Compact relaxations for polynomial programming problems

Sonia Cafieri, Pierre Hansen, Lucas Létocart, Leo Liberti, Frédéric Messine

Compact relaxations for polynomial programming problems*

Sonia Cafieri[1], Pierre Hansen[2,4], Lucas Létocart[3], Leo Liberti[4], and Frédéric Messine[5]

[1] Laboratoire MAIAA, Ecole Nationale de l'Aviation Civile, 7 av. E. Belin, 31055 Toulouse, France, sonia.cafieri@enac.fr
[2] GERAD, HEC Montreal, Canada, pierre.hansen@gerad.ca
[3] LIPN, Univ. de Paris Nord, Avenue J.B. Clément, 93430 Villetaneuse, France, lucas.letocart@lipn.univ-paris13.fr
[4] LIX, École Polytechnique, 91128 Palaiseau, France, liberti@lix.polytechnique.fr
[5] ENSEEIHT-IRIT, 2 rue Charles Camichel, BP 7122, F-31 071 Toulouse, France, frederic.messine@n7.fr

Abstract. Reduced RLT constraints are a special class of Reformulation-Linearization Technique (RLT) constraints. They apply to nonconvex (both continuous and mixed-integer) quadratic programming problems subject to systems of linear equality constraints. We present an extension to the general case of polynomial programming problems and discuss the derived convex relaxation. We then show how to perform rRLT constraint generation so as to reduce the number of inequality constraints in the relaxation, thereby making it more compact and faster to solve. We present some computational results validating our approach.
Keywords: polynomial, nonconvex, MINLP, sBB, reformulation, convex relaxation, RLT.

1 Introduction

We target Mixed-Integer Nonlinear Programming (MINLP) problems of the form:

$$\left.\begin{array}{rr} \min_x & f(x) \\ & g(x) \leq 0 \\ & Ax = b \\ & x^L \leq x \leq x^U \\ \forall\, i \in Z & x_i \in \mathbb{Z}, \end{array}\right\} \tag{1}$$

where $x, x^L, x^U \in \mathbb{R}^n$, $Z \subseteq \mathcal{N} = \{1, \ldots, n\}$, A is a full rank $m \times n$ matrix, $b \in \mathbb{R}^m$, $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^n \to \mathbb{R}^{m'}$ are polynomial functions of x.

We describe an extension to polynomial programming of an existing automatic reformulation technique [1–3] called reduced Reformulation-Linearization

* We are grateful to Dr. Tatjana Davidović and Christoph Dürr for useful discussions. Financial support by grants: Digiteo Chair 2009-14D "RMNCCO", Digiteo Emergence 2009-55D "ARM" is gratefully acknowledged.

Technique (rRLT). This technique was originally defined only for quadratic problems subject to linear equality constraints. It replaces some of the quadratic terms with suitable linear constraints, called rRLT constraints. These turn out to be a subset of the RLT constraints for quadratic programming [4]. The original RLT linearizes all quadratic terms in the problem and generates valid linear equation and inequality cuts by considering multiplications of bound factors (terms like $x_i - x_i^L$ and $x_i^U - x_i$) by linear constraint factors (the left hand side of a constraint such as $\sum_{j=1}^{n} a_j x_j - b \geq 0$ or $\sum_{j=1}^{n} a_j x_j - b = 0$). Since bound and constraint factors are always non-negative, so are their products: this way one can generate sets of valid problem constraints. An extension of the RLT to polynomial programming is described in [5], and to more general factorable programming problems in [6]. These results find their practical limitations in the extremely large number of adjoined constraints. Some heuristic techniques [4, 7] were proposed to help filter out RLT constraints which are redundant. In the rRLT the presence of linear equality constraints in the original problem allows the generation of only those linear RLT constraints that are guaranteed to replace a set of quadratic terms.

We aim to improve performance of spatial Branch-and-Bound (sBB) algorithms targeted at nonconvex NLPs and MINLPs; in particular, the rRLT tightens the lower bound computed by solving a convex relaxation of (1) at each sBB node. We make two original contributions. First, we extend rRLT theory from quadratic to polynomial programs. Second, as rRLT constraint generation depends on an arbitrary choice (the basis of a certain matrix) we show how to choose this basis in such a way as to yield a more compact (i.e., fewer constraints) convex relaxation, denoted by rRLT-C; the rRLT-C relaxation may be weaker than the rRLT one, but experiments show that the loss in tightness is greatly offset by the gain in CPU time taken to solve it. We assume our polynomial programs to be dense up to Sect. 3.2 for simplicity, and deal with sparsity in Sect. 4.

Notationwise, we deal sometimes with indexed symbols which are scalars and indexed symbols which are vectors; in order to avoid ambiguities, we denote with boldface all indexed symbols indicating vectors. For example, w_{ij} is a scalar but $\mathbf{w}_{ij}$ is the vector $(w_{ij1}, \ldots, w_{ijn})$; in line with current optimization literature, we do not use boldface to indicate the name of a whole array, so w might be either a scalar or an array depending on the context.

The rest of this paper is organized as follows: Sect. 2 extends rRLT to polynomial programming. In Sect. 3 we discuss how to construct the rRLT-C compact relaxation. In Sect. 4 we address sparse polynomial programs. Sect. 5 discusses some computational experiments on randomly generated instances.

