Comparison of convex relaxations of quadrilinear terms

Sonia Cafieri[1] Jon Lee[2] Leo Liberti[1]

[1]LIX, École Polytechnique, 91128 Palaiseau, France

[2]Dept. of Mathematical Sciences, IBM T.J. Watson Research Center,
PO Box 218, Yorktown Heights, N.Y. 10598, USA

Abstract

In this paper we compare four different ways to compute a convex linear relaxation of a quadrilinear monomial on a box, analyzing their relative tightness. We computationally compare the quality of the relaxations, and we provide a general theorem on pairwise-comparison of relaxation strength, which applies to some of our pairs of relaxations for quadrilinear monomials. Our results can be used to configure a spatial Branch-and-Bound global optimization algorithm. We apply our results to the Molecular Distance Geometry Problem, demonstrating the usefulness of the present study.

Keywords quadrilinear; convex relaxation; reformulation; global optimization, spatial Branch-and-Bound

1 Introduction

Monomials of degree four occur often in the mathematical programming formulation of important applications, such as the Molecular Distance Geometry Problem [7] and the Hartree-Fock Problem [10]. Such applications can be modeled as non-convex Nonlinear Programs (NLPs), which often exhibit several local and global optima. The most widely employed deterministic method for the global solution of non-convex NLPs is the spatial Branch-and-Bound (sBB) algorithm [1]. One of the most crucial steps of this algorithm is the computation of the lower bound at each sBB node, which is usually based on providing tight convex and/or concave relaxations (or preferably envelopes) for each term appearing nonlinearly in the objective functions and/or constraints.

Convex/concave envelopes in explicit form currently exist for concave/convex univariate functions [14], bilinear terms [12], trilinear terms [13], univariate monomials of odd degree [11] and fractional terms [15]. The multivariate monomials of smallest degree for which the convex envelopes are not completely known are the quartic ones. We focus in particular on the quadrilinear term $x_1x_2x_3x_4$, for which we compare four convex relaxations.

A key idea in sBB is that given a sufficiently rich set of "elementary" convex envelopes, one can compose convex relaxations (albeit not envelopes) of complex functions relatively easily. For example, given two functions $f(x)$ and $g(x)$ for which the convex/concave envelopes are available, a convex relaxation for the product $f(x)g(x)$ can be obtained by applying the bilinear convex envelope to the product w_1w_2, where the necessary "defining constraints" $w_1 = f(x)$, $w_2 = g(x)$, can be replaced by the convex/concave envelopes of f and of g. This strategy, however, due to the associativity of the product, yields sometimes non-unique ways of combining terms when we have products of many functions and consequently different convex relaxations.

The relaxation strength is crucial for the performance of the sBB process. In this paper, we undertake a computational as well as a theoretical investigation of the relative tightness of four relaxations of quadrilinear terms. Our results indicate, not surprisingly, that a grouping leading to the exploitation of a trilinear envelope yields tightest bounds. This is important especially in view of the fact that the traditional grouping used by sBB algorithms [1, 3] is $((x_1x_2)x_3)x_4$. We remark that our main result (Theorem 1 in Section 2) can be applied in full generality to any pair of relaxations for which one is derived from the other by a natural contraction operation — for products this amounts to the deletion of parentheses.

2 Comparison of relaxations

Given a quadrilinear monomial $x_1x_2x_3x_4$, we consider the following four types of term grouping: $((x_1x_2)x_3)x_4$, $(x_1x_2)(x_3x_4)$, $(x_1x_2x_3)x_4$, $(x_1x_2)x_3x_4$. We will derive four corresponding linear relaxations for $x_1x_2x_3x_4$. Let us consider the following sets:

$$
\begin{aligned}
S_1 &= \{(x,w) \in \mathbb{R}^4 \times \mathbb{R}^3 \,|\, x_i \in [x_i^L, x_i^U],\, w_1 = x_1x_2, w_2 = w_1x_3, w_3 = w_2x_4\},\\
S_2 &= \{(x,w) \in \mathbb{R}^4 \times \mathbb{R}^3 \,|\, x_i \in [x_i^L, x_i^U],\, w_1 = x_1x_2, w_2 = x_3x_4, w_3 = w_1w_2\},\\
S_3 &= \{(x,w) \in \mathbb{R}^4 \times \mathbb{R}^2 \,|\, x_i \in [x_i^L, x_i^U],\, w_1 = x_1x_2x_3, w_2 = w_1x_4\},\\
S_4 &= \{(x,w) \in \mathbb{R}^4 \times \mathbb{R}^2 \,|\, x_i \in [x_i^L, x_i^U],\, w_1 = x_1x_2, w_2 = w_1x_3x_4\}.
\end{aligned}
$$

To derive the four relaxations, we exploit a bilinear envelope thrice for the first two cases, a trilinear envelope followed by a bilinear envelope for S_3 and a bilinear envelope followed by a trilinear envelope for S_4.

