ISSN 1342-2804

Research Reports on Mathematical and Computing Sciences

Sums of Squares and Semidefinite Programming Relaxations for Polynomial Optimization Problems with Structured Sparsity

Hayato Waki, Sunyoung Kim, Masakazu Kojima and Masakazu Muramatsu

October 2004, B–411
Revised February 2005

Department of
Mathematical and
Computing Sciences
Tokyo Institute of Technology

SERIES B: Operations Research

B-411 Sums of Squares and Semidefinite Programming Relaxations for Polynomial Optimization Problems with Structured Sparsity
Hayato Waki*, Sunyoung Kim†, Masakazu Kojima‡, Masakazu Muramatsu♯
October 2004. Revised February 2005

Abstract. Unconstrained and inequality constrained sparse polynomial optimization problems (POPs) are considered. A correlative sparsity pattern graph is defined to find a certain sparse structure in the objective and constraint polynomials of a POP. Based on this graph, sets of supports for sums of squares (SOS) polynomials that lead to efficient SOS and semidefinite programming (SDP) relaxations are obtained. Numerical results from various test problems are included to show the improved performance of the SOS and SDP relaxations.

Key words.

Polynomial optimization problem, sparsity, global optimization, Lagrangian relaxation, Lagrangian dual, sums of squares optimization, semidefinite programming relaxation

* Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152-8552 Japan. *Hayato.Waki@is.titech.ac.jp*

† Department of Mathematics, Ewha Women's University, 11-1 Dahyun-dong, Sudaemoon-gu, Seoul 120-750 Korea. A considerable part of this work was conducted while this author was visiting Tokyo Institute of Technology. Research supported by KRF 2003-041-C00038. *skim@ewha.ac.kr*

‡ Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, 2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152-8552 Japan. Research supported by Grant-in-Aid for Scientific Research on Priority Areas 16016234. *kojima@is.titech.ac.jp*

♯ Department of Computer Science, The University of Electro-Communications, Chofugaoka, Chofu-Shi, Tokyo 182-8585 Japan. Research supported in part by Grant-in-Aid for Young Scientists (B) 15740054. *muramatu@cs.uec.ac.jp*

1 Introduction

Polynomial optimization problems (POPs) arise from various applications in science and engineering. Recent developments [11, 17, 19, 20, 23, 28, 29, 32] in semidefinite programming (SDP) and sums of squares (SOS) relaxations for POPs have attracted a lot of research from diverse directions. These relaxations have been extended to polynomial SDPs [13, 15, 18] and POPs over symmetric cones [21]. In particular, SDP and SOS relaxations have been popular for their theoretical convergence to the optimal value of a POP [23, 28]. From a practical point of view, improving the computational efficiency of SDP and SOS relaxations using the sparsity of polynomials in POPs has become an important issue [17, 20].

A polynomial f in real variables $x_1, x_2, \ldots, x_n$ of a positive degree d can have all monomials of the form $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$ with nonnegative integers α_i $(i = 1, 2, \ldots, n)$ such that $\alpha_i \geq 0$ and $\sum_{i=1}^n \alpha_i \leq d$; all monomials of different form add up to $\binom{n+d}{d}$. We call such a polynomial *fully dense*. When we examine polynomials in POPs from applications, we notice in many cases that they are *sparse* polynomials having a few or some of all possible monomials as defined in [20]. The sparsity provides a computational edge if it is handled properly when deriving SDP and SOS relaxations. More precisely, taking advantage of the sparsity of POPs is essential to obtaining an optimal value of a POP by applying SDP and SOS relaxations in practice. Without exploiting the sparsity of POPs, the size of the POPs that can be solved is very limited.

For sparse POPs, generalized Lagrangian duals and their SOS relaxations were proposed in [17]. The SOS relaxations are derived using SOS polynomials for the Lagrangian multipliers with similar sparsity to the associated constraint polynomials, and then converted into equivalent SDPs. As a result, the size of the resulting SOS and SDP relaxations is reduced and computational efficiency is improved. Theoretical convergence to the optimal value of a POP, however, is not guaranteed. This approach is shown to have an advantage in implementation over the SDP relaxation given in [23] whose size depends only on the degrees of objective and constraint polynomials of the POP. This is because SOS polynomials can be freely chosen for Lagrangian multipliers taking account of the sparsity of objective and constraint polynomials.

