ISSN 1342-2804

Research Reports on Mathematical and Computing Sciences

SDP Relaxations for Quadratic Optimization Problems Derived from Polynomial Optimization Problems

Martin Mevissen and Masakazu Kojima

April 2009, B–455

Department of
Mathematical and
Computing Sciences
Tokyo Institute of Technology

SERIES B: Operations Research

B-455 **SDP Relaxations for Quadratic Optimization Problems Derived from Polynomial Optimization Problems**

Martin Mevissen[⋆] and Masakazu Kojima[†]

April 2009

Abstract.

Based on the convergent sequence of SDP relaxations for a multivariate polynomial optimization problem (POP) by Lasserre, Waki et al. constructed a sequence of sparse SDP relaxations to solve sparse POPs efficiently. Nevertheless, the size of the sparse SDP relaxation is the major obstacle in order to solve POPs of higher degree. This paper proposes an approach to transform general POPs to quadratic optimization problems (QOPs), which allows to reduce the size of the SDP relaxation substantially. We introduce different heuristics resulting in equivalent QOPs and show how sparsity of a POP is maintained under the transformation procedure. As the most important issue, we discuss how to increase the quality of the SDP relaxation for a QOP. Moreover, we increase the accuracy of the solution of the SDP relaxation by applying additional local optimization techniques. Finally, we demonstrate the high potential of this approach through numerical results for large scale POPs of higher degree.

Key words.

Polynomial optimization, Quadratic optimization, Semidefinite programming relaxation, Sparsity

⋆ Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152-8552 Japan. *martime6@is.titech.ac.jp*. Research supported by the Doctoral Scholarship of the German Academic Exchange Service (DAAD).

† Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152-8552 Japan. *kojima@is.titech.ac.jp*. Research supported by Grant-in-Aid for Scientific Research (B) 19310096.

1 Introduction - SDP relaxations for POP

The global minimization of a multivariate polynomial over a semialgebraic set is a severely nonconvex, difficult optimization problem in general. In recent years semidefinite programming (SDP) relaxations for polynomial optimization problems (POP) have been investigated by a growing number of researchers. In [6] a hierarchy of SDP relaxations has been proposed whose optima have been proven to converge to the optimum of a POP for increasing order of the relaxation. The practical use of this powerful theoretical result has been limited by the capacity of current SDP solvers, as the size of the SDP relaxations grows rapidly with increasing order. An approach to attempt this problem has been the concept to exploit structured sparsity in a POP [5]. Whenever a POP satisfies a certain sparsity pattern, a convergent sequence of sparse SDP relaxations of substantially smaller size can be constructed [12, 7]. Compared to the original SDP relaxation in [6], the sparse SDP relaxation [12] provides stronger numerical results.

Still, the size of the sparse SDP relaxation remains the major obstacle in order to solve large scale POPs, which contain polynomials of higher degree. We propose a substitution procedure to transform an arbitrary POP into an equivalent quadratic optimization problem (QOP). It is based on replacing quadratic terms in higher degree monomials by new variables successively, and adding the substitution relations as constraints to the optimization problem. As the substitution procedure is not unique, we introduce different heuristics which aim at deriving a QOP with as few additional variables as possible. Moreover, we show that sparsity of a POP is maintained under the substitution procedure. The main advantage of deriving an equivalent QOP for a POP is that the sparse SDP relaxation of first order can be applied to solve it approximately.

The substitution procedure and the considerations to minimize the number of additional variables while maintaining the sparsity are presented in Section 2. While a POP and the QOP derived from it are equivalent, we face the problem that the quality of the SDP relaxation for a QOP deteriorates in many cases. We discuss in Section 3 how to tighten the SDP relaxation for a QOP in order to achieve good approximations to the global minimum even for SDP relaxation of first or second order. For that purpose methods as choosing appropriate lower and upper bounds for the multivariate variables, Branch-and-Cut bounds to shrink the feasible region of the SDP relaxation and locally convergent optimization methods are proposed. Finally, the power of this technique is demonstrated in Section 4, where it is applied to solve various large scale POP of higher degree.

2 Transforming a polynomial into a quadratic program

2.1 Sparse SDP relaxations for polynomial programs

The aim of this paper is to propose a technique to reduce the size of sparse semidefinite program (SDP) relaxations for general polynomial optimization problems (POP), which enables us to attempt large scale polynomial optimization efficiently. As the framework for our approach, we will briefly introduce the basic notations and the sparse SDP relaxations from [12].

Let $f_k \in \mathbb{R}[x]$ $(k = 0, 1, \ldots, m)$, where $\mathbb{R}[x]$ denotes the set of real-valued multivariate polynomials in $x \in \mathbb{R}^n$. Consider the following inequality constrained POP:

$$\begin{array}{lll} \text{minimize} & f_0(x) & \\ \text{subject to} & f_k(x) & \geq 0 \quad \forall\, k \in \{1, \ldots, m\}. \end{array} \tag{2.1}$$

Given a polynomial $f \in \mathbb{R}[x]$, $f(x) = \sum_{\alpha \in \mathbb{N}^n} c_\alpha(f) x^\alpha$ $(c_\alpha(f) \in \mathbb{R})$, we define its **support** by $\mathrm{supp}(f) = \{\alpha \in \mathbb{N}^n \mid c_\alpha(f) \neq 0\}$. Then, let $\xi^\star = \inf\{f_0(x) :\ f_k(x) \geq 0\ (k = 1, \ldots, m)\}$ and

$$F_k = \{i :\ \alpha_i \geq 1 \text{ for some } \alpha \in \mathrm{supp}(f_k) \subset \mathbb{N}^n\},$$

the index set of variables x_i involved in the polynomial f_k. To construct a sequence of SDP relaxations, a nonnegative integer $\omega \geq \omega_{\max}$ is chosen for the **relaxation order**, where $\omega_{\max} = \max\{\omega_k :\ k = 0, 1, \ldots, m\}$ and $\omega_k = \lceil \frac{1}{2}\deg(f_k) \rceil$ $(k = 0, \ldots, m)$. The dense SDP relaxation for (2.1) of order ω due to [6] is given by

$$\begin{array}{llll} \mathrm{dSDP}_\omega & \min & \sum_{\alpha \in \mathrm{supp}(f_0)} c_\alpha(f_0) y_\alpha & \\ & \text{s.t.} & M_{\omega - \omega_k}(f_k\, y) \succcurlyeq 0 & \forall\, k \in \{1, \ldots, m\}, \\ & & M_\omega(y) \succcurlyeq 0, & \end{array} \tag{2.2}$$

where $M_\omega(y)$ and $M_\omega(p\,y)$ denote the moment matrix and the localizing matrix for $p \in \mathbb{R}^n$ of order ω [6], respectively. The approach by Waki et al. [12] takes structured sparsity of a POP into account in order to reduce the size of the linear matrix inequalities in (2.2). We define the $n \times n$ **correlative sparsity pattern matrix (csp matrix)** R of the POP (2.1) such that

$$R_{i,j} = \begin{cases} \star & \text{if} \quad i = j, \\ \star & \text{if} \quad \alpha_i \geq 1 \text{ and } \alpha_j \geq 1 \text{ for some } \alpha \in \operatorname{supp}(f_0), \\ \star & \text{if} \quad i \in F_k \text{ and } j \in F_k \text{ for some } k \in \{1, 2, \ldots, m\}, \\ 0 & \text{otherwise.} \end{cases}$$

If R is sparse, the POP (2.1) is called **correlatively sparse**. The csp matrix R induces the **correlative sparsity pattern graph (csp graph)** $G(N, E)$. Its node set N and edge set E are defined as

$$N = \{1, 2, \ldots, n\} \quad \text{and } E = \{\{i, j\} : R_{i,j} = \star,\ i < j\},$$

respectively. Let $C_1, \ldots, C_p \subseteq N$ be the maximal cliques of $G(N, E')$, the chordal extension of $G(N, E)$. The number p of maximal cliques is upper bounded by n. Since each $F_k \subseteq N$ forms a clique, it must be contained in some maximal C_q. Let $\hat{F}_k$ be such a maximal clique C_q. The sparse SDP relaxation of order $\omega \geq \omega_{\max}$ [12] in dual form of (2.1) is then given by the following optimization problem:

$$\begin{array}{lll} \text{sSDP}_\omega & \min & \sum_{\alpha \in \operatorname{supp}(f_0)} c_\alpha(f_0) y_\alpha \\ & \text{s.t.} & M_{\omega - \omega_k}(f_k\, y,\ \hat{F}_k) \succcurlyeq 0 \quad \forall\, k \in \{1, \ldots, m\}, \\ & & M_\omega(y,\ C_l) \succcurlyeq 0 \qquad\quad\ \forall\, l \in \{1, \ldots, p\}, \end{array} \tag{2.3}$$