2 rRLT for Polynomial Programming

The results presented herein extend [1] to the general polynomial case. Let $Q = \{2, \ldots, q\}$. For each monomial $x_{j_1} \cdots x_{j_p}$ appearing in the original problem (1) where $p \in Q$, we define a finite sequence $J = (j_1, \ldots, j_p)$ and, consistent with the

notation introduced by Sherali [5], consider defining constraints of the following form:

$$w_J = \prod_{\ell \le |J|} x_{j_\ell} \tag{2}$$

(for $|J| = 1$, i.e. $J = (j)$, we also define $w_J = x_j$). For all $p \in Q$, $J \in \mathcal{N}^p$ and any permutation π in the symmetric group S_p we have that $w_J = w_{\pi J}$ by commutativity. We therefore define an equivalence relation $\sim$ on $\mathcal{N}^p$ stating that for $J, K \in \mathcal{N}^p$, $J \sim K$ only if $\exists \pi \in S_p$ such that $J = \pi K$. We then consider the set of equivalence classes $\bar{\mathcal{N}}^p = \mathcal{N}^p/\!\sim$ to quantify over when indexing added variables w_J. In practice, we choose an equivalence class representative for each $J \in \bar{\mathcal{N}}^p$ which we also denote by J. With a slight abuse of notation, if $J' \in \bar{\mathcal{N}}^{p'}$ and $J'' \in \bar{\mathcal{N}}^{p''}$ such that $p' + p'' = p$ and (J', J'') is in the equivalence class represented by $J \in \bar{\mathcal{N}}^p$, we write $(J', J'') = J$. We also define, for all $p \in Q$, $\mathcal{M}_p = \bigcup_{1<p' \le p} \bar{\mathcal{N}}^{p'}$ and $\mathcal{M}_p^1 = \bigcup_{p' \le p-1} \bar{\mathcal{N}}^{p'}$.

We multiply the original linear constraints $Ax = b$ by all monomials $\prod_{\ell \le p-1} x_{j_\ell}$ and replace them by the corresponding added variables $w_{(J',j)}$, where $J' \in \bar{\mathcal{N}}^{p-1}$. This yields the following rRLTS:

$$\forall p \in Q, J' \in \bar{\mathcal{N}}^{p-1} \quad A\, \mathbf{w}_{J'} = b w_{J'}, \tag{3}$$

where $\mathbf{w}_{J'} = (w_{(J',1)}, \ldots, w_{(J',n)})$. We then consider the companion system:

$$\forall p \in Q, J' \in \bar{\mathcal{N}}^{p-1} \quad A\, \mathbf{z}_{J'} = 0. \tag{4}$$

Since (4) is a linear homogeneous system, there is a matrix M such that the companion system is equivalent to $Mz = 0$, the columns of which are indexed by sequences in $\mathcal{M}_p$. We let $B \subseteq \mathcal{M}_p$ and $N \subseteq \mathcal{M}_p$ be index sets for basic and nonbasic columns of M. We define the following sets:

$$C = \{(x, w) \mid Ax = b \wedge \forall p \in Q, J \in \bar{\mathcal{N}}^p (w_J = \prod_{\ell \le |J|} x_{j_\ell})\} \tag{5}$$

$$R_N = \{(x, w) \mid Ax = b \wedge \forall p \in Q, J' \in \bar{\mathcal{N}}^{p-1} (A\, \mathbf{w}_{J'} = b w_{J'}) \wedge \forall J \in N (w_J = \prod_{\ell \le |J|} x_{j_\ell})\}. \tag{6}$$

Theorem 2.1. *For each partition B, N into basic and nonbasic column indices for the companion system $Mz = 0$, we have $C = R_N$.*

We remark that for this proof to hold, all possible nonlinear monomials must be present in the problem, which is generally not the case. We address this problem in Sect. 4. A different treatment of the essentially the same concepts, which only employs a bases of A instead of the (larger) companion system, was given in [8].

Replacing C with R_N for some nonbasis N effectively replaces some nonlinear monomial terms with linear constraints, and therefore contributes to simplify the problem. A convex relaxation for the reformulated problem is readily obtained by applying monomial convexification methods in the literature .

3 Compact convex relaxation

First, we remark that virtually no practical polynomial problem exhibits *all* possible nonlinear monomials. Let $\mathcal{M} = \mathcal{M}_n$ and $\mathcal{M}^1 = \mathcal{M}_n^1$. We introduce two sets: $\beta \subseteq \mathcal{M}$ indexing all nonlinear monomials appearing in the original problem (1) and $\beta' \subseteq \mathcal{M}$ indexing all nonlinear monomials appearing in (1) *and* the rRLTS (3). Reduced RLT constraints are likely to give rise to compact yet tight convex relaxations if $N \subsetneq \beta$, in view of the fact that, by Thm. 2.1, only monomials indexed by N need appear in the formulation — so the lower and upper relaxations to monomials outside N can be dropped. Furthermore, the proof of Thm. 2.1 also implies that the number of monomials that can be replaced is equal to the rank ρ of the rRLTS. If the original problem has few monomials, N might not be a proper subset of β, and in practice this occurrence is not rare. Limited to quadratic polynomials, we address this problem in [2]; in Sect. 4 we propose a technique to deal with sparse polynomial programs.

In this section we discuss a choice of N whereby the monomial relaxations that are dropped define "large volumes", and are therefore more likely to be dominated by the relaxations of monomials in N. Intuitively, this should yield a compact relaxation whose bound is not far from the normal rRLT relaxation.

3.1 Convexity gap

Definition 3.1. *Consider a function $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$. Let $\underline{f}(x)$ be a convex lower bounding function for f and $\bar{f}(x)$ be a concave upper bounding function for f. Then the set $\bar{S} = \{(x, w) \mid \underline{f}(x) \leq w \leq \bar{f}(x)\}$ is a convex relaxation of the set $S = \{(x, w) \mid w = f(x)\}$. We define the convexity gap $V(S)$ between S and $\bar{S}$ to be the volume of the set $\bar{S}$; namely,*

$$V(S) = \int_{x \in X} (\bar{f}(x) - \underline{f}(x))dx. \tag{7}$$

We denote the convexity gap for a quadratic term $x_i x_j$ with V_{ij}.