The aim of this paper is to present practical SOS and SDP relaxations for a sparse POP and show their performance for various test problems. The framework of SOS and SDP relaxations presented here are based on the one proposed in the paper [17]. However, the sparsity of a POP is defined more precisely to find a structure in the polynomials in the variables $x_1, x_2, \ldots, x_n$ of the POP and to derive sparse SOS and SDP relaxations accordingly. In particular, we introduce *correlative sparsity*, which is a special case of the sparsity [20] mentioned above; the correlative sparsity implies the sparsity, but the converse does not hold. The correlative sparsity is described in terms of an $n \times n$ symmetric matrix $\boldsymbol{R}$, which we call the *correlative sparsity pattern matrix* (*csp matrix*) of the POP. Each element R_{ij} of the csp matrix $\boldsymbol{R}$ is either 0 or $\star$ representing a nonzero value. We assign $\star$ to every diagonal element R_{ii} $(i = 1, 2, \ldots, n)$, and also to each off-diagonal element $R_{ij} = R_{ji}$ $(1 \leq i < j \leq n)$ if and only if either (i) the variables x_i and x_j appear simultaneously in a term of the objective function, or (ii) they appear in an inequality constraint. The csp matrix $\boldsymbol{R}$ constructed in this way represents the sparsity pattern of the Hessian matrix of the generalized Lagrangian function of the paper [17] (or the Hessian matrix of the objective function in unconstrained cases) except for the diagonal elements; some diagonal elements of the Hessian matrix may vanish while $R_{ii} = \star$ ($i = 1, 2, \ldots, n$) by definition. We say that the POP is *correlatively sparse* if the csp matrix $\boldsymbol{R}$ (or the Hessian matrix of the generalized Lagrangian function) is sparse.

From the csp matrix $\boldsymbol{R}$, it is natural to induce graph $G(N, E)$ with the node set $N = \{1, 2, \ldots, n\}$ and the edge set $E = \{\{i, j\} : R_{ij} = \star,\ i < j\}$ corresponding to the nonzero off-diagonal elements of $\boldsymbol{R}$. We call $G(N, E)$ the *correlation sparsity pattern graph* (*csp graph*) of the POP. We employ some results of graph theory regarding maximal cliques of chordal graphs [2]. A key idea in this paper is to use the maximal cliques of a chordal extension of the csp graph $G(N, E)$ to construct sets of supports for a sparse SOS relaxation. This idea is motivated by the recent work [8] that proposed positive semidefinite matrix completion techniques for exploiting sparsity in primal-dual interior-point methods for SDPs.

A simple example is a POP with a separable objective polynomial consisting of a sum of n polynomials in a single variable x_i ($i = 1, 2, \ldots, n$) and $n - 1$ inequality constraints each of which contains two variables x_i and x_{i+1} ($i = 1, 2, \ldots, n-1$). In this case, the csp matrix $\boldsymbol{R}$ is represented as a tridiagonal matrix and the csp graph $G(N, E)$ is a chordal graph with $n-1$ maximal cliques with 2 nodes, and the size of the proposed SOS and SDP relaxations become considerably smaller than the one obtained from the dense SDP relaxation given in [23].

The computational efficiency of the proposed sparse SOS and SDP relaxations depends on how sparse a chordal extension of the csp graph is. We note that the following two conditions are equivalent: (i) a chordal extension of the csp graph is sparse, (ii) a sparse Cholesky factorization can be applied to the Hessian matrix of the generalized Lagrangian function or the Hessian matrix of the objective function in unconstrained problems. When we compare the condition (ii) with the standard condition of traditional numerical methods, such as Newton's method for convex optimization, to be efficient for large scale problems, there exists a difference between the generalized Lagrangian function and the Lagrangian function in the Lagrange multipliers. SOS polynomials are the Lagrangian multipliers in the former whereas nonnegative real numbers in the latter. If a linear inequality constraint is involved in the POP, which may be the simplest constraint, it is multiplied by a SOS polynomial in the former. As a result, the Hessian matrix of the former can become denser than the Hessian matrix of the latter. In this sense, the condition (ii) in the proposed sparse SOS and SDP relaxations is a stronger requirement on the sparsity in the POP than the standard condition for traditional numerical methods. This stronger requirement, however, can be justified if we understand the study of nonconvex and large scale POPs in global optimization as a more complicated issue.