where $M_\omega(y,\ C_l)$ and $M_\omega(p\,y,\ C_l)$ denote the partial moment matrix and the partial localizing matrix of order ω for the set C_l, respectively. For instance in the case of $n = 3$ and $g(x) = x_1^2 + 2x_2 + 3$, they have the form

$$\begin{aligned} M_2(y, \{1, 3\}) &= \begin{pmatrix} y_{000} & y_{100} & y_{001} & y_{200} & y_{101} & y_{002} \\ y_{100} & y_{200} & y_{101} & y_{300} & y_{201} & y_{102} \\ y_{001} & y_{101} & y_{002} & y_{201} & y_{102} & y_{003} \\ y_{200} & y_{300} & y_{201} & y_{400} & y_{301} & y_{202} \\ y_{101} & y_{201} & y_{102} & y_{301} & y_{202} & y_{103} \\ y_{002} & y_{102} & y_{003} & y_{202} & y_{103} & y_{004} \end{pmatrix}, \\ M_1(g\, y, \{1, 2\}) &= \begin{pmatrix} y_{200} + 2y_{010} + 3y_{000} & y_{300} + 2y_{110} + 3y_{100} & y_{210} + 2y_{020} + 3y_{010} \\ y_{300} + 2y_{110} + 3y_{100} & y_{400} + 2y_{210} + 3y_{200} & y_{310} + 2y_{120} + 3y_{110} \\ y_{210} + 2y_{020} + 3y_{010} & y_{310} + 2y_{120} + 3y_{110} & y_{220} + 2y_{030} + 3y_{020}. \end{pmatrix} \end{aligned}$$

Note, $M_\omega(y, C_l)$ is a symmetric matrix of size $\begin{pmatrix} n_l + \omega \\ \omega \end{pmatrix}$, where $n_l := \mid C_l \mid$. The length of a partial moment vector $y(C_l)$ is given by $\begin{pmatrix} n_l + 2\omega \\ 2\omega \end{pmatrix}$. The number of variables in (2.3) is bounded by $p \begin{pmatrix} n_{\max} + 2\omega \\ 2\omega \end{pmatrix}$, where $n_{\max} = \max_{1 \leq i \leq p} n_i$. This is to be compared with $\begin{pmatrix} n + 2\omega \\ 2\omega \end{pmatrix}$, the upper bound for the number of variables in the dense SDP relaxation dSDP_ω. Thus, in the case $n_{\max} \ll n$ the size of the sparse SDP relaxation is substantially smaller than the size of the dense one.

2.2 Sparse SDP relaxations for quadratic programs

The technique to transform a POP into an equivalent quadratic optimization problem (QOP), which we introduce in this section, aims at reducing the size of the SDP relaxation by decreasing the minimum relaxation order $\omega_{\max}$, whereas the technique in [12] aims at reducing the SDP relaxation by decreasing n. A general quadratic optimization problem (QOP) is a special case of the POP (2.1), where the polynomials f_i $(i = 0, \ldots, m)$ are at most of degree 2. With respect to the definition of ω_k, the minimal relaxation order $\omega_{\max}$ of the sparse SDP relaxation (2.3) equals one. As pointed out in [12], the sparse SDP relaxation sSDP_1 and the dense SDP relaxation dSDP_1 of order one are equivalent for any QOP. The equivalence of a QOP and its SDP relaxation has been shown for a few restrictive classes of QOPs. For instance, if in a QOP f_0 and $-f_i$ $(i = 1, \ldots, m)$ are convex quadratic polynomials, the QOP is equivalent to the corresponding SDP relaxation [8]. Also, equivalence of QOPs and their SDP relaxations was shown for the class of uniformly OD-nonpositive QOPs [4].

2.3 Transformation algorithm

To illustrate the idea of our transformation technique, consider the following example of a simple unconstrained POP, whose optimal value is $-\infty$:

$$\min\ 10x_1^3 - 10^2x_1^3x_2 + 10^3x_1^2x_2^2 - 10^4x_1x_2^3 + 10^5x_2^4 \tag{2.4}$$

It is straight forward that POP (2.4) is equivalent to

$$\begin{array}{ll} \min & 10x_1x_3 - 10^2x_3x_4 + 10^3x_4^2 - 10^4x_4x_5 + 10^5x_5^2 \\ \text{s.t.} & x_3 = x_1^2, \\ & x_4 = x_1x_2, \\ & x_5 = x_2^2, \end{array} \tag{2.5}$$

where we introduced three additional variables x_3, x_4 and x_5. Obviously QOP (2.5) is not the only QOP equivalent to POP (2.4): The QOP

$$\begin{array}{ll} \min & 10x_3 - 10^2x_2x_3 + 10^3x_5x_6 - 10^4x_1x_4 + 10^5x_2x_4 \\ \text{s.t.} & x_3 = x_1x_5, \\ & x_4 = x_2x_6, \\ & x_5 = x_1^2, \\ & x_6 = x_2^2, \end{array} \tag{2.6}$$

is equivalent to (2.4) as well. We notice the number of additional variables in QOP (2.5) equals three, whereas it equals four in QOP (2.6). Thus, there are numerous ways to transform a higher degree POP into a QOP in general. For the transformation procedures we are proposing, we consider 1) the number of additional variables should be as small as possible, in order to obtain a SDP relaxation of smaller size, 2) sparsity of a POP should be maintained under the transformation and 3) the quality of the SDP relaxation for the derived QOP should be as good as possible. How to deal with 3) is discussed in Section 3, 1) and 2) are discussed in the following.

2.4 Maintaining sparsity

The transformation algorithm proposed in the previous subsection raises the question, whether the correlative sparsity of a POP is preserved under the transformation, i.e., whether the resulting QOP is correlative sparse as well.

Let POP* be a correlative sparse POP of dimension n, $G(N, E')$ the chordal extension of its csp graph, $(C_1, \ldots, C_p)$ the maximal cliques of $G(N, E')$ and $n_{\max} = \max_{i=1,\ldots,p} \mid C_i \mid$. Let $x_{n+1} = x_ix_j$ be the substitution variable for some $i, j \in \{1, \ldots, n\}$ determined by either criterion A or B, which are explained in Section 2.5. Let P$\tilde{\text{O}}$P denote the POP derived after substituting $x_{n+1} = x_ix_j$ in POP*. Given the chordal extension $G(N, E')$ of the csp graph of POP*, a chordal extension of the csp graph of P$\tilde{\text{O}}$P over the vertex set $\tilde{N} = N \cup \{n+1\}$ can be obtained by the extension: For a clique C_l with $\{i, j\} \subset C_l$ add the edges $\{v, n+1\}$ for all $v \in C_l$ and obtain the clique $\tilde{C}_l$. For each clique C_k not containing $\{i, j\}$, set $\tilde{C}_k = C_k$. In the end we obtain the graph $G(\tilde{N}, \tilde{E}')$ which is a chordal extension of the csp graph $G(\tilde{N}, \tilde{E})$ of P$\tilde{\text{O}}$P. Note, $(\tilde{C}_1, \ldots, \tilde{C}_p)$ are maximal cliques for $G(\tilde{N}, \tilde{E}')$ and for all $\tilde{C}_l$ holds $\mid \tilde{C}_l \mid \leq \mid C_l \mid +1$, i.e. $\tilde{n}_{\max} \leq n_{\max} + 1$. Moreover, the number of maximal cliques p remains unchanged under the transformation. As pointed out, $G(\tilde{N}, \tilde{E}')$ is one possible chordal extension of $G(\tilde{N}, \tilde{E})$. It seems reasonable to expect that the heuristics we are using for the chordal extension, such as the reverse Cuthill-McKee and the symmetric minimum degree ordering, add less edges to $G(\tilde{N}, \tilde{E})$ than we did in constructing $G(\tilde{N}, \tilde{E}')$. Thus, we are able to apply the sparse SDP relaxation efficiently to the POPs derived after each iteration of the transformation algorithm. For illustration we consider Figure 1 and Figure 2, where the csp matrices of two POPs and their QOPs are pictured.

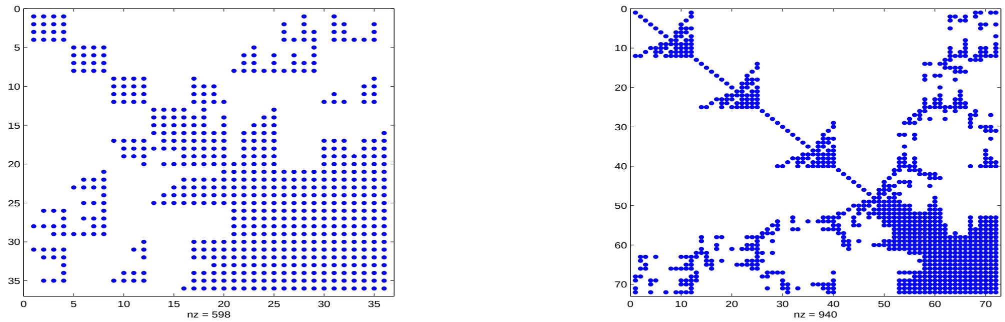

Figure 1: CSP matrix of the chordal extension of POP *pdeBifurcation(7)* (left) and its QOP (right) derived under strategy BI (c.f. 2.5).