Convexity gap for a quadratic term x_i^2. The convex envelope of the set $\xi = \{(x_i, w_{ii}) \mid w_{ii} = x_i^2, x_i^L \leq x_i \leq x_i^U\}$ (where $i \leq n$) consists of the area between the function x_i^2 and the chord. The convexity gap of ξ is:

$$V_{ii} = \int_{x_i^L}^{x_i^U} \left((x_i^L)^2 + \frac{(x_i^U)^2 - (x_i^L)^2}{x_i^U - x_i^L}(x_i - x_i^L) - x_i^2 \right) dx = \frac{1}{6}(x_i^U - x_i^L)^3. \tag{8}$$

Convexity gap for a bilinear term $x_i x_j$. For all $i < j \leq n$, the convex envelope of the set $\{(x_i, x_j, w_{ij}) \mid w_{ij} = x_i x_j, x_i^L \leq x_i \leq x_i^U, x_j^L \leq x_j \leq x_j^U\}$ is a tetrahedron Δ in $\mathbb{R}^3$ given by the McCormick inequalities [9, 10]. The vertices

of Δ are: $(x_i^U, x_j^U, x_i^U x_j^U)$, $(x_i^U, x_j^L, x_i^U x_j^L)$, $(x_i^L, x_j^U, x_i^L x_j^U)$, $(x_i^L, x_j^L, x_i^L x_j^L)$. Let

$$\begin{aligned}
\mu &= x_i^L - x_i^U \\
\nu &= x_j^L - x_j^U \\
\xi &= x_i^L x_j^L - x_i^U x_j^U \\
\zeta &= x_i^L x_j^U - x_i^U x_j^L \\
\hat{D} &= \begin{vmatrix}
0 & 1 & 1 & 1 & 1 \\
1 & 0 & \nu^2(1+a^2) & \mu^2(1+c^2) & \mu^2+\nu^2+\xi^2 \\
1 & \nu^2(1+a^2) & 0 & \mu^2+\nu^2+\zeta^2 & \mu^2(1+d^2) \\
1 & \mu^2(1+c^2) & \mu^2+\nu^2+\zeta^2 & 0 & \nu^2(1+b^2) \\
1 & \mu^2+\nu^2+\xi^2 & \mu^2(1+d^2) & \nu^2(1+b^2) & 0
\end{vmatrix}.
\end{aligned}$$

The volume V_{ij} of Δ can be computed using the Cayley-Menger formula in 3 dimensions [11], i.e. $V_{ij} = (x_i^L, x_i^U, x_j^L, x_j^U) = \frac{\sqrt{2\hat{D}}}{24}$.

Convexity gap for a multilinear monomial. By [12], the convex under- and over-approximating envelopes of a multilinear monomial $x_1 \cdots x_p$ of degree p, are polyhedral. Therefore, the facet defining inequalities of the enveloping polytope can be computed using the 2^p polytope vertices $v^\ell = (x_1^{\ell_1}, \ldots, x_p^{\ell_p}, \prod_{j \leq p} x_j^{\ell_j})$ in $\mathbb{R}^{n+1}$, for every p-sequence $\ell \in \{L, U\}^p$. We carry out this computation using the Porta software [13]: although its worst-case complexity is exponential in p, it is practically efficient for low values of p. Since our computational experiments only address problems up to degree 4, this methodological choice is appropriate. Standard methods to compute volumes of polytopes exist [14]. Our preliminary implementation uses the volume of the smallest bounding box — this will be changed later.

If J is the (ordered) sequence of p variable indices appearing in a multilinear monomial $\mu(x)$, we let V_J denote the convexity gap for $\mu(x)$.

Convexity gap for a general monomial. By associativity, for any $p \in Q$ and a given sequence $J \in \mathcal{N}^p$ it is always possible to express the monomial $\mu(x) = \prod_{\ell \leq p} x_{j_\ell}$ as a product of multilinear factors $(x_{j_1} \cdots x_{j_2}) \cdots (x_{j_3} \cdots x_{j_4})$. After replacement by the appropriate added variable, the monomial is reduced to $w_{J_1} \cdots w_{J_2}$, where $J_1 = \{j_1, \ldots, j_2\}$ and $J_2 = \{j_3, \ldots, j_4\}$. Associativity can then be re-applied recursively. This allows us to use the results of the preceding sections to derive a convexity gap for $\mu(x)$. This approach, albeit simple, is similar, up to commutativity, to the standard reformulation exploited by sBB implementations [15–19] in view to obtain the convex relaxation of general monomials, and more specifically to the approach followed in [20] for quadrilinear monomials. Explicit monomial convex envelopes are also known for trilinear terms [21, 22] and univariate terms of odd degree [23].

Since we already treated the multilinear case separately, we assume that $\mu(x)$ is *not* multilinear. In such cases, the recursive techniques described above yield

nonlinear convex relaxations. Because our computational results only refer to *linear* relaxations, however, we take a simpler approach and use interval arithmetic [24] to compute an interval range $[\mu^L, \mu^U]$ such that $\mu^L \leq \mu(x) \leq \mu^U$ for all $x \in [x^L, x^U]$. If J is the (ordered) sequence of variable indices appearing in the monomial $\mu(x)$, we compute V_J as the volume of the bounding box $[x^L, x^U] \times [\mu^L, \mu^U]$.

3.2 Choosing a good basis for the companion system

Let B, N be the basic/nonbasic sets of column indices of the companion system (4), which we write in this section as $Mz = 0$, or, equivalently, as $M_B z_B + M_N z_N = 0$. As shown in Sect. 2, the elements of B, N are sequences $J \in \mathcal{M}$. For $S \subseteq \mathcal{M}$ and $p \in Q$ we define $V^{S,p} = \sum_{\substack{J \in S \\ |J|=p}} V_J$ and $V^S = \sum_{p \in Q} V^{S,p}$. If, for all $p \in Q$, $V^{N,p} < V^{\beta,p}$ then the total convexity gap of R_N is smaller than that of C. Thus, we aim to find N such that $V^{N,p}$ is minimized, or equivalently, to find B such that $V^{B,p}$ is maximized for all $p \in Q$. This yields the multi-objective problem:

$$\left.\begin{array}{c} \forall p \in Q \quad \max V^{B,p} \\ M_B \text{ is a basis of (4)} \end{array}\right\} \tag{9}$$

Next, we show that (9) is equivalent to a single-objective problem.