In the next subsections we first describe a computational study and get some significant evidence for what is the best relaxation, then we provide a theorem that confirms the validity of some of the obtained results.

2.1 Computational comparison

We generated a set of eighty test instances by varying the signs of the bounds on the variables x_i and, starting from the same initial width of the bound intervals for all variables, progressively reducing the width of the bound interval of x_i, $i = 1,2,3$. A bound interval $[x_i^L, x_i^U]$ is changed to $[x_i^L + 1/2, x_i^U - 1/2]$. This simulates a typical behavior of a sBB algorithm, that progressively reduces the size of the variable intervals.

We compare the considered relaxations in terms of the volume of the corresponding enveloping polytopes. This method of comparison, introduced in [8], is independent of any objective function.

Because exploiting envelopes for bilinear and trilinear terms leads to an increased number of variables, so that the obtained polytopes S_1, S_2 live in $\mathbb{R}^7$, and S_3, S_4 in $\mathbb{R}^6$. So we project the polytopes on $\{(x, f(x) := x_1x_2x_3x_4) \in \mathbb{R}^5\}$ in order to compare the results. The projection is computed by using the software `CDD` [4]. Then, the volume of each of the obtained projected polytopes is computed by using the `LRS` code [2]. Both codes provide results in exact arithmetic.

For each problem instance, we compare the volumes of the polytopes corresponding to S_1, S_2, S_3, S_4 projected on $\mathbb{R}^5$. As expected, reducing the width of the bounds on variables, the polytopes have decreasing volumes, while keeping the same relative size with respect to the others. Our results are that for 85% of the test instances the smallest volume are obtained with the relaxation corresponding to S_4. For a small percentage of instances (5%), these volumes are also obtained with (S_3 and (1) S_1 or with (1). S_2. We also find that for some instances the same volume is obtained for all four relaxations. For the remaining instances, the smallest values are obtained with S_3. We never find that S_1 and S_2 provide the lowest volumes.

Our computational results suggest that S_3 and S_4 always provide the best relaxations. That is, the best relaxations appear to be obtained employing convex envelopes for trilinear terms and not just bilinear ones. Our computational evidence for this apparent fact is confirmed by Theorem 1, reported in the next section.

We computationally get a further more precise information about tightness of the considered relaxations by checking relative containments of the corresponding (projected) polytopes. For each pair of polytopes P, Q, we check if P is contained in Q by checking that every extreme point of P satisfies all the inequalities defining Q. We find that, as expected, relaxation S_4 gives a polytope that is the most frequently contained in the others. In particular, it is always contained or equivalent to that corresponding to S_1 and S_2. The polytope corresponding to S_3 is also always contained in or equivalent to that given by S_1. This is particularly interesting, because S_1 is currently the most utilized in *sBB* implementations.

2.2 Mathematical comparison

The aim of this subsection is to provide a theoretical framework to investigate relaxation strength. We point out that it can be applied to convex relaxations of any mathematical program.

We define a language whose strings are the functions used in the objective and constraints of a mathematical program, and we define a semantic of strings of this language. To construct the language, we consider the alphabet $\mathscr{A} = \mathscr{X} \cup \mathbb{R} \cup \mathscr{O}$, where $\mathscr{X} = \{x_1, \dots, x_n\}$ is a set of variable symbols and $\mathscr{O}$ is a set of operator symbols. Let $\mathscr{L}$ be the class of strings built recursively in such a way that atomic expressions of a single variable or real number are in the language ($\forall \ell \in \mathbb{R} \cup \mathscr{X}\ (\ell \in \mathscr{L})$), and for every operator and potential arity, if the arity p is compatible with the operator, then by applying the operator to p (ordered) elements of the language, we get another element of the language ($\forall \otimes \in \mathscr{O}, p \in \omega\ (p \in \alpha(\otimes) \rightarrow \forall \ell_1 \dots, \ell_p \in \mathscr{L}\ (\otimes(\ell_1, \dots, \ell_p) \in \mathscr{L})$)).