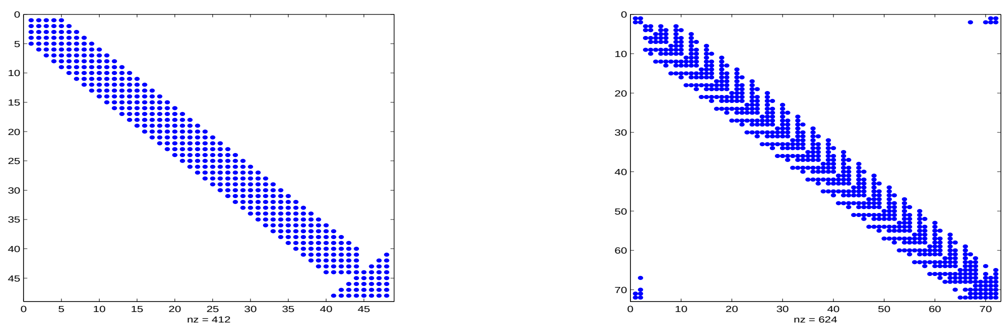

Figure 2: CSP matrix of the chordal extension of POP *Mimura(25)* (left) and its QOP (right) derived under strategy BI (c.f. 2.5).

We observe that the sparsity pattern of the chordal extension of the csp graph is maintained under the substitution procedure. Nevertheless, if the number of substitutions, which is required to transform a higher degree POP into a QOP, is far greater than the number of variables of the original POP, it may occur that we obtain a dense QOP under the transformation procedure. To illustrate this effect, consider the chordal extension of csp matrix of the QOP derived for the POP *randomEQ(7,3,5,8,0)* which is pictured in Figure 3. In that example, the number n of variables of the original POP equals seven, the number of additional variables equals 108.

2.5 Minimizing the number of additional variables

Let n denote the number of variables involved in a POP and $\tilde{n}$ the number of variables in the corresponding QOP. The first question we are facing is, how to transform a POP into a QOP such that the number $k_0 := \tilde{n} - n$ of additional variables is as small as possible. Each additional variable x_{n+k} corresponds to the substitution of a certain quadratic monomial $x_i x_j$ by x_{n+k}. Given an arbitrary POP, the question to find a substitution procedure minimizing $\tilde{n}$ is a difficult problem. We propose four different heuristics for transforming a POP into a QOP, which aim at reducing the number k_0 of additional variables. At the end of this section we give some motivation, why it is more important to find a strategy optimizing the quality of the SDP relaxation than one that minimizes the number k_0 of additional variables.

Our transformation algorithm iterates substitutions of pairs of quadratic monomials $x_i x_j$ in the higher degree monomials in objective function and constraints by a new variable x_{n+k}, and adding the substitution

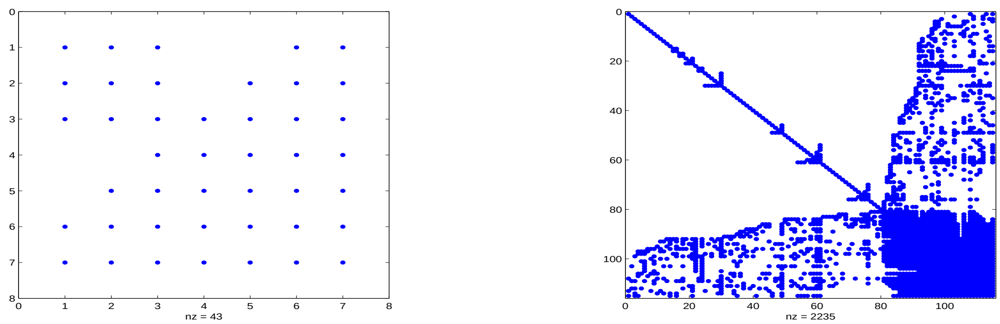

Figure 3: CSP matrix of the chordal extension of POP randomWithEQ(7,3,5,8,0) (left) and its QOP (right).

relation $x_{n+k} = x_i x_j$ as constraints to the POP. Let POP^0 denote the original POP, and POP^k the POP obtained after the k-th iteration, i.e. after substituting $x_{n+k} = x_i x_j$ and adding it as constraint to POP^{k-1}. The algorithm terminates as soon as POP^{k_0} is a QOP for some $k_0 \in \mathbb{N}$. In each iteration of the transformation algorithm we distinguish two steps. The first one is to choose which pair of variables (x_i, x_j) $(1 \leq i, j \leq n+k)$ is substituted by the additional variable x_{n+k+1}. The second one is to choose to which extent $x_i x_j$ is substituted by x_{n+k+1} in each higher degree monomial.

Step 1: Choosing the substitution variables

Definition 1 *Let POP^k be a POP of dimension $\tilde{n}$ with $\tilde{m}$ constraints. The* **higher monomial set** $\mathcal{M}_S^k$ *of POP^k is given by*

$$\mathcal{M}_S^k = \left\{ \alpha \in \mathbb{N}^{\tilde{n}} \mid \exists i \in \{0, \ldots, \tilde{m}\} \text{ s.t. } \alpha \in supp(f_i) \text{ and } \mid \alpha \mid \geq 3 \right\},$$

and the **higher monomial list** $\mathcal{M}^k$ *of POP^k by*

$$\mathcal{M}^k = \left\{ (\alpha, w_\alpha) \mid \alpha \in \mathcal{M}_S^k \text{ and } w_\alpha := \# \left\{ i \mid \alpha \in supp(f_i) \right\} \right\}.$$

By Definition 1, the higher monomial list of a QOP is empty.

Definition 2 *Given $\alpha \in \mathbb{N}^n$ and a pair (i, j) where $1 \leq i, j \leq n$, we define the* **dividing coefficient** $k_{i,j}^\alpha \in \mathbb{N}_0$ *as the integer that satisfies* $\frac{x^\alpha}{(x_i x_j)^{k_{i,j}^\alpha}} \in \mathbb{R}[x]$ *and* $\frac{x^\alpha}{(x_i x_j)^{k_{i,j}^\alpha + 1}} \notin \mathbb{R}[x]$.

Given POP^k the k-th iterate of POP^0 and its higher monomial list $\mathcal{M}^k$, determine the symmetric matrix $C(\text{POP}^k) \in \mathbb{R}^{(n+k)\times(n+k)}$ given by

$$C(\text{POP}^k)_{i,j} = C(\text{POP}^k)_{j,i} = \sum_{(\alpha, w_\alpha) \in \mathcal{M}^k} k_{i,j}^\alpha \, w_\alpha.$$

We consider two alternatives to choose a pair $(x_i, x_j)(1 \leq i, j \leq n+k)$ to be substituted by x_{n+k+1}:

A. **Naive criterion:** Choose a pair (x_i, x_j) such that there exists a $\alpha \in \mathcal{M}_S(\text{POP}^k)$ which satisfies $\frac{x^\alpha}{x_i x_j} \in \mathbb{R}[x]$.

B. **Maximum criterion** Choose a pair (x_i, x_j) such that $C(\text{POP}^k)_{i,j} \geq C(\text{POP}^k)_{u,v}$ $\forall 1 \leq u, v \leq n+k$.

Step 2: Choose the substitution strategy Next we have to decide to what extent we substitute $x_{n+k+1} = x_i x_j$ in each monomial of $\mathcal{M}_S(\text{POP}^k)$. We will distinguish **full** and **partial substitution**. Let us demonstrate the importance of considering that question on the following two examples.

Example 1 *Consider two different substitution strategies for transforming the problem to minimize x_1^4 into a QOP.*

$$\begin{array}{lllcll} & & & \min x_1^4 & & \\ (1) & \swarrow & & & \searrow & (2) \\ \min & x_2^2 & & & \min & x_1^2 x_2 \\ s.t. & x_2 = x_1^2 & & & s.t. & x_2 = x_1^2 \\ & & & & \downarrow & \\ & & & & \min & x_1 x_3 \\ & & & & s.t. & x_2 = x_1^2 \\ & & & & & x_3 = x_1 x_2 \end{array} \tag{2.7}$$

In both substitution strategies, we choose x_1^2 for substitution in the first step. In (1) we fully substituted x_1^2 by x_2, whereas in (2) we substituted x_1^2 partially. By choosing full substitution in the first iteration in (1), we need one additional variable to obtain a QOP, partial substitution requires two additional variables to yield a QOP.