Consider a block diagonal $\bar{m} \times \bar{n}$ matrix $\bar{A}$ with r blocks A_s (each a rectangular, full-rank $m_s \times n_s$ matrix for each $s \leq r$) having a basis indexed by the set $\bar{B} \subseteq \{1, \ldots, \bar{n}\}$. For all $s \leq r$ let α_s be the set of column indices corresponding to the submatrix A_s of $\bar{A}$. For a matrix T let $\mathrm{span}(T)$ be the space spanned by the columns of T.

Lemma 3.2. *Let $i, j \leq \bar{n}$ be such that $i \in \bar{B}, j \notin \bar{B}$ and π be the swap (i, j). If $i \in \alpha_q$ and $j \in \alpha_t$ with $q \neq t$ then the columns of $\bar{A}$ indexed by $\pi\bar{B}$ do not form a basis.*

We remark that M is a block-diagonal matrix. Instead of showing a formal proof of this fact, which would be long and tedious, we exhibit an example for the case of polynomials of degree at most 3. Example 3.3 does not exploit any specific property of the given matrix, and therefore appropriately illustrates what happens in the general case.

Example 3.3. The companion system (4) $Mz = 0$ derived from the system $Ax = b$ with $A = \begin{pmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{pmatrix}$, $x \in \mathbb{R}^3$ with $Q = \{2, 3\}$ is:

$$\left(\begin{array}{cccccccccc}
a_1\ a_2\ a_3 & & & & & & & & & \\
a_4\ a_5\ a_6 & & & & & & & & & \\
a_1 & & a_2\ a_3 & & & & & & & \\
a_4 & & a_5\ a_6 & & & & & & & \\
 & a_1 & & a_2\ a_3 & & & & & & \\
 & a_4 & & a_5\ a_6 & & & & & & \\
\hline
 & & & & a_1\ a_2\ a_3 & & & & & \\
 & & & & a_4\ a_5\ a_6 & & & & & \\
 & & & & a_1 & & a_2\ a_3 & & & \\
 & & & & a_4 & & a_5\ a_6 & & & \\
 & & & & & a_1 & & a_2\ a_3 & & \\
 & & & & & a_4 & & a_5\ a_6 & & \\
 & & & & & & a_1 & & a_2\ a_3 & \\
 & & & & & & a_4 & & a_5\ a_6 & \\
 & & & & & & & a_1 & & a_2\ a_3 \\
 & & & & & & & a_4 & & a_5\ a_6
\end{array}\right)
\left(\begin{array}{c} z_{11} \\ z_{12} \\ z_{13} \\ z_{22} \\ z_{23} \\ z_{33} \\ z_{111} \\ z_{112} \\ z_{113} \\ z_{122} \\ z_{123} \\ z_{133} \\ z_{222} \\ z_{223} \\ z_{233} \\ z_{333} \end{array}\right) = 0$$

The number of monomials of n variables of degree exactly p is given in [25] as $\binom{p+n-1}{p}$. From this, it follows that M has $m \sum_{p \in Q} \binom{p+n-2}{p-1}$ rows and $\sum_{p \in Q} \binom{p+n-1}{p}$ columns.

Theorem 3.4. *Any solution B of (9) maximizing V^B also maximizes $V^{B,p}$ for all $p \in Q$.*

The single-objective problem $\max\{v^B \mid M_B \text{ a basis of (4)}\}$ has a matroidal structure and can therefore be solved using a greedy algorithm.

Evidently, the technique based on compact rRLT constraints is not significant whenever the bounds are the same across all decision variables, because in this case all the V_J's are equal. This, however, is rarely true if bound tightening techniques [26, 18, 19, 27] are used as a preprocessing step; and it is *never* true during sBB with rectangular partitioning schemes, as the variable ranges are partitioned at each node. We also emphasize that the remark given in [8] p. 11 is also valid in our setting: for polynomial degrees ≥ 3, the proposed convex relaxation might not be monotonically increasing w.r.t. following branches of the sBB tree down from the root, thus preventing the sBB from converging. This can be fixed at each node by adjoining those bound factor inequalities (derived from the multiplication of different bound factors) that involve the branching variable.

4 Dealing with sparsity

Polynomial problems are rarely dense; this might prevent the set N, introduced in Sect. 2 to index the added variables corresponding to nonbasic columns of

the companion system (4), from being a subset of β, the set indexing the added variables corresponding to all monomials appearing in (1). We deal with this possibility by looking for a subset ρ of rows of $Ax = b$ to be multiplied by a subset σ of added variables indexed by $\mathcal{M}^1$. The rRLTS (3) and its companion system (4) are derived in the same way as for dense polynomial problems.

Theoretically, we should require $N \subsetneq \beta$: as remarked in Sect. 3, the rRLTS is then likely to give rise to compact yet tight convex relaxations. In practice, in sparse problems, this requirement will often force ρ or σ to be empty, which means that the reformulation does not take place. Following the principle that every equation corresponds to one fewer degree of freedom, we aim to find ρ, σ such that the size of the sparse rRLTS obtained by multiplying rows in ρ by added variables in σ exceeds the number of new monomials (i.e. monomials not in β) generated by these multiplications. This flexibility in the choice of ρ, σ has a trade-off in terms of relaxation compactness. In order for Thm. 2.1 to hold, the quantifications $J \in \bar{\mathcal{N}}^p$ in (5) and $J' \in \bar{\mathcal{N}}^{p-1}$ in (6) should be replaced by $J \in \tau$, $J' \in \sigma$ respectively, where $\tau = \{(J', j) \mid J' \in \sigma\ (|J'| = p) \wedge j \in \mathcal{N}\}$. Since $\sigma \subseteq \mathcal{M}^1$, fewer monomials are replaced by the rRLTS, yielding a relaxation which might not be much more compact than the standard RLT relaxation.