Theoretically, the proposed sparse SOS and SDP relaxations are not guaranteed to generate lower bounds of the same quality as the dense SDP relaxation [23] for general POPs. Practical experiences, however, show us that the performance gap between the two relaxations is small as we observe from numerical experiments presented in Section 6. In particular, the definition of a structured sparsity based on the csp matrix $\boldsymbol{R}$ and the csp graph $G(N, E)$ enables us to achieve the same quality of lower bounds for quadratic optimization problems (QOPs) where all polynomials in the objective function and constraints are quadratic. More precisely, the proposed sparse SOS and SDP relaxations of order 1 obtain lower bounds of the same quality as the dense SDP relaxation of order 1, as shown in Section 5.4. Here the latter SDP relaxation of order 1 corresponds to classical SDP relaxation [7, 9, etc.], which have been widely studied for QOPs including the maxcut problem.

This motivates the use of the csp matrix and graph for structured sparsity in the derivation of SOS and SDP relaxations.

The remaining of the paper is organized as follows. After introducing basic notation and symbols on polynomials, we define sums of squares polynomials in Section 2. In Section 3, we first describe dense SOS relaxation of unconstrained POPs, and then sparse SOS relaxation. We show how a csp matrix is defined from a given unconstrained POP, and how a sparse SOS relaxation is constructed using the maximal cliques of a chordal extension of a csp graph induced from the csp matrix. Section 4 contains the description of a SOS relaxation of an inequality constrained POP with a structured sparsity characterized by a csp matrix and a csp graph. We introduce a generalized Lagrangian dual for the inequality constrained POP and a sparse SOS relaxation. Section 5 discusses some additional techniques which enhance the practical performance of the sparse SOS relaxation such as computing optimal solutions, handling equality constraints and scaling. Section 6 includes numerical results on various test problems. We show the proposed sparse SOS and SDP relaxations exhibit much better performance in practice. Finally, we give concluding remarks in Section 7.

2 Preliminaries

We first describe the representation of polynomials and sums of squares of polynomials in this section.

2.1 Polynomials

Let $\mathbb{R}$ be the set of real numbers, and $\mathbb{Z}_+$ the set of nonnegative integers. We use $\mathbb{R}^n$ and $\mathbb{Z}^n_+$ to denote the n-dimensional Euclidean space and the set of nonnegative integer vectors in $\mathbb{R}^n$. $\mathbb{R}[\boldsymbol{x}]$ means the set of real valued polynomials in x_i $(i = 1, 2, \ldots, n)$. Each polynomial $f \in \mathbb{R}[\boldsymbol{x}]$ is represented as

$$f(\boldsymbol{x}) \quad = \quad \sum_{\boldsymbol{\alpha}\in\mathcal{F}} c(\boldsymbol{\alpha})\boldsymbol{x}^{\boldsymbol{\alpha}} \ (\forall \boldsymbol{x} \in \mathbb{R}^n)$$

for some nonempty finite subset $\mathcal{F} \subset \mathbb{Z}^n_+$ and some real numbers $c(\boldsymbol{\alpha})$ $(\boldsymbol{\alpha} \in \mathcal{F})$. Here $\boldsymbol{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$; we assume that $\boldsymbol{x}^{\boldsymbol{0}} = 1$. The *support* of f is defined by

$$\mathrm{supp}(f) = \{\boldsymbol{\alpha} \in \mathcal{F} : c(\boldsymbol{\alpha}) \neq 0\} \subset \mathbb{Z}^n_+,$$

and the *degree* of $f \in \mathbb{R}[\boldsymbol{x}]$ by

$$\deg(f) \quad = \quad \max\left\{\sum_{i=1}^{n} \alpha_i : \boldsymbol{\alpha} \in \mathrm{supp}(f)\right\}.$$