Example 2

$$\begin{array}{lllcll} & & & \min x_1^6 & & \\ & & & s.t.\ \ x_1^3 x_2 \geq 0 & & \\ (1) & \swarrow & & & \searrow & (2) \\ \min & x_3^3 & & & \min & x_1^2 x_3^2 \\ s.t. & x_1 x_2 x_3 \geq 0 & & & s.t. & x_1 x_2 x_3 \geq 0 \\ & x_3 = x_1^2 & & & & x_3 = x_1^2 \\ \downarrow & & & & \downarrow & \\ \min & x_3 x_4 & & & \min & x_4^2 \\ s.t. & x_1 x_2 x_3 \geq 0 & & & s.t. & x_2 x_4 \geq 0 \\ & x_3 = x_1^2 & & & & x_3 = x_1^2 \\ & x_4 = x_3^2 & & & & x_4 = x_1 x_2 \\ \downarrow & & & & & \\ \min & x_3 x_4 & & & & \\ s.t. & x_2 x_5 \geq 0 & & & & \\ & x_3 = x_1^2 & & & & \\ & x_4 = x_3^2 & & & & \\ & x_5 = x_1 x_3 & & & & \end{array} \tag{2.8}$$

In this example full substitution (1) of x_1^2 requires three, and partial substitution (2) only two additional variables to yield a QOP.

The examples illustrate it depends on the structure of the monomial set, whether partial or full substitution require less additional variables and result in a smaller size of the SDP relaxation. In general partial and full substitution are given as follows.

I. **Full substitution:** Let $t_{f_{i,j}^r} : \mathbb{R}[x] \to \mathbb{R}[z]$, where $x \in \mathbb{R}^r$ and $z \in \mathbb{R}^{r+1}$ for a $r \in \mathbb{N}$ and $i, j \in \{1, \dots, r\}$, be a linear operator defined by its mappings for each monomial x^α,

$$t_{f_{i,j}^r}(x^\alpha) = \begin{cases} z_1^{\alpha_1} \dots z_{i-1}^{\alpha_{i-1}} z_i^{\alpha_i - \min(\alpha_i,\alpha_j)} z_{i+1}^{\alpha_{i+1}} \dots z_{j-1}^{\alpha_{j-1}} z_j^{\alpha_j - \min(\alpha_i,\alpha_j)} z_{j+1}^{\alpha_{j+1}} \dots z_r^{\alpha_r} z_{r+1}^{\min(\alpha_i,\alpha_j)}, & \text{if } i \neq j, \\ z_1^{\alpha_1} \dots z_{i-1}^{\alpha_{i-1}} z_i^{\mathrm{mod}(\alpha_i,2)} z_{i+1}^{\alpha_{i+1}} \dots z_r^{\alpha_r} z_{r+1}^{\lfloor \frac{\alpha_i}{2} \rfloor}, & \text{if } i = j. \end{cases}$$

Thus, $t_{f_{i,j}^r}(g(x)) = \sum_{\alpha \in \mathrm{supp}(g)} c_\alpha(g) t_{f_{i,j}^r}(x^\alpha)$ for any $g \in \mathbb{R}[x]$. The operator $t_{f_{i,j}^{n+k}}$ substitutes $x_i x_j$ by x_{n+k+1} in each monomial to the maximal possible extent.

II. **Partial substitution:** Let $t_{p_{i,j}^r} : \mathbb{R}[x] \to \mathbb{R}[z]$, where $x \in \mathbb{R}^r$ and $z \in \mathbb{R}^{r+1}$ for a $r \in \mathbb{N}$ and $i, j \in \{1, \dots, r\}$, be a linear operator defined by its mappings for each monomial x^α,

$$t_{p_{i,j}^r}(x^\alpha) = \begin{cases} t_{f_{i,j}^r}(x^\alpha), & \text{if } i \neq j, \\ t_{f_{i,j}^r}(x^\alpha), & \text{if } i = j \text{ and } \alpha_i \text{ odd}, \\ t_{f_{i,j}^r}(x^\alpha), & \text{if } i = j \text{ and } \log_2(\alpha_i) \in \mathbb{N}_0, \\ z_1^{\alpha_1} \dots z_{i-1}^{\alpha_{i-1}} z_i^{g_i} z_{i+1}^{\alpha_{i+1}} \dots z_r^{\alpha_r} z_{r+1}^{\frac{1}{2}(\alpha_i - g_i)}, & \text{else}, \end{cases}$$

where $g_i := \gcd(2^{\lfloor \log_2(\alpha_i) \rfloor}, \alpha_i)$. Thus, $t_{p_{i,j}}^{\,r}(g(x)) = \sum_{\alpha \in \mathrm{supp}(g)} c_\alpha(g) t_{p_{i,j}}^{\,r}(x^\alpha)$ for any $g \in \mathbb{R}[x]$.

We notice that full and partial substitution only differ in the case $i = j$, α_i even and $\log_2(\alpha_i) \notin \mathbb{N}_0$ holds. By pairwise combining the choice of A or B in Step 1 and the choice of I or II in Step 2, we obtain four different procedures to transform POP^{k-1} into POP^k that we denote as AI, AII, BI and BII. We do not expect AI or AII to result in a POP with a small number of substitutions, as A does not take into account the structure of the higher degree monomial list $\mathcal{M}_S^{k-1}$, but we use AI and AII to evaluate the potential of BI and BII. The numerical performance of these four procedures is demonstrated on some example POPs in Table 2, where n denotes the number of variables in the original POP, *deg* the degree of the highest order polynomial in the POP, and k_0 the number of additional variables required to transform the POP into a QOP under the respective substitution strategy. The POPs *pdeBifurcation(n)* are derived from discretizing differential equations [9], the other POPs are test problems from [12]. As expected, strategy B is superior to A for all but one example class of POP, when reducing the number of variables is concerned.

The entire algorithm to transform a POP into a QOP can be summarized by the scheme in Table 1. As mentioned before the QOP of dimension $n + k$ derived by AI, AII, BI or BII is equivalent to the original POP of dimension n. In fact it is easy to see, if $\tilde{x} \in \mathbb{R}^{n+k}$ an optimal solution of the QOP, the vector $(\tilde{x}_1, \ldots, \tilde{x}_n)$ of the first n components of $\tilde{x}$ is an optimizer of the original POP.

INPUT		POP^0 with $\mathcal{M}_S^0$
WHILE		$\mathcal{M}_S^k \neq \emptyset$
	1.	Determine the pair (x_i, x_j) for substitution by A or B.
	2.	Apply $t_{f_{i,j}}^{\,k}$ or $t_{p_{i,j}}^{\,k}$ to each polynomial in POP^k and derive POP^{k+1}.
	3.	Update $k \to k+1$, $\mathrm{POP}^k \to \mathrm{POP}^{k+1}$, $\mathcal{M}_S^k \to \mathcal{M}_S^{k+1}$.
OUTPUT		$\mathrm{QOP} = \mathrm{POP}^{k_0}$

Table 1: Scheme for transforming a POP into a QOP

POP	n	deg	k_0(AI)	k_0(AII)	k_0(BI)	k_0(BII)
BroydenBand(20)	20	6	229	211	60	40
BroydenBand(60)	60	6	749	691	180	120
nondquar(32)	32	4	93	93	94	94
nondquar(8)	8	4	21	21	22	22
optControl(10)	60	4	60	60	60	60
randINEQ(8,4,6,8,0)	8	8	253	307	248	238
randEQ(7,3,5,8,0)	7	8	135	146	116	115
pdeBifurcation(5)	25	3	25	25	25	25
pdeBifurcation(10)	100	3	100	100	100	100
randINEQ(3,1,3,16,0)	3	16	145	192	105	117
randUnconst(3,2,3,14,0)	3	14	86	107	63	69

Table 2: Number of required variables for strategies AI, AII, BI and BII

2.6 Computational complexity

Finally, let us consider how the size of the sparse SDP relaxation of order $\omega = 1$ for a QOP depends on the number k_0 of additional variables. Let a sparse POP of dimension n be given by the polynomials $(f_0, f_1, \ldots, f_m)$ and the maximal cliques $(C_1, \ldots, C_p)$ of the chordal extension. With the construction in Section 2.4, the corresponding QOP of dimension $\tilde{n} = n + k_0$ has the maximal cliques $(\tilde{C}_1, \ldots, \tilde{C}_p)$ such that $C_i \subseteq \tilde{C}_i$ and $\tilde{n}_i \leq n_i + k_0$ for all $(i = 1, \ldots, p)$, where $n_i = \mid C_i \mid$ and $\tilde{n}_i = \mid \tilde{C}_i \mid$. All partial localizing matrices $M_0(f_k\, y, \hat{F}_k)$ are scalars in $\mathrm{sSDP}_1(QOP)$. The size of the partial moment matrices $M_1(y, \tilde{C}_i)$ is

$$d(1, \tilde{n}_i) = \tilde{n}_i + 1 \leq n_i + k_0 + 1 = O(k_0). \tag{2.9}$$

Thus, the size of the linear matrix inequality is bounded by

$$\sum_{j=1}^{m+k_0} 1 + \sum_{i=1}^{p} d(1, \tilde{n}_i) \leq m + k_0 + p\,(n_{\max} + k_0 + 1) \leq m + k_0 + n\,(n_{\max} + k_0 + 1). \tag{2.10}$$