We formalize the problem of finding suitable ρ, σ by considering a bipartite graph that represents the incidence of monomials indexed by $\mathcal{M}$ in products of rows $a_i x = b_i$ of $Ax = b$ multiplied by monomials indexed by $\mathcal{M}^1$. Let $U = \{(i, J') \mid i \leq m \wedge J' \in \mathcal{M}^1\}$ and define a set E such that, for $(i, J') \in U$ and $J \in \mathcal{M} \smallsetminus \beta$, $\{(i, J'), J\} \in E$ if there is $j \in \mathcal{N}$ such that $a_{ij} \neq 0$ and $(J', j) = J$. Consider the bipartite graph $G = (U, \mathcal{M} \smallsetminus \beta, E)$: we want to find an induced subgraph $G' = (U', V', E')$ of G, with $U' \subseteq U$ and $V' \subseteq \mathcal{M} \smallsetminus \beta$, such that: $|U'|$ is maximum, $|U'| > |V'|$ and $V' = N_E(U') = \{v \in \mathcal{M} \smallsetminus \beta \mid \exists u \in U'\ (\{u, v\} \in E)\}$. We define this problem using a Mathematical Programming (MP) formulation: we employ binary variables $u_{i,J'}$ for all $(i, J') \in U$ and v_J for all $J \in \mathcal{M} \smallsetminus \beta$:

$$\left.\begin{array}{rr} \max \sum\limits_{(i,J') \in U} u_{i,J'} & \\ \sum\limits_{(i,J') \in U} u_{i,J'} \geq \sum\limits_{J \notin \beta} v_J + 1 & \\ \forall \{(i, J'), J\} \in E \quad v_J \geq u_{i,J'} & \\ u \in \{0,1\}^{|U|} & \\ v \in \{0,1\}^{|\mathcal{M} \smallsetminus \beta|}. & \end{array}\right\} \tag{10}$$

We can then define $\rho = \{i \leq m \mid \exists J' \in \mathcal{M}^1\ (u_{i,J'} = 1)\}$ and $\sigma = \{J' \in \mathcal{M}^1 \mid \exists i \leq m\ (u_{i,J'} = 1)\}$. This is a Binary Linear Program (BLP), so it cannot be solved in polynomial time with standard MILP technology (i.e. using a Branch-and-Bound algorithm).

4.1 Bipartite matching based algorithm for (10)

We propose an efficient algorithm based on bipartite matching for solving (10). With respect to a matching M of G, a vertex of G is *exposed* if it is incident to an edge which is not in M. For a subset of edges $F \subseteq E$, let $\mathsf{U}(F) = \{u \in U \mid \exists e \in$

$F\ (u \in e)\}$ and $\mathsf{V}(F) = \{v \in \mathcal{M} \smallsetminus \beta \mid \exists e \in F\ (v \in e)\}$. If M is a maximum matching of G such that there exists an exposed vertex $u \in U \smallsetminus \mathsf{U}(M)$, the search for an augmenting path from u which is alternating with respect to M will fail by maximality of M [28]. A *dilation* rooted in u with respect to a maximum matching M is a maximal simple alternating path p_u (seen as a sequence of edges in E) in G, from u to a vertex $u' \in \mathsf{U}(M)$, whose even-indexed edges are in M and odd-indexed edges are in $E \smallsetminus M$. Dilations are the certificates used in cardinality bipartite matching algorithms to prove that the current matching is optimal; informally, they certify the failure to find an augmenting alternating path to increase the cardinality of the current matching in the classical bipartite matching algorithm (see Fig. 10-3 in [29]).

Lemma 4.1. *For a dilation p_u from u in G w.r.t. a maximum matching M, we have $|\mathsf{U}(p_u)| = |\mathsf{V}(p_u)| + 1$.*

Lemma 4.2. *If p, p' are different dilations in E, then $|\mathsf{U}(p \cup p')| > |\mathsf{V}(p \cup p')|$.*

In order to deal with the case of isolated vertices, if $u \in U$ is isolated, then it is exposed w.r.t. the empty matching, and any empty path p_u rooted at u is a dilation; in this case, with a slight abuse of notation, we define $\mathsf{U}(p_u) = \{u\}$. A *dilation set* is the set of edges in all dilations rooted at u with respect to M; dilation sets can be found in polynomial time using breadth first search (bfs) from u (see [29], Sect. 10.2). By Lemma 4.2 above, dilation sets P are such that $|\mathsf{U}(P)| > |\mathsf{V}(P)|$.

Algorithm 1 Matching-based algorithm for solving (10).

Require: A bipartite graph $G = (U, \mathcal{M} \smallsetminus \beta, E)$
Ensure: A nontrivial subgraph $G' = (U', V', E)$ solving (10), or $\varnothing$ if none exists
Let $G' = \varnothing$
while $|U| > 0$ **do**
  Let $M \subseteq E$ be a maximum matching in G
  if $|\mathsf{U}(M)| < |U|$ **then**
    Find $u \in U$ exposed and a corresponding dilation set P_u
    Let $H = (\mathsf{U}(P_u), \mathsf{V}(P_u), P_u)$
    Update $G' \leftarrow G' \cup H$
    Update $G \leftarrow G \smallsetminus H$
  else
    break
  end if
end while
return G'

Proposition 4.3. *In polynomial time, Alg. 1 finds a subgraph $G' = (U', V', E')$ of G such that $|U'|$ is maximum, $|U'| > |V'|$ and $V' = N_E(U')$, or determines that no such subgraph exists.*

5 Computational results

The best practical indication of the importance of rRLT techniques for linear equality constrained polynomial programming is best observed by the computational results given in [2], where the presence or absence of rRLT yields differences of up to five orders of magnitude on a class of pooling problems (these are sparse polynomial problems of degree 2) from the oil industry. Those results are limited to quadratic programs and employ a restricted version of the results for sparsity given in Sect. 4. We are currently in the process of extending our code to deal with polynomial programming, so in this paper we only show empirically that the compact rRLT-C formulation generally takes less time to solve and yields bounds that are not much worse than those given by the rRLT formulation.