For every nonempty finite set $\mathcal{G} \subset \mathbb{Z}^n_+$, $\mathbb{R}[\boldsymbol{x}, \mathcal{G}]$ denotes the set of polynomials in x_i $(i = 1, 2, \ldots, n)$ whose support is included in $\mathcal{G}$; $\mathbb{R}[\boldsymbol{x}, \mathcal{G}] = \{f \in \mathbb{R}[\boldsymbol{x}] : \mathrm{supp}(f) \subset \mathcal{G}\}$. Let $N = \{1, 2, \ldots, n\}$. Suppose that $\emptyset \neq C \subset N$ and $\omega \in \mathbb{Z}_+$. Consider the set

$$\mathcal{A}^C_\omega = \left\{\boldsymbol{\alpha} \in \mathbb{Z}^n_+ : \alpha_i = 0 \text{ if } i \notin C \ \text{ and } \sum_{i\in C} \alpha_i \leq \omega\right\}.$$

In the succeeding discussions on sparse SOS relaxations, the set $\mathcal{A}^C_\omega$ serves as the support of fully dense polynomials in x_i $(i \in C)$ whose degree is at most ω.

2.2 Sums of squares of polynomials

Let $\mathcal{G}$ be a nonempty finite subset of $\mathbb{Z}^n_+$. We denote $\mathbb{R}[\boldsymbol{x}, \mathcal{G}]^2$ the set of sums of squares of polynomials in $\mathbb{R}[\boldsymbol{x}, \mathcal{G}]$;

$$\mathbb{R}[\boldsymbol{x}, \mathcal{G}]^2 = \left\{ \sum_{i=1}^{q} g_i^2 : \exists q \in \mathbb{Z}_+, \ \exists g_i \in \mathbb{R}[\boldsymbol{x}, \mathcal{G}] \ (i = 1, 2, \ldots, q) \right\}.$$

By construction, we see that $\text{supp}(g) \subset \mathcal{G} + \mathcal{G}$ if $g \in \mathbb{R}[\boldsymbol{x}, \mathcal{G}]^2$. Here $\mathcal{G} + \mathcal{G}$ denotes the Minkovski sum of two $\mathcal{G}$'s; $\mathcal{G} + \mathcal{G} = \{\boldsymbol{\alpha} + \boldsymbol{\beta} : \boldsymbol{\alpha} \in \mathcal{G}, \ \boldsymbol{\beta} \in \mathcal{G}\}$. Specifically, we observe that $\mathcal{A}^C_\omega + \mathcal{A}^C_\omega = \mathcal{A}^C_{2\omega}$ $(N \supset \forall C \neq \emptyset, \ \forall \omega \in \mathbb{Z}_+)$.

Let $\mathbb{R}^{\mathcal{G}}$ denote the $|\mathcal{G}|$-dimensional Euclidean space whose coordinates are indexed by $\boldsymbol{\alpha} \in \mathcal{G}$. Each vector of $\mathbb{R}^{\mathcal{G}}$ is denoted as $\boldsymbol{w} = (w_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{G})$. Although the order of the coordinates of $\boldsymbol{w} = (w_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{G})$ is not relevant in the succeeding discussions, we may assume that the coordinates are arranged according to the lexicographically increasing order; if $\boldsymbol{0} \in \mathcal{G}$ then $w_{\boldsymbol{0}}$ is the first element of $\boldsymbol{w} \in \mathbb{R}^{\mathcal{G}}$. We use the symbol $\mathcal{S}(\mathcal{G})$ for the set of $|\mathcal{G}| \times |\mathcal{G}|$ symmetric matrices with coordinates $\boldsymbol{\alpha} \in \mathcal{G}$; each $\boldsymbol{V} \in \mathcal{S}(\mathcal{G})$ has elements $V_{\boldsymbol{\alpha\beta}}$ $(\boldsymbol{\alpha} \in \mathcal{G}, \ \boldsymbol{\beta} \in \mathcal{G})$ such that $V_{\boldsymbol{\alpha\beta}} = V_{\boldsymbol{\beta\alpha}}$. Let $\mathcal{S}_+(\mathcal{G})$ denote the set of positive semidefinite matrices in $\mathcal{S}(\mathcal{G})$;

$$\boldsymbol{w}^T \boldsymbol{V} \boldsymbol{w} = \sum_{\boldsymbol{\alpha} \in \mathcal{G}} \sum_{\boldsymbol{\beta} \in \mathcal{G}} V_{\boldsymbol{\alpha\beta}} w_{\boldsymbol{\alpha}} w_{\boldsymbol{\beta}} \geq 0 \text{ for every } \boldsymbol{w} = (w_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \mathcal{G}) \in \mathbb{R}^{\mathcal{G}}.$$