The length of the vector variable y in $\mathrm{sSDP}_1(QOP)$ is bounded by

$$\begin{array}{ll} \mid y \mid & \leq \sum_{i=1}^{p} \mid y(\tilde{C}_p) \mid = \sum_{i=1}^{p} d(2, 2\tilde{n}_i) \leq \frac{1}{2} p\,(2n_{\max} + 2k_0 + 2)\,(2n_{\max} + 2k_0 + 1) \\ & \leq 2p\,(n_{\max} + k_0 + 1)^2 \leq 2n\,(n_{\max} + k_0 + 1) = O(k_0^2). \end{array} \tag{2.11}$$

Thus, the size of the linear matrix inequalities of the sparse SDP relaxation is linear and the length of the moment vector y quadratic in the number k_0 of additional variables. For this reason the computational cost does not grow too fast, even if k_0 is not minimal. Heuristics BI and BII are sufficient in order to derive QOP with a small number k_0 of additional variables.

Moreover, the bounds (2.10) and (2.11) for the size of the primal and dual variables of the SDP relaxation for the QOP are to be compared to the respective bounds for the SDP relaxation of the POP. If we assume $\omega_{\max} = \omega_i$ for all $i \in \{1, \ldots, m\}$, the size of the linear matrix inequality in the SDP relaxation of order $\omega_{\max}$ for the original POP can be bounded by

$$\sum_{j=1}^{m} d(n_j, \omega_{\max} - \omega_j) + \sum_{i=1}^{p} d(n_i, \omega_{\max}) \leq m + n \begin{pmatrix} n_{\max} + \omega_{\max} \\ \omega_{\max} \end{pmatrix}, \tag{2.12}$$

and the length of the moment vector by

$$\sum_{i=1}^{p} d(2n_i, 2\omega_{\max}) \leq n \begin{pmatrix} 2n_{\max} + 2\omega_{\max} \\ 2n_{\max} \end{pmatrix}. \tag{2.13}$$

Already for $\omega_{\max} = 2$ the bounds (2.12) and (2.13) are of second and fourth degree in $n_{\max}$, whereas (2.10) and (2.11) are linear and quadratic in $n_{\max} + k_0$, respectively. Therefore we can expect a substantial reduction of the SDP relaxation under the transformation procedure. Note, we did not exploit any sparsity in the SDP relaxation or any intersection of the maximal cliques $(C_1, \ldots, C_p)$ and $(\tilde{C}_1, \ldots, \tilde{C}_p)$ when deriving these bounds. Thus, the actual size of SDP relaxations in numerical experiments may be far smaller than the one suggested by these bounds.

3 Quality of SDP relaxations for QOP

A polynomial optimization problem (POP) and the quadratic optimization problem (QOP) derived from it under one of the transformation strategies AI, AII, BI or BII are equivalent. Nevertheless, the same statement does not hold for the SDP relaxations of both problems. In fact, the SDP relaxation for QOP is weaker than the SDP relaxation for the original POP as the following negative result states.

Proposition 1 *Let a POP of dimension n with $\omega_{\max} > 1$ of form (2.1) be given by the set of polynomials $(f_0, f_1, \ldots, f_m)$ and the corresponding QOP of dimension $n + k$ derived via AI, AII, BI or BII by $(\tilde{f}_0, \tilde{f}_1, \ldots, \tilde{f}_m)$. Then, for each feasible solution of $\mathrm{dSDP}_{\omega_{\max}}$(POP), there exists a feasible solution of dSDP_1(QOP) with the same objective value. Thus,* $\min$*(dSDP_1(QOP))*$\leq \min$*($\mathrm{dSDP}_{\omega_{\max}}$(POP)).*

Proof:

Let $y \in \mathbb{R}^{d(n, 2\omega_{\max})}$ be a feasible solution of $\mathrm{SDP}_{\omega_{\max}}(\mathrm{POP})$, where $d(n, \omega) = \begin{pmatrix} n + \omega \\ \omega \end{pmatrix}$. Each y_α corresponds to a monomial x^α for all α with $\mid \alpha \mid \leq 2\omega_{\max},\ x \in \mathbb{R}^n$. Moreover, with respect to the substitution relation for all monomials $\tilde{x}^\alpha \in \mathbb{R}^{n+k}$ with $\mid \alpha \mid \leq 2$ there exists a monomial $x^{\beta(\alpha)} \in \mathbb{R}^n$, such that

$$\tilde{x}^\alpha = x^{\beta(\alpha)}, \mid \beta(\alpha) \mid \leq 2\omega_{\max}. \tag{3.1}$$

As $\beta(\cdot)$ in (3.1) is constructed via the substitution relations,

$$\beta(\alpha_1) = \beta(\alpha_2) \tag{3.2}$$

holds for $\alpha_1, \alpha_2 \in \mathbb{N}^{n+k}$ with $\mid \alpha_1 \mid = \mid \alpha_2 \mid \leq 2$, whenever QOP has a substitution constraint $\tilde{x}^{\alpha_1} = \tilde{x}^{\alpha_2}$. Now, define $\tilde{y} \in \mathbb{R}^{d(n+k,2)}$ where $\tilde{y}_\alpha := y_{\beta(\alpha)}$ for all $\mid \alpha \mid \leq 2$. Then, $\tilde{y}$ is feasible for $\mathrm{dSDP}_1(\mathrm{QOP})$, as all equality constraints derived from substitutions are satisfied due to (3.2), and as the moment matrix $M_1(\tilde{y})$ and the localizing matrices $M_0(\tilde{y}\tilde{f}_k)$ $(k = 1, \ldots, m)$ are principal submatrices of $M_{\omega_{\max}}(y)$ and $M_{\omega_{\max}-\omega_k}(y\, f_k)$ (k=1,...,m), respectively. Finally, the objective values for y and $\tilde{y}$ coincide. $\square$

This result for the dense SDP relaxation can be extended to the sparse SDP relaxation of minimal relaxation order in an analog manner, if the maximal cliques $(\tilde{C}_1, \ldots, \tilde{C}_m)$ of the chordal extended csp graph of the QOP are chosen appropriately with respect to the maximal cliques of the chordal extended csp graph of the POP. Therefore it seems reasonable to expect that in general the sparse SDP relaxation of minimal order $\omega = 1$ for the QOP provides an approximation to the global minimum of the POP which is far weaker than the one provided by the sparse SDP relaxation with $\omega = \omega_{\max} > 1$ applied to the original POP. In the following we introduce different techniques which can be used to improve the quality of the sparse SDP relaxation for QOPs.

3.1 Local optimization methods

As pointed out before, the minimum of the sparse SDP relaxation converges to the minimum of the QOP for $\omega \to \infty$. Moreover, an accurate approximation can be obtained by the sparse SDP relaxation of order $\omega \in \{\omega_{\max}, \ldots, \omega_{\max} + 3\}$ for many POPs [12]. Nevertheless, the quality of the sparse SDP relaxation for the QOP is weaker than the one for the original POP. Therefore, the solution provided by the sparse SDP relaxation for the QOP can be understood as a first approximation to the global minimizer of the original POP, and it may serve as initial point for a locally convergent optimization technique applied to the original POP. For instance sequential quadratic programming (SQP) [2] can be applied to POP where the sparse SDP solution for the corresponding QOP is taken as starting point. In the case a POP has equality constraints only, the number of constraints coincides with the number of variables and the feasible set is finite, we may succeed in finding the global optimizer of the POP by applying Newton's method for nonlinear systems [10] to the polynomial system given by the feasible set of the POP, again starting from the solution provided by the sparse SDP relaxation for the QOP.