We generate two sets of random polynomial programming instances of degree 2,3 and 4 with varying numbers of variables and linear equality constraints (all the monomials are in the objective function, weighted by random scalars). Set 1 consists of 10 instances of degrees 2 and 3 with random variable ranges whose widths follow a Gaussian distribution. Set 2 consists of 8 instances of degrees 3 and 4 with random variable ranges whose widths follow a superposition of two Gaussian distributions ($n/2$ ranges have width of order 1, the other $n/2$ have width of order 10); set 2 is designed to simulate the typical sBB node after a few levels of branching, where some of the variable ranges have become small whereas others are still at their original bounds.

For all these instances we construct and solve a linear relaxation with no rRLT constraints (column labeled "simple" in Table 1), the rRLT linear relaxation (column labeled "rRLT") and the rRLT-C linear relaxation (column labeled "rRLT-C"). We recall that rRLT-C is like rRLT without the constraints relaxing monomial terms corresponding of basic columns of the companion system (4). We record bound value and CPU time. Notice some of the generated instances are infeasible: this is consistent with the fact that in a typical sBB search tree some of the nodes represent infeasible subproblems. Since the infeasibility is determined by the linear relaxation, performance on infeasible LPs is also an important factor. These results were obtained using CPLEX 11 [30] on a Pentium Xeon 2.4GHz CPU with 8 GB of RAM running Linux. The results in Table 1 show that rRLT brings considerable benefits to bound tightness within polynomial optimization, and that the (significant) CPU time reduction yielded by rRLT-C is not offset by an excessive loss in bound quality with respect to rRLT: **the cumulative bound worsening is 0.07% against a time improvement of nearly 40%**. The CPU time taken by the simple relaxation is of course much lower than those of the rRLT relaxations, but already for the quadratic case it was shown in [2] that this time difference is not sufficient to offset the benefits of the bound improvement — hence the corresponding values do not appear in Table 1. We remark that rRLT-C does not always yield better CPU time results. This is simply because the relationship between CPU time and number of constraints in solving an LP is neither regular nor monotonic.

instance					simple	rRLT		rRLT-C	
name	p	n	m	q	*bound*	*bound*	*time*	*bound*	*time*
1	1	20	10	2	2864.8	**2982.63**	0.20	**2982.63**	**0.17**
2	1	30	10	2	5286.23	**5517.85**	2.56	**5517.85**	**1.29**
3	1	10	3	3	48.6115	**478.184**	**0.25**	**478.184**	0.30
4	1	10	3	3	infeas	infeas	0.23	infeas	**0.22**
5	1	20	3	3	infeas	infeas	14.81	infeas	**12.92**
6	1	10	5	3	infeas	infeas	0.11	infeas	**0.10**
7	1	10	5	3	infeas	infeas	0.16	infeas	0.16
8	1	20	5	3	infeas	infeas	**29.01**	infeas	31.41
9	1	10	7	3	infeas	infeas	0.19	infeas	**0.15**
10	1	20	7	3	infeas	infeas	65.22	infeas	**46.98**
11	2	10	3	3	130.693	**1546.4**	0.20	1542.39	**0.13**
12	2	10	3	3	18.8459	**772.417**	0.09	**772.417**	0.09
13	2	15	5	3	17.3797	**701.723**	**10.13**	701.588	11.03
14	2	16	8	3	infeas	infeas	488.05	infeas	**287.38**
15	2	7	2	4	infeas	infeas	0.17	infeas	0.17
16	2	8	3	4	47.4445	**3468.56**	1.26	3458.4	**0.85**
17	2	10	3	4	26.2698	**4038.69**	250.88	**4038.68**	**131.95**
18	2	12	3	4	56.9232	**13127.5**	166.62	13118.6	**109.95**

Table 1. Comparing bound strength and CPU time for linear relaxations.

References

1. Liberti, L.: Linearity embedded in nonconvex programs. Journal of Global Optimization **33**(2) (2005) 157–196
2. Liberti, L., Pantelides, C.: An exact reformulation algorithm for large nonconvex NLPs involving bilinear terms. Journal of Global Optimization **36** (2006) 161–189
3. Liberti, L.: Compact linearization of binary quadratic problems. 4OR **5**(3) (2007) 231–245
4. Sherali, H., Alameddine, A.: A new reformulation-linearization technique for bilinear programming problems. Journal of Global Optimization **2** (1992) 379–410
5. Sherali, H., Tuncbilek, C.: A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique. Journal of Global Optimization **2** (1991) 101–112
6. Sherali, H., Wang, H.: Global optimization of nonconvex factorable programming problems. Mathematical Programming **89** (2001) 459–478
7. Sherali, H., Tuncbilek, C.: New reformulation linearization/convexification relaxations for univariate and multivariate polynomial programming problems. Operations Research Letters **21** (1997) 1–9
8. Sherali, H., Dalkiran, E., Liberti, L.: Reduced rlt representations for nonconvex polynomial programming problems. Journal of Global Optimization (submitted)
9. McCormick, G.: Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems. Mathematical Programming **10** (1976) 146–175
10. Al-Khayyal, F., Falk, J.: Jointly constrained biconvex programming. Mathematics of Operations Research **8**(2) (1983) 273–286

