL 5.0: a fortran package for nonlinear programming. Technical Report SOL 86–1 (1998)
Gounaris, C.E., Floudas, C.A.: Tight convex underestimators for $\mathcal{C}^2$-continuous functions: I. Univariate functions. J. Global Optim. (2008). doi:10.1007/s10898-008-9287-9
Gümüş, Z.H., Floudas, C.A.: Global optimization of nonlinear bilevel programming problems. J. Global 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)
Horst, R., Pardalos, P.M., Thoai N.V.: Introduction to Global Optimization (Second Edition). Kluwer Academic Publishers, (2000)
Horst, R., Tuy, H.: Global Optimization: Deterministic Approaches, 3rd ed. Springer (2003)
Kokossis, A.C., Floudas, C.A.: Optimization of complex reactor networks: I-isothermal operation. Chem. Eng. Sci. **43**, 595–614 (1990)
Kokossis, A.C., Floudas, C.A.: Optimization of complex reactor networks: II-nonisothermal operation. Chem. Eng. Sci. **49**, 1037–1051 (1994)
Liberti, L., Pantelides, C.C.: Convex envelopes of monomials of odd degree. J. Global Optim. **25**, 157–168 (2003)
Maranas, C.D., Floudas, C.A.: Global minimum potential energy conformations of small molecules. J. Global Optim. **4**, 135–170 (1994)
Maranas, C.D., Floudas, C.A.: Finding all solutions of nonlinearly constrained systems of equations. J. Global Optim. **7**, 143–182 (1995)
Maranas, C.D., McDonald, C.M., Harding, S.T., Floudas, C.A.: Locating all azeotropes in homogeneous azeotropic systems. Comp. Chem. Eng. **20**, S413–S418 (1996)
McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: part I – convex underestimating problems. Mathem. Prog. **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. Gl. 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. Comp. Chem. Eng. **19**, 1111–1141 (1995)
McDonald, C.M., Floudas, C.A.: GLOPEQ: a new computational tool for the phase and chemical equilibrium problem. Comp. 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. Global Optim. **29**, 125–155 (2004)
Meyer, C.A., Floudas, C.A.: Convex envelopes for edge-concave functions. Math Prog. **103**, 207–224 (2005)
Moré, J.J., Garbow, B.S., Hillstrom, K.E.: Testing unconstrained optimization software. ACM Trans. Math Software **7**, 17–41 (1981)
Ryoo, H.S., Sahinidis, N.V.: Analysis of bounds for multilinear functions. J. Global Optim. **19**, 403–424 (2001)

Sherali, H.D., Adams, W.P.: Reformulation-linearization techniques in discrete and continuous optimization. Kluwer Academic Publishers, (2002)
Tardella, F.: On the existence of polyhedral convex envelopes. In: Floudas, C.A., Pardalos, P.M., (eds) Frontiers in Global Optimization. Kluwer Academic Publishers (2003)
Tawarmalani, M., Sahinidis, N.V.: Semidefinite relaxations of fractional programs via novel convexification techniques. J. Global 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. Prog. **93**, 247–263 (2002b)
Yee, T.F., Grossmann, I.E.: Simultaneous-optimization models for heat integration. 2-heat exchanger network synthesis. Comp. Chem. Eng. **14**, 1165–1184 (1990)
Yeomans, H., Grossmann, I.E.: A systematic modeling framework of superstructure optimization in process synthesis. Comp. Chem. Eng. **23**, 709–731 (1999)
Zabinsky, Z.B.: Stochastic Adaptive Search for Global Optimization. Kluwer Academic Publishers (2003)