We now introduce the formal definition of relaxed semantic of strings in $\mathscr{L}$. Let $x \in \mathbb{R}^n$ be such that $x^L \leq x \leq x^U$ for $x^L, x^U \in \mathbb{R}^n$, and let $f \in \mathscr{L}$. Consider the sets:

$$
\begin{aligned}
\mathscr{S}(f) &= \{(w_f, x) \mid w_f = f(x),\ x^L \leq x \leq x^U\}, \\
\mathscr{R}(f) &= \{(w_f, x) \mid A_f(w_f, x) \leq b_f,\ x^L \leq x \leq x^U\},
\end{aligned}
$$

where $b_f \in \mathbb{R}^m$ and $A_f \in \mathbb{R}^{m\times(n+1)}$ are such that $\mathscr{S}(f) \subseteq \mathscr{R}(f)$. We call $\mathscr{R}(f)$ the *relaxed semantic* of f.
We also consider a relaxed semantic over substrings. For all $i \leq p$ let $f_i, g, h \in \mathscr{L}$ be such that $h(x) = g(f_1(x),\ldots,f_p(x))$. Let $\mathbf{w}_f = (w_{f_1},\ldots,w_{f_p})$, $\mathbf{w} = (w_1,\ldots,w_p)$, and consider sets

$$
\begin{aligned}
\bar{R}(h) &= \{(w_g,\mathbf{w}_f,x) \mid A_g(w_g,\mathbf{w}_f) \leq b_g,\ A_{f_i}(w_{f_i},x) \leq b_{f_i}\ \forall i \leq p,\ x^L \leq x \leq x^U\}, \\
R(h) &= \{(w,x) \mid \exists \mathbf{w} \in \mathbb{R}^p \text{ such that } (w,\mathbf{w},x) \in \bar{R}\},
\end{aligned}
$$

where $R(h)$ is the projection of $\bar{R}(h)$ on the subspace $(w,x) \in \mathbb{R}^{n+1}$. $R(h)$ is the *relaxed composite semantic* of h with respect to its substring $g(f_1,\ldots,f_p)$.

We assume $\mathscr{R}(h) \subseteq R(h)$, i.e. the relaxed semantic is tighter than the relaxed composite semantic. Let $F \in \mathscr{L}$ and h such that $h(x) = g(f_1(x),\ldots,f_p(x))$. Let F' be F rewritten using the rule $g(f_1,\ldots,f_p) \to h$, i.e. using the alphabet $\mathscr{A}' = \mathscr{X} \cup \mathbb{R} \cup \mathscr{O}'$, where $\mathscr{O}' = \mathscr{O} \cup \{h\}$. The following theorem compares the strength of two relaxations.

Theorem 1. $R(F') \subseteq R(F)$.

Theorem 1 applied to relaxations of a quadrilinear term confirms the validity of some of the computational results reported in the previous section. In particular, it proves that among the relaxations that we considered, those utilizing trilinear envelopes (namely S_3 and S_4) always provided relaxations that are at least as tight as the other two (i.e., S_1 and S_2) For example, if we compare $(x_1x_2x_3)x_4$ and $((x_1x_2)x_3)x_4$, Theorem 1 ensures that the relaxed semantic of the former is at least as tight as the relaxed semantic of the latter (using the known convex envelopes for bilinear and trilinear terms).

3 The Molecular Distance Geometry Problem

We applied our results to the well-known Molecular Distance Geometry Problem (MDGP), whose main use is to find the three-dimensional structure of a molecule given a subset of the atomic distances [7]. Consider an undirected graph $G = (V,E)$ with weights $d : E \to \mathbb{R}_+$, where V is the set of vertices (also called *atoms*) and E is the set of weighted edges (also called *inter-atomic distances*). Let $d_{ij} = d(\{i,j\})$, for $\{i,j\} \in E$. A solution of the MDGP is a set of points $x_1,\ldots,x_{|V|} \in \mathbb{R}^3$ satisfying

$$||x_i - x_j||_2 = d_{ij}, \quad \forall \{i,j\} \in E. \tag{1}$$

The MDGP can be naturally cast as a continuous non-convex polynomial NLP with terms of degree up to four, by minimizing the sum of squared errors over the equations (1):

$$\min_{x\in\mathbb{R}^{|V|\times 3}} f(x) = \sum_{\{i,j\}\in E} (||x_i - x_j||_2^2 - d_{ij}^2)^2. \tag{2}$$

Note that (2) typically has a large number of local minima, so from a practical point of view, this is a hard global-optimization problem.

The natural application of tight lower bounds computed through a convex relaxation is within the sBB algorithm. In order to quickly assess the quality of our proposed alternative bound for quadrilinear terms on the MDGP without having to implement a full sBB framework, we implemented a simplified "partial sBB" algorithm which, at each branching step, only records the most promising node and discards the other, thus exploring a single branch up to a leaf. This corresponds to well-known "diving heuristics" employed in integer programming. At each node, a (linear) convex relaxation is constructed automatically by the `Rose` software [9] in the four different ways corresponding to the relaxations $S_1 - S_4$. It is then solved by `CPLEX` [5]. Table 3 shows the results obtained on four MDGP instances, that are randomly generated as described in [6]. We report the lower bounds obtained with the four relaxations. On all of the instances, the best lower bound is that obtained with a relaxation involving a trilinear envelope. In particular, S_4 gives the tightest bound for most cases, and this bound is significantly better than the values obtained with bilinear relaxations on the largest instance. This confirms the results of the previous sections and suggests that they can be used to configure a sBB algorithm to be efficiently applied to problems containing quadrilinear terms.