11. Jiao, Y., Stillinger, F., Torquato, S.: Geometrical ambiguity of pair statistics I. point configurations. Technical Report 0908.1366v1, arXiv (2009)
12. Rikun, A.: A convex envelope formula for multilinear functions. Journal of Global Optimization **10**(4) (1997) 425–437
13. Christof, T., Löbel, A.: The porta manual page. Technical Report v. 1.4.0, ZIB, Berlin (1997)
14. D. Avis: User's Guide for `lrs`. (2009)
15. Smith, E., Pantelides, C.: A symbolic reformulation/spatial branch-and-bound algorithm for the global optimisation of nonconvex MINLPs. Computers & Chemical Engineering **23** (1999) 457–478
16. Adjiman, C., Dallwig, S., Floudas, C., Neumaier, A.: A global optimization method, αBB, for general twice-differentiable constrained NLPs: I. Theoretical advances. Computers & Chemical Engineering **22**(9) (1998) 1137–1158
17. Sahinidis, N., Tawarmalani, M.: BARON 7.2.5: Global Optimization of Mixed-Integer Nonlinear Programs, *User's Manual*. (2005)
18. Liberti, L.: Writing global optimization software. In Liberti, L., Maculan, N., eds.: Global Optimization: from Theory to Implementation. Springer, Berlin (2006) 211–262
19. Belotti, P., Lee, J., Liberti, L., Margot, F., Wächter, A.: Branching and bounds tightening techniques for non-convex MINLP. Optimization Methods and Software **24**(4) (2009) 597–634
20. Cafieri, S., Lee, J., Liberti, L.: On convex relaxations of quadrilinear terms. Journal of Global Optimization **47** (2010) 661–685
21. Meyer, C., Floudas, C.: Trilinear monomials with positive or negative domains: Facet s of the convex and concave envelopes. In Floudas, C., Pardalos, P., eds.: Frontiers in Global Optimization. Kluwer Academic Publishers, Amsterdam (2003) 327–352
22. Meyer, C., Floudas, C.: Trilinear monomials with mixed sign domains: Facets of the convex and concave envelopes. Journal of Global Optimization **29** (2004) 125–155
23. Liberti, L., Pantelides, C.: Convex envelopes of monomials of odd degree. Journal of Global Optimization **25** (2003) 157–168
24. Moore, R., Kearfott, R., Cloud, M.: Introduction to Interval Analysis. SIAM, Philadelphia (2009)
25. Katzman, M.: Counting monomials. Journal of Algebraic Combinatorics **22** (2005) 331–341
26. Messine, F., Lagouanelle, J.: Enclosure methods for multivariate differentiable functions and application to global optimization. Journal of Universal Computer Science **4**(6) (1998) 589–603
27. Belotti, P., Cafieri, S., Lee, J., Liberti, L.: Feasibility-based bounds tightening via fixed points. In Du, D.Z., Pardalos, P., Thuraisingham, B., eds.: Combinatorial Optimization, Constraints and Applications (COCOA10). Volume 6508 of LNCS., New York, Springer (2010) 65–76
28. Berge, C.: Two theorems in graph theory. Proceedings of the National Academy of Science of the U.S. **43** (1957) 842–844
29. Papadimitriou, C., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Dover, New York (1998)
30. ILOG: ILOG CPLEX 11.0 User's Manual. ILOG S.A., Gentilly, France. (2008)

A Appendix: Proofs

Theorem A.1 (Thm. 2.1). *For each partition B, N into basic and nonbasic column indices for the companion system $Mz = 0$, we have $C = R_N$.*

Proof. Since (3) is implied by $Ax = b$ and $N \subseteq \mathcal{M}_p$, it follows that $C \subseteq R_N$. Conversely, for all $p \in Q$ and $J \in \bar{\mathcal{N}}^p$ we define:

$$z_J = w_J - \prod_{\ell \leq p} x_{j_\ell}. \tag{11}$$

The definition of R_N implies (3). We consider the companion system (4): since B, N form a partition of the columns of $Mz = 0$ we can write the latter as $M_B z_B + M_N z_N = 0$. By (11) and the fact that the definition of R_N imposes the condition $\forall J \in N(w_J = \prod_{\ell \leq |J|} x_{j_\ell})$, it follows that $\forall J \in N(z_J = 0)$, and hence $M_N z_N = 0$. Since B is a basic index set, we have $M_B z_B = 0$; since M_B is invertible, $Mz = 0$ has the unique solution $z = 0$. This, in turn, again by (11), implies $\forall J \in \bar{\mathcal{N}}^p(w_J = \prod_{\ell \leq |J|} x_{j_\ell})$ as in the definition of C. Hence, the conditions defining R_N imply the conditions defining C. □

Lemma A.2 (Lemma 3.2). *Let $i, j \leq \bar{n}$ be such that $i \in \bar{B}, j \notin \bar{B}$ and π be the swap (i, j). If $i \in \alpha_q$ and $j \in \alpha_t$ with $q \neq t$ then the columns of $\bar{A}$ indexed by $\pi\bar{B}$ do not form a basis.*

Proof. Let A' be the (square) $\bar{m} \times \bar{m}$ invertible matrix corresponding to the basis indexed by $\bar{B}$. Since $\mathrm{span}(\bar{A}) = \mathrm{span}(A')$ and $\bar{B} \subseteq \{1, \ldots, \bar{n}\}$, A' is a block-diagonal matrix with r full-rank, square $m_s \times m_s$ blocks A'_s such that $\mathrm{span}(A_s) = \mathrm{span}(A'_s)$ for all $s \leq r$, and such that $\mathrm{span}(A') = \bigoplus_{s \leq r} \mathrm{span}(A'_s)$. Suppose $\pi = (i, j)$ where $i \in \alpha_q$, $j \in \alpha_t$ for some $q \neq t \leq r$; let A''_q be the matrix A'_q without column i, and let A''_t be the matrix A'_t with column j adjoined. Since A'_q, A'_t are square, A''_q, A''_t are rectangular with $\mathrm{span}(A''_q) \subsetneq \mathrm{span}(A'_q)$ and $\mathrm{span}(A''_t) = \mathrm{span}(A'_t)$. Thus, letting $A''_s = A'_s$ for all $s \in \bar{B} \smallsetminus \{q, t\}$, we have $\bigoplus_{s \leq r} \mathrm{span}(A''_s) \subsetneq \bigoplus_{s \leq r} \mathrm{span}(A'_s) = \mathrm{span}(\bar{A})$, which implies that $\pi\bar{B}$ does not index a basis of $\bar{A}$. This concludes the proof. □