3.2 Higher accuracy via Branch-and-Cut bounds

The sparse SDP relaxations proposed in [12] incorporate lower and upper bounds for each component of the n-dimensional variable,

$$\mathrm{lbd}_i \leq x_i \leq \mathrm{ubd}_i, \qquad \forall i \in \{1, \ldots, n\}, \tag{3.3}$$

in order to establish the compactness of the feasible set of a POP. Compactness is a necessary condition to guarantee the convergence of the sequence of sparse SDP relaxations towards the global optimum of the POP. Moreover, the numerical performance for solving the sparse SDP relaxations depends heavily on the bounds (3.3). The tighter we choose these bounds, the better approximates the solution of the SDP the minimizer of the POP. Prior to solving the sparse SDP relaxations for the QOP derived from a POP, we fix the bounds (3.3) for the components of the POP and determine lower and upper bounds for the additional variables according to the substitution relation. For instance for $x_{n+1} = x_i^2$ the bounds are defined as

$$\begin{aligned} \mathrm{lbd}_{n+1} &= \begin{cases} 0, & \text{if } \mathrm{lbd}_i \leq 0 \leq \mathrm{ubd}_i \\ \min(\mathrm{lbd}_i^2, \mathrm{ubd}_i^2), & \text{else} \end{cases}, \\ \mathrm{ubd}_{n+1} &= \max(\mathrm{lbd}_i^2, \mathrm{ubd}_i^2). \end{aligned} \tag{3.4}$$

In Section 4 we will discuss the sensitivity of the choice of the lower and upper bounds on the accuracy of the SDP solution for some example POPs.

A more sophisticated technique to increase the quality of the SDP relaxation of the QOP is inspired by a Branch-and-Cut algorithm for bilinear matrix inequalities due to Fukuda and Kojima [3]. As nonconvex quadratic constraints can be reduced to bilinear ones, we are able to adapt this technique for a QOP derived from a higher degree POP. The technique is based on cutting the feasible region of the SDP, such that every feasible solution of the QOP remains feasible for the SDP. We distinguish two sets of constraints which resemble the convex relaxations (5) proposed in [3]. Let $(f_0, f_1, \ldots, f_m)$ be a QOP with lower and upper bounds lbd_i and ubd_i for all components x_i $(i = 1, \ldots, n)$. The first set of constraints we consider is the following. For each constraint f_i $(i = 1, \ldots, m)$ of form $x_k = x_i x_j$ with $i \neq j$ we add the following constraints to the QOP

$$\begin{aligned} x_k &\leq \mathrm{ubd}_j x_i + \mathrm{lbd}_i x_j - \mathrm{lbd}_i \mathrm{ubd}_j \\ x_k &\leq \mathrm{lbd}_j x_i + \mathrm{ubd}_i x_j - \mathrm{ubd}_i \mathrm{lbd}_j. \end{aligned} \tag{3.5}$$

For each constraint of the form $x_k = x_i^2$ we add the following constraint to the QOP

$$x_k \leq (\mathrm{ubd}_i + \mathrm{lbd}_i)\, x_i - \mathrm{lbd}_i \mathrm{ubd}_i. \tag{3.6}$$

The second set of constraints shrinks the feasible set of the SDP relaxation even further than the constraints (3.5) and (3.6). For each monomial $x_i x_j$ of degree 2 which occurs in the objective f_0 or one of the constraints f_i $(i = 1, \ldots, m)$ of the QOP, we add constraints as follows. If the QOP contains a constraint f_i $(i = 1, \ldots, m)$ of the form $x_k = x_i x_j$, we add the constraints (3.5) for $i \neq j$ and (3.6) for $i = j$. If the QOP does not contain a constraint $x_k = x_i x_j$, we add the quadratic constraints

$$\begin{aligned} x_i x_j &\leq \mathrm{ubd}_j x_i + \mathrm{lbd}_i x_j - \mathrm{lbd}_i \mathrm{ubd}_j \\ x_i x_j &\leq \mathrm{lbd}_j x_i + \mathrm{ubd}_i x_j - \mathrm{ubd}_i \mathrm{lbd}_j \end{aligned} \tag{3.7}$$

for $i \neq j$ and the constraint

$$x_i^2 \leq (\mathrm{ubd}_i + \mathrm{lbd}_i)\, x_i - \mathrm{lbd}_i \mathrm{ubd}_i \tag{3.8}$$

for $i = j$. When linearized, both, the linear constraints (3.5) and (3.6) and the quadratic constraints (3.7) and (3.8) result in a smaller feasible region of the SDP relaxation which still contains the feasible region of the QOP. The efficiency of these sets of additional constraints is demonstrated in Section 4 as well.

4 Numerical Examples

The substitution procedure and the sparse SDP relaxations are applied to a number of test problems. These test problems encompass medium and large scale POPs of higher degree. The numerical performance of the sparse SDP relaxations of these POPs under the transformation algorithm is evaluated. In the following the Branch-and-Cut bounds (3.5) and (3.6) are denoted as *linear BC-bounds*, (3.7) and (3.8) as *quadratic BC-bounds*. The optional application of sequential quadratic programming starting from the solution of the SDP relaxation is abbreviated as *SQP*. Given a numerical solution x of an equality and inequality constrained POP, its *scaled feasibility error* is given by

$$\epsilon_{\mathrm{sc}} = \min \left\{ - \mid h_i(x)/\sigma_i(x) \mid,\ \min \left\{ g_j(x)/\hat{\sigma}_j(x), 0 \right\}\ \forall\, i, j \right\},$$

where h_i $(i = 1, \ldots, k)$ denote the equality constraints, g_j $(j = 1, \ldots, l)$ the inequality constraints, and σ_i and $\hat{\sigma}_j$ are the maxima of the monomials in the corresponding polynomials h_i and g_j at x, respectively. Note, an equality and inequality constrained POP is a special case of the POP (2.1), if we define $f_i := g_i$ $(i = 1, \ldots, l)$, $f_i := h_i$ $(i = l+1, \ldots, l+k)$ and $f_i := -h_i$ $(i = k+l+1, \ldots, 2k+l)$. The value of the objective function at x is given by $f_0(x)$. Let N_C denote the number of constraints of a POP. 'OOM' as entry for the scaled feasibility error denotes the size of the SDP is too large to be solved by SeDuMi [11] and results in a memory error ('Out of memory'). A two-component entry for lbd or ubd indicates that the first component is used as a bound for the first $\frac{n}{2}$ variables and the second component for the remaining $\frac{n}{2}$ variables of the POP.

All numerical experiments are conducted on a LINUX OS with CPU 2.4 GHz and 8 Gb memory. The total processing time in seconds is denoted as t_C.

4.1 Randomly generated POPs

As a first class of test problems, consider randomly generated POPs with inequality or equality constraints. We are interested in the numerical performance of the sparse SDP relaxation for the corresponding QOPs for different substitution strategies and different choices of lower, upper and Branch-and-Cut bounds. We will focus on comparing strategies BI and BII as they yield POPs with a small number of additional variables.

For the random equality constrained POP *randEQ(7,3,5,8,0)* [12] of degree 8 with 7 variables, the size of the SDP relaxation sSDP_4 is described by the matrix A_{p} of size $[2870, 95628]$ with 124034 non-zero entries.

Substitution	$\tilde{n}$	size(A_q)	nnz(A_q)
AI	138	$[777, 6934]$	7106
AII	153	$[828, 6922]$	7116
BI	115	$[753, 5785]$	5934
BII	119	$[788, 6497]$	6653

Table 3: Size of the SDP relaxation sSDP_1 for QOPs from POP *randEQ(7,3,5,8,0)* with $n = 7$, respectively

This size is reduced substantially under each of the four substitution strategies, as can be seen in Table 3. In that table the matrix A_q in SeDuMi input format [11] and its number of nonzero entries nnz(A) describes the size of the sparse SDP relaxation. The reduction of the size of the SDP relaxation results in reducing the total processing time t_C by two magnitudes, as can be seen in Table 4.

Substitution	SQP	BC-bounds	(lbd, ubd)	ω	n or $\tilde{n}$	N_C	ϵ_{sc}	$f_0(x)$	t_C
-	no	none	$(-\infty, \infty)$	4	7	4	6e-11	-0.706	333
-	yes	none	$(-\infty, \infty)$	4	7	4	7e-18	-0.708	334
AI	no	none	(-1,1)	1	138	135	1e-13	-0.508	3
AII	no	none	(-1,1)	1	153	150	1e-13	-0.517	3
BI	no	none	(-1,1)	1	115	112	1e-13	-0.567	2
BII	no	none	(-1,1)	1	119	116	1e-13	-0.455	3
BI	yes	none	(-1,1)	1	115	112	7e-18	-0.708	3
BI	no	none	(-0.5,0.5)	1	115	112	9e-14	-0.706	3
BI	no	none	(-0.3,0.3)	1	115	112	1e-13	-0.708	2

Table 4: Results for SDP relaxation of *randEQ(7,3,5,8,0)*

Moreover, as reported in Table 4, the performance of AI, AII, BI and BII is similar with the one of BI being slightly better than the others. In this example with few equality constraints, feasibility of the solutions is east to obtain but optimality of the solutions requires additional techniques as SQP. Note, the optimality of the solutions improves significantly if the lower and upper bounds are chosen tighter. When chosen sufficiently tight, optimality of the solutions can be achieved without applying SQP.