The symbol $\boldsymbol{u}(\boldsymbol{x}, \mathcal{G})$ is used for the $|\mathcal{G}|$-dimensional column vector consisting of elements $\boldsymbol{x}^{\boldsymbol{\alpha}}$ $(\boldsymbol{\alpha} \in \mathcal{G})$, where we may assume that the elements $\boldsymbol{x}^{\boldsymbol{\alpha}}$ $(\boldsymbol{\alpha} \in \mathcal{G})$ are arranged in the column vector according to the lexicographically increasing order of $\boldsymbol{\alpha} \in \mathcal{G}$. Then, the sets $\mathbb{R}[\boldsymbol{x}, \mathcal{G}]^2$ can be rewritten as

$$\mathbb{R}[\boldsymbol{x}, \mathcal{G}]^2 = \left\{ \boldsymbol{u}(\boldsymbol{x}, \mathcal{G})^T \boldsymbol{V} \boldsymbol{u}(\boldsymbol{x}, \mathcal{G}) : \boldsymbol{V} \in \mathcal{S}_+(\mathcal{G}) \right\}. \tag{1}$$

For more details, see the papers [5, 28].

3 SOS relaxations of unconstrained polynomial optimization problems

In this section, we consider a POP

$$\text{minimize } f_0(\boldsymbol{x}) \text{ over } \boldsymbol{x} \in \mathbb{R}^n, \tag{2}$$

where $f_0 \in \mathbb{R}[\boldsymbol{x}]$ is represented as $f_0(\boldsymbol{x}) = \sum_{\boldsymbol{\alpha} \in \mathcal{F}_0} c_0(\boldsymbol{\alpha}) \boldsymbol{x}^{\boldsymbol{\alpha}}$ $(\forall \boldsymbol{x} \in \mathbb{R}^n)$ for some nonempty finite subset $\mathcal{F}_0$ of $\mathbb{Z}^n_+$ and some real numbers $c_0(\boldsymbol{\alpha})$ $(\boldsymbol{\alpha} \in \mathcal{F}_0)$. We assume that $c_0(\boldsymbol{\alpha}) \neq 0$ for every $\boldsymbol{\alpha} \in \mathcal{F}_0$; hence $\text{supp}(f_0) = \mathcal{F}_0$. Let ζ^* denote the optimal value of the POP (2); $\zeta^* = \inf \{f_0(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^n\}$. Throughout this section, we assume that $\zeta^* > -\infty$. Then $\deg(f_0)$ is an even integer, *i.e.*, $\deg(f_0) = 2\omega_0$ for some $\omega_0 \in \mathbb{Z}_+$. Let $\omega = \omega_0$. By Lemma in Section 3 of [31], we also know that $\mathcal{F}_0 = \text{supp}(f_0) \subset$ the convex hull of $\mathcal{F}^e_0$, where $\mathcal{F}^e_0 = \{\boldsymbol{\alpha} \in \mathcal{F}_0 : \alpha_i \text{ is an even nonnegative integer } (i = 1, 2, \ldots, n)\}$.

3.1 SOS relaxations

In this subsection, we describe the SOS relaxation of the POP (2) and how the size of the resulting SOS relaxation is reduced according to the papers [20, 31], which is an important issue in practice.

In order to apply the SOS relaxation to the POP (2), we first convert the problem into an equivalent problem

$$\text{maximize } \zeta \text{ subject to } f_0(\boldsymbol{x}) - \zeta \geq 0 \; (\forall \boldsymbol{x} \in \mathbb{R}^n). \tag{3}$$

It should be noted that only $\zeta \in \mathbb{R}$ is a variable and $\boldsymbol{x} \in \mathbb{R}^n$ is an index parameter to represent a continuum number of constraints $f_0(\boldsymbol{x}) - \zeta \geq 0$ $(\forall \boldsymbol{x} \in \mathbb{R}^n)$; hence the problem above is a semi-infinite programming problem.