Theorem A.3 (Thm. 3.4). *Any solution B of (9) maximizing V^B also maximizes $V^{B,p}$ for all $p \in Q$.*

Proof. Let M be the matrix of the companion system (4) and B' be a solution of (9) for which, for a given $p \in Q$, we have $V^{B',p} > V^{B,p}$. We remark that B' can be obtained from B by applying a sequence of t pivots. These correspond to a column index permutation π that can be expressed as the product of transpositions $\pi = \prod_{k \leq t}(I^k, J^k)$, where I^k indexes an exiting column and J^k an entering column of M for all $k \leq t$: thus, $B' = B \smallsetminus \{I^k \mid k \leq t\} \cup \{J^k \mid k \leq t\}$. Since M is block diagonal, then as in the proof of Lemma 3.2 we have that $M_B, M_{B'}$ are block diagonal. Furthermore, by Lemma 3.2, for all $k \leq t$ we have that the

column indices I^k, J^k are in the same block of M. Furthermore, each block corresponds to the subsystem of (4) given by $\forall J' \in \bar{\mathcal{N}}^{p-1}$ $A\ \mathbf{z}_{J'} = 0$ for a fixed $p > 1$: in other words, each block corresponds to a given $p > 1$. Thus, for a given $k \leq t$, in order for $\tilde{B} = (I^k, J^k)B$ to correspond to a basis of M, then I^k, J^k must index columns of M indexed by monomials with the same degree p: this implies that $V^{\tilde{B},p'} = V^{B,p'}$ for all $p' \neq p$. Hence, if $V^{\tilde{B},p} > V^{B,p}$ then $V^{\tilde{B}} > V^B$, contradicting the maximality of B. Induction on t concludes the proof. □

Lemma A.4 (Lemma 4.1). *For a dilation p_u from u in G w.r.t. a maximum matching M, we have $|\mathsf{U}(p_u)| = |\mathsf{V}(p_u)| + 1$.*

Proof. This follows because a dilation p_u is alternating w.r.t. M and starts and ends in U.

Lemma A.5 (Lemma 4.2). *If p, p' are different dilations in E, then $|\mathsf{U}(p \cup p')| > |\mathsf{V}(p \cup p')|$.*

Proof. Since all dilations are alternating w.r.t. M, $p \cup p'$ is incident to as at least many vertices of U as of $\mathcal{M} \smallsetminus \beta$; but because p, p' are also incident to their exposed root vertices, which are not themselves incident to M, the result follows by Lemma 4.1.

Proposition A.6 (Prop. 4.3). *In polynomial time, Alg. 1 finds a subgraph $G' = (U', V', E')$ of G such that $|U'|$ is maximum, $|U'| > |V'|$ and $V' = N_E(U')$, or determines that no such subgraph exists.*

Proof. Assume $G' = \varnothing$. This means that $\mathsf{U}(M) = U$ at the first iteration, which implies that $N_E(U) = \mathcal{M} \smallsetminus \beta$, therefore making G an infeasible instance. Now assume $G' \neq \varnothing$; then it is the union of different dilation sets found in different iterations of the **while** loop, and by Lemma 4.2 is such that $|U'| > |V'|$. We have to show that $V' = N_E(U')$, so let $\{u, v\} \in E$ with $u \in U'$. Suppose $v \in (\mathcal{M} \smallsetminus \beta) \smallsetminus V'$; then $\{u, v\}$ was never part of a dilation set at any iteration, and the only way $v \notin V'$ might occur is because u was removed from U at a certain iteration i whilst v was never removed. Consider the iteration (call it i) where u is removed: v is not removed, so v is not incident to any edge in the dilation set P found at itn. i. Let u' be the root node of P; since u is incident to P, there is a dilation $p \subseteq P$ from u' to which u is incident (i.e. there is an alternating path p' from u' to u), which implies that u is incident to M. Since $p \subseteq P$, v is not in p, which means that $\{v, u\} \notin M$. Now, either v is incident to M or not. If it is, then there is an edge $\{v, u''\} \in M$, which means that $p' \cup \{u'', v\}$ is an alternating path longer than p', which can be extended to a dilation using bfs, against the assumption that v is not adjacent to edges in P. If it is not, then v is an exposed vertex; but then p' can be extended to an augmenting alternating path from u' to v by simply adding $\{u, v\}$ to p' as the last edge, contradicting the maximality of M. Maximality of $|U'|$ follows because all dilations (including isolated vertices) are considered. The worst-case running time is $O(|U| \min(|U|, |\mathcal{M} \smallsetminus \beta|)|E|)$: a factor $O(|U|)$ for the **while** loop, a factor

$O(\min(|U|, |\mathcal{M} \setminus \beta|))$ for updating the data structures (the cardinality of M is bounded above by $\min(|U|, |\mathcal{M} \setminus \beta|)$), and a factor $O(|E|)$ in order to find maximum matchings and dilation sets (which are in the data structures used to find the matching in Step 3).

We remark that a more careful running time analysis of Alg. 1 would consider that $|U|$, $|\mathcal{M} \setminus \beta|$ and $|E|$ decrease at each iteration, and would therefore yield considerably lower worst-case complexity order.