The results for the inequality constrained POP *randINEQ(8,4,6,8,0)* [12] with $\omega_{\max} = 4$ and 8 variables are given in Table 5. In the column for (lbd, ubd) the entries $(-0.5, 0.5)^{\star}$ and $(-0.5, 0.5)^{\star\star}$ denote $\mathrm{ubd}_2 = 0.75 \neq 0.5$ and $(\mathrm{ubd}_2, \mathrm{ubd}_5) = (0.75, 0) \neq (0.5, 0.5)$, respectively. By imposing linear Branch-and-Cut bounds feasibility can be achieved and by choosing tighter lower and upper bounds the objective value of the approximative solution is improved. Though we did not achieve the optimal value attained by the solution of the SDP relaxation for the original POP, it seems reasonable to expect that successively tightening the bounds further yields a feasible solution with optimal objective value. As in the previous example the total processing time could be reduced by two magnitudes.

Substitution	SQP	BC-bounds	(lbd, ubd)	ω	n or $\tilde{n}$	N_C	ϵ_{sc}	$f_0(x)$	t_C
-	no	none	$(-\infty, \infty)$	4	8	3	0	-1.5	1071
BI	no	none	$(-0.75, 0.75)$	1	239	234	-1.3	-0.9	14
BI	no	linear	$(-0.75, 0.75)$	1	239	680	0	-0.6	17
BI	no	linear	$(-0.5, 0.5)^{\star}$	1	239	680	0	-0.8	17
BI	no	linear	$(-0.5, 0.5)^{\star\star}$	1	239	680	0	-1.2	16

Table 5: Results for SDP relaxation of *randINEQ(8,4,6,8,0)*

4.2 BroydenBand

Another test problem is the *BroydenBand(n)* problem [12]. It is an unconstrained POP of degree 6 and dimension n, and its global minimum is 0. Numerical results are given in Table 6. It is interesting to observe that tight lower and upper bounds, Branch-and-Cut bounds and applying sequential quadratic programming are crucial to obtain the global minimum by solving the sparse SDP relaxation for the QOP. In fact, when applying substitution strategy BI only imposing quadratic Branch-and-Cut bounds yields the global minimum, whereas in the case of applying BII Branch-and-Cut bounds are not necessary to obtain the global minimum. Note, the total processing time is reduced from around 1300 seconds to less than 5 seconds.

Substitution	SQP	BC-bounds	(lbd, ubd)	ω	n or $\tilde{n}$	N_C	$f_0(x)$	t_C
-	no	none	$(-\infty, +\infty)$	3	20	0	5e-9	1328
BI	yes	none	(-1, 1)	1	80	60	3	3
BI	yes	linear	(-1, 1)	1	80	140	3	3
BI	yes	quadratic	(-1, 1)	1	80	1284	3	4
BI	yes	none	(-0.75, 0)	1	80	60	3	4
BI	yes	linear	(-0.75, 0)	1	80	140	3	5
BI	yes	quadratic	(-0.75, 0)	1	80	1284	6e-8	5
BII	yes	none	(-1, 1)	1	60	40	3	4
BII	yes	linear	(-1, 1)	1	60	100	3	4
BII	yes	quadratic	(-1, 1)	1	60	1244	3	4
BII	yes	none	(-0.75, 0)	1	60	40	1e-10	5
BII	yes	linear	(-0.75, 0)	1	60	100	1e-6	4
BII	yes	quadratic	(-0.75, 0)	1	60	1244	2e-7	5

Table 6: Results for SDP relaxation for *BroydenBand(20)*

4.3 Polynomial optimization problems derived from partial differential equations

An important class of large scale polynomial optimization problems of higher degree is derived from discretizing systems of partial differential equations [9]. Many POPs of this class are of degree 3, but as the number of their constraints is in the same order as the number of variables, transformation into QOPs results in SDP relaxations of vastly reduced size. The structure of the higher degree monomial set of these POPs of degree three is such that there is an unique way to transform them into QOPs. Therefore, we examine the impact of lower, upper and Branch-and-Cut bounds and not the choice of the substitution strategy.

POP	n	$\tilde{n}$	ω_p	size(A_{p})	nnz(A_{p})	ω_q	size(A_{q})	nnz(A_{q})
pdeBifurcation(6)	36	72	2	$[2186, 17605]$	23801	1	$[422, 4039]$	4174
pdeBifurcation(10)	100	200	2	$[16592, 139245]$	189737	1	$[1643, 18646]$	19039
pdeBifurcation(14)	196	392	2	$[454497, 3822961]$	5208475	1	$[4126, 45189]$	46000
Mimura(50)	100	150	2	$[3780, 31258]$	39068	1	$[690, 5728]$	6078
Mimura(50)	100	150	3	$[19300, 280007]$	354067	2	$[7223, 76383]$	91755
Mimura(100)	200	300	3	$[39100, 565357]$	713767	2	$[14623, 155183]$	186155
Mimura(100)	200	300	2	$[7630, 63158]$	78818	2	$[1390, 11628]$	12328
StiffDiff(6,12)	144	216	2	$[18569, 163162]$	219020	1	$[878, 6700]$	7402
ginzOrDiri(9)	162	324	2	$[74628, 666987]$	906558	1	$[4567, 49305]$	50233
ginzOrNeum(11)	242	484	2	$[166092, 1451752]$	2504418	1	$[8063, 96367]$	97776

Table 7: Size of the SDP relaxation for POP and QOP, respectively

Consider the POPs in Table 7, where ω_p and ω_q the relaxation order of sSDP_ω for POP and QOP, to demonstrate the reduction of the size of the SDP relaxation described by the size of the matrix A in SeDuMi input format [11] and its number of nonzero entries $\mathrm{nnz}(A)$. Thus, the SDP relaxations for the QOPs can be solved in vastly shorter time than the one for the original POPs. The computational results of the original SDP relaxation and the SDP relaxation of the QOPs for different lower, upper and Branch-and-Cut bounds are reported in Table 8 for the POP *pdeBifurcation*. In this example the accuracy of the sparse SDP relaxation for the QOP is improved by tightening the upper bounds for the components of the variable $\tilde{x}$ in QOP. Also, the additional application of SQP improves the accuracy a lot. Additional Branch-and-Cut bounds seem to have no impact on the quality of the solution. The total processing time is reduced substantially under the transformation. The original sparse SDP relaxation for *pdeBifurcation(14)* of dimension 200 cannot be solved in SeDuMi due to memory error, but the SDP relaxation for the corresponding QOP with tight upper bounds can be solved in 100 seconds.

POP	Substitution	SQP	BC-bounds	ubd	ω	n or $\tilde{n}$	ϵ_{sc}	$f_0(x)$	t_C
pdeBifurcation(6)	-	no	none	0.99	2	36	8e-11	-9.0	14
pdeBifurcation(6)	AI	no	none	0.99	1	72	9.6e-2	-22.1	2
pdeBifurcation(6)	AI	no	linear	0.99	1	72	9.6e-2	-22.1	2
pdeBifurcation(6)	AI	no	quadratic	0.99	1	72	9.6e-2	-22.1	2
pdeBifurcation(6)	AI	yes	none	0.99	1	72	7.3e-9	-9.0	5
pdeBifurcation(6)	AI	no	none	0.45	1	72	1.5e-2	-9.5	1
pdeBifurcation(6)	AI	yes	none	0.45	1	72	1.4e-11	-9.0	2
pdeBifurcation(10)	-	no	none	0.99	2	100	3.1e-10	-21.6	2159
pdeBifurcation(10)	AI	no	none	0.99	1	200	4.7e-2	-56.0	20
pdeBifurcation(10)	AI	yes	none	0.99	1	200	2.7e-13	-21.6	66
pdeBifurcation(10)	AI	no	none	0.45	1	200	6.4e-3	-23.2	13
pdeBifurcation(10)	AI	yes	none	0.45	1	200	1e-11	-21.6	22
pdeBifurcation(14)	-	no	none	0.99	1	196	OOM	-	-
pdeBifurcation(14)	AI	no	none	0.99	1	392	2.4e-2	-103.1	90
pdeBifurcation(14)	AI	yes	none	0.99	1	392	7.9e-14	-39.9	418
pdeBifurcation(14)	AI	no	none	0.45	1	392	3.6e-3	-43.1	85
pdeBifurcation(14)	AI	yes	none	0.45	1	392	5.2e-11	-39.9	107

Table 8: Results for SDP relaxation for POP *pdeBifurcation* with lbd=0

In the case of POP *Mimura(50)*, c.f. Table 9, quadratic Branch-and-Cut bounds are necessary in addition to applying SQP, in order to obtain an accurate approximate solution of the global minimizer. In the POPs in Table 10 it is sufficient to apply SQP starting from the solution of the sparse SDP relaxation for the QOP. In that case the total processing time t_C can be reduced by up to two magnitudes. Furthermore, the original SDP relaxation for *ginzOrDiri(9)* and *ginzOrDiri(13)* is too large to be solved, whereas the SDP relaxations for the QOPs are tractable.