Instance	$((x_1x_2)x_3)x_4$	$(x_1x_2)(x_3x_4)$	$(x_1x_2x_3)x_4$	$(x_1x_2)x_3x_4$
`lavor6`	-1006.75	-1839.21	-1006.75	**-990.167**
`lavor7`	-1285.67	-1279.88	**-1175.95**	-1216.91
`lavor8`	-1711.27	-1694.56	-1718.41	**-1671.09**
`lavor10`	-3149.29	-3172.05	-3007.41	**-2755.04**

Table 1: Results obtained on MDGP instances.

4 Conclusion

We computationally and mathematically evaluated four linear relaxations of a quadrilinear term, showing that the tightest one can be obtained by combining the convex envelope of a trilinear term and that of a bilinear term. A more complete view of our computational results will appear in a full-paper version of this extended abstract. Our results can be exploited in a sBB algorithm to compute tight bounds. Our mathematical result can be applied to compare relaxations of more general problems.

Acknowledgements

We thank Komei Fukuda for helping us to use his code `CDD` to compute projections of polytopes and extreme point representations of polytopes given via inequalities. We thank Jesus De Loera for advising us to use David Avis' code `LRS`, and in turn we thank David Avis for `LRS`, which we used to compute exact volumes of polytopes given via extreme points. Financial support of the ANR grant 07-JCJC-0151 is gratefully acknowledged.

References

[1] C.S. Adjiman, S. Dallwig, C.A. Floudas, and A. Neumaier. A global optimization method, αBB, for general twice-differentiable constrained NLPs: I. Theoretical advances. *Computers & Chemical Engineering*, 22(9):1137–1158, 1998.

[2] D. Avis. `cgm.cs.mcgill.ca/~avis/C/lrs.html`.

[3] P. Belotti, F. Margot, J. Lee, A. Wächter, I. Grossmann, and L. Liberti. Branching and bounds tightening techniques for non-convex MINLP, IBM Research Report, 2008.

[4] K. Fukuda. `www.ifor.math.ethz.ch/~fukuda/ cdd_home/cdd.html`.

[5] ILOG. *ILOG CPLEX 11.0 User's Manual*. ILOG S.A., Gentilly, France, 2008.

[6] C. Lavor. On generating instances for the molecular distance geometry problem. In Liberti and Maculan ed., *Global Optimization: from Theory to Implementation*, Springer, Berlin, 2006, 405–414.

[7] C. Lavor, L. Liberti, and N. Maculan. Molecular distance geometry problem. In C. Floudas and P. Pardalos, editors, *Encyclopedia of Optimization*, pages 2305–2311. Springer, New York, 2 ed., 2008.

[8] J. Lee and W.D. Morris, Jr. Geometric comparison of combinatorial polytopes *Discrete Applied Mathematics*, 55(2):163 – 182, 1994.

[9] L. Liberti, S. Cafieri, and F. Tarissan. Reformulations in mathematical programming: A computational approach. In A.-E. Hassanien, A. Abraham, F. Herrera, W. Pedrycz, A. Carvalho, P. Siarry, and A. Engelbrecht, ed., *Foundations on Computational Intelligence*, Studies in Computational Intelligence. Springer, Berlin, to appear.

[10] L. Liberti, C. Lavor, M.A. Chaer Nascimento, and N. Maculan. Reformulation in mathematical programming: an application to quantum chemistry. *Discrete Applied Mathematics*, doi:10.1016/j.dam.2007.08.044.

[11] L. Liberti and C.C. Pantelides. Convex envelopes of monomials of odd degree. *Journal of Global Optimization*, 25:157–168, 2003.

[12] G.P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems. *Mathematical Programming*, 10:146–175, 1976.

[13] C.A. Meyer and C.A. Floudas. Trilinear monomials with mixed sign domains: Facets of the convex and concave envelopes. *Journal of Global Optimization*, 29:125–155, 2004.

[14] E.M.B. Smith. *On the Optimal Design of Continuous Processes*. PhD thesis, Imperial College of Science, Technology and Medicine, University of London, 1996.

[15] M. Tawarmalani and N. Sahinidis. Convex extensions and envelopes of semi-continuous functions. *Mathematical Programming*, 93(2):247–263, 2002.