POP	Substitution	SQP	BC-bounds	ubd	ω	n or $\tilde{n}$	ϵ_{sc}	$f_0(x)$	t_C
Mimura(50)	-	no	none	$[11, 14]$	2	100	1.8e-1	-899	20
Mimura(50)	-	yes	none	$[11, 14]$	2	100	4.1e-9	-701	31
Mimura(50)	AI	no	none	$[11, 14]$	1	150	6.1e-1	-1067	2
Mimura(50)	AI	yes	none	$[11, 14]$	1	150	5.1e-3	-731	163
Mimura(50)	AI	no	quadratic	$[11, 14]$	1	150	3.3e-1	-1017	2
Mimura(50)	AI	yes	quadratic	$[11, 14]$	1	150	1.0e-13	-719	16
Mimura(100)	-	no	none	$[11, 14]$	3	200	4.5e-2	-733	532
Mimura(100)	-	yes	none	$[11, 14]$	3	200	2.0e-11	-712	557

Table 9: Results for SDP relaxation for POP *Mimura* with lbd $= [0, 0]$

The POP *ginzOrNeum* in Table 11 is another example where the global optimizer can be found in a processing time reduced by a factor 100, if the lower bounds lbd and upper bounds ubd are chosen sufficiently tight and SQP is applied.

POP	Substitution	SQP	ubd	ω	n or $\tilde{n}$	$\epsilon_{\rm sc}$	$f_0(x)$	t_C
ginzOrDiri(5)	-	no	0.6	2	50	6e-6	-25	598
ginzOrDiri(5)	-	yes	0.6	2	50	4e-15	-25	598
ginzOrDiri(5)	AI	no	0.6	1	100	3e-1	-100	7
ginzOrDiri(5)	AI	yes	0.6	1	100	4e-11	-22	10
ginzOrDiri(9)	-	no	0.6	2	162	OOM	-	-
ginzOrDiri(9)	AI	no	0.6	1	324	1e-1	-324	144
ginzOrDiri(9)	AI	yes	0.6	1	324	6e-12	-72	185
ginzOrDiri(13)	-	no	0.6	2	338	OOM	-	-
ginzOrDiri(13)	AI	yes	0.6	1	676	7e-9	-158	1992
StiffDiff(4,8)	-	yes	5	2	64	2e-11	-32	54
StiffDiff(4,8)	AI	yes	5	2	96	7e-10	-32	4
StiffDiff(6,12)	-	yes	5	1	144	4e-9	-71	1008
StiffDiff(6,12)	AI	yes	5	1	216	8e-10	-71	48

Table 10: Results for SDP relaxation for POP *ginzOrDiri* with lbd =0 and *StiffDiff* with lbd=0

POP	Substitution	SQP	lbd	ubd	ω	n	$\epsilon_{\rm sc}$	$f_0(x)$	t_C
ginzOrNeum(5)	-	no	$[0,0]$	$[4,2]$	2	50	2.6	-47	448
ginzOrNeum(5)	-	yes	$[0,0]$	$[4,2]$	2	50	2e-13	-45	449
ginzOrNeum(5)	AI	no	$[0,0]$	$[4,2]$	1	100	24	-100	9
ginzOrNeum(5)	AI	yes	$[0,0]$	$[4,2]$	1	100	8e-10	-45	10
ginzOrNeum(5)	-	no	$[1,0.5]$	$[4,1.5]$	2	50	1e-1	-45	582
ginzOrNeum(5)	-	yes	$[1,0.5]$	$[4,1.5]$	2	50	2e-13	-45	583
ginzOrNeum(5)	AI	no	$[1,0.5]$	$[4,1.5]$	1	100	6e-2	-57	6
ginzOrNeum(5)	AI	yes	$[1,0.5]$	$[4,1.5]$	1	100	4e-10	-45	7
ginzOrNeum(11)	-	no	$[1,0.5]$	$[4,1.5]$	2	242	OOM		
ginzOrNeum(11)	AI	no	$[1,0.5]$	$[4,1.5]$	1	484	4e-2	-263	740
ginzOrNeum(11)	AI	yes	$[1,0.5]$	$[4,1.5]$	1	484	5e-11	-207	748

Table 11: Results for SDP relaxation for POP *ginzOrNeum*

5 Conclusion

Large scale polynomial optimization problems (POP) of higher degree arise in a variety of areas and efficient methods to find their global optimizers are in high demand. Based on the sparse SDP relaxations for POPs due to Lasserre [6, 7] and Waki et al. [12], we proposed four different heuristics to transform a general POP into an equivalent quadratic optimization problem (QOP). The advantage of this transformation is that the sparse SDP relaxation of order one can be applied to the QOP. The sparse SDP relaxation of order one is of vastly smaller size than the sparse SDP relaxation of minimal order $\omega_{\max}$ for the original POP. By solving the sparse SDP relaxation of the QOP, approximates to the global minimizer of a large scale POP of higher degree can be derived. The reduction of the SDP relaxation and the gain in numerical tractability come at the cost of deteriorating feasibility and optimality errors of the approximate solution obtained by solving the SDP relaxation. In general the SDP relaxation of order one for the QOP is weaker than the SDP relaxation of order $\omega_{\max}$ for the original POP. We discussed how to overcome this difficulty by imposing tighter lower and upper bounds for the components of the n-dimensional variable of a POP, by adding linear or quadratic Branch-and-Cut bounds, and by applying local convergent optimization methods such as sequential quadratic programming (SQP) to the POP starting from the solution provided by the SDP relaxation for the QOP. The proposed heuristics have been demonstrated with success on various medium and large-scale POPs. We have seen that imposing additional Branch-and-Cut bounds was necessary to yield accurate approximations to the global optimizer for some problems. Nevertheless, for most problems it was crucial to choose the lower and upper bounds for the variable x sufficiently tight and to apply SQP to obtain a highly accurate approximation of the global optimizer. The total processing time could be reduced by up to three magnitudes for the problems we tested. For these reasons we think the proposed technique is promising to find first approximate solutions for POPs, whose size is too large to be solved by the more precise, original SDP relaxations. Particular classes of problems where this technique is interesting to apply are POPs derived from discretizing partial differential equations.

As a future problem it remains to tighten the sparse SDP relaxations of order one for the QOP further, in order to increase the accuracy of their solutions. In that respect, it is desirable to find a systematic approach to tighten lower and upper bounds successively, without shrinking the optimal set of the POP.

References

[1] J.R.S. Blair and B. Peyton, An introduction to chordal graphs and clique trees, Graph Theory and Sparse Matrix Computation, Springer Verlag (1993), pp. 1-29.

[2] P.T. Boggs and J.W. Tolle, Sequential Quadratic Programming, Acta Numerica 4 (1995), pp. 1-50.

[3] M. Fukuda and M. Kojima, Branch-and-Cut Algorithms for the Bilinear Matrix Inequality Eigenvalue Problem, Computational Optimization and Applications, 19 (2001), pp. 79-105.

[4] S. Kim and M. Kojima, Exact solutions of some nonconvex quadratic optimization problems via SDP and SOCP relaxations, Computational Optimization and Applications, 26 (2003), pp. 143-154.

[5] M. Kojima, S. Kim and H. Waki, Sparsity in sums of squares of polynomials, Mathematical Programming, 103 (2005) pp. 45-62.

[6] J.B. Lasserre, Global optimization with polynomials and the problem of moments, SIAM Journal on Optimization, 11 (2001), pp. 796-817.

[7] J.B. Lasserre, Convergent SDP-Relaxations in Polynomial Optimization with Sparsity, SIAM Journal on Optimization, 17 (2006), No. 3, pp. 822-843.

[8] M. Laurent, Sums of squares, moment matrices and optimization over polynomials, Emerging Applications of Algebraic Geometry, Vol. 149 of IMA Volumes in Mathematics and its Applications (2009), M. Putinar and S. Sullivant (eds.), Springer, pp. 157-270.

[9] M. Mevissen, M. Kojima, J. Nie and N. Takayama, Solving partial differential equations via sparse SDP relaxations, Pacific Journal of Optimization, 4 (2008), No. 2, pp. 213-241.

[10] J. Nocedal and S.J. Wright, Numerical Optimization, Series in Operations Research, Springer, New York 2006.

[11] J.F. Sturm, *SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*, Optimization Methods and Software, 11 and 12 (1999), pp. 625-653.

[12] H. Waki, S. Kim, M. Kojima and M. Muramatsu, Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity, SIAM Journal of Optimization 17 (2006) 218-242.