If we replace the constraint $f_0(\boldsymbol{x}) - \zeta \geq 0$ $(\forall \boldsymbol{x} \in \mathbb{R}^n)$ in the problem (3) by an SOS constraint

$$f_0(\boldsymbol{x}) - \zeta \in \mathbb{R}[\boldsymbol{x}, \mathcal{A}^N_\omega]^2, \tag{4}$$

then, we obtain an SOS optimization problem

$$\text{maximize } \zeta \text{ subject to } f_0(\boldsymbol{x}) - \zeta \in \mathbb{R}[\boldsymbol{x}, \mathcal{A}^N_\omega]^2. \tag{5}$$

Note that if the SOS constraint (4) is satisfied, then the constraint of (3) holds, but the converse is not true because a polynomial which is nonnegative over $\mathbb{R}^n$ is not necessarily a sum of squares of polynomials in general [14]. Hence the optimal objective value of the SOS optimization problem does not exceed the common optimal value ζ^* of the POPs (2) and (3). Therefore, the SOS optimization problem (5) serves as a relaxation of the POP (2). We call the parameter $\omega \in \mathbb{Z}_+$ in (5) the *(relaxation) order*. We can rewrite the SOS constraint (4) using the relation (1) as

$$f_0(\boldsymbol{x}) - \zeta = \boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^N_\omega)^T \boldsymbol{V} \boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^N_\omega) \; (\forall \boldsymbol{x} \in \mathbb{R}^n) \text{ and } \boldsymbol{V} \in \mathcal{S}_+(\mathcal{A}^N_\omega). \tag{6}$$

We call a polynomial $f_0 \in \mathbb{R}[\boldsymbol{x}, \mathcal{A}^N_{2\omega}]$ *sparse* if the number of elements in its support $\mathcal{F}_0 = \text{supp}(f_0)$ is much smaller than the number of elements in $\mathcal{A}^N_{2\omega}$ that forms a support of fully dense polynomials in $\mathbb{R}[\boldsymbol{x}, \mathcal{A}^N_{2\omega}]$. When the objective function f_0 is a sparse polynomial in $\mathbb{R}[\boldsymbol{x}, \mathcal{A}^N_{2\omega}]$, the size of the SOS constraint (5) can be reduced by eliminating redundant elements from $\mathcal{A}^N_\omega$. In fact, by applying Theorem 1 of [31], $\mathcal{A}^N_\omega$ in the problem (5) can be replaced by

$$\mathcal{G}^0_0 = \left(\text{the convex hull of } \left\{\frac{\boldsymbol{\alpha}}{2} : \boldsymbol{\alpha} \in \mathcal{F}^e_0 \bigcup \{\boldsymbol{0}\}\right\}\right) \cap \mathbb{Z}^n_+ \subset \mathcal{A}^N_\omega.$$

Note that $\{\boldsymbol{0}\}$ is added as the support for the real number variable ζ.

A method that can further reduce the size of the SOS optimization problem by eliminating redundant elements in $\mathcal{G}^0_0$ was proposed by Kojima *et al* in [20]. We write the resulting SOS constraint from their method as

$$f_0(\boldsymbol{x}) - \zeta \in \mathbb{R}[\boldsymbol{x}, \mathcal{G}^*_0]^2, \tag{7}$$

where $\mathcal{G}^*_0 \subset \mathcal{G}^0_0 \subset \mathcal{A}^N_\omega$ denotes the set obtained by applying the method. Preliminary numerical results were presented in [20] for the problems with $|\mathcal{G}^*_0|$ significantly smaller than

$|\mathcal{G}_0^0|$. However, their method is not robust in the sense that the performance of the method is not effective for some problems as the followings, which may be considered practically important problems. If $\mathcal{F}_0 = \text{supp}(f_0) \subset \mathcal{A}_{2\omega}^N$ contains n vectors $2\omega \boldsymbol{e}^1,\ 2\omega_0 \boldsymbol{e}^2,\ \ldots,\ 2\omega \boldsymbol{e}^n$, or equivalently, if the objective polynomial function $f_0 \in \mathbb{R}[\boldsymbol{x}, \mathcal{A}_{2\omega}^N]$ involves n monomials $x_1^{2\omega},\ x_2^{2\omega},\ \ldots,\ x_n^{2\omega}$, then $\mathcal{G}_0^*$ becomes fully dense, *i.e.*, $\mathcal{G}_0^* = \mathcal{A}_\omega^N$. Specifically, even if the objective polynomial function $f_0 \in \mathbb{R}[\boldsymbol{x}]_{2\omega}$ is a separable polynomial of the form

$$f_0(\boldsymbol{x}) = \sum_{i=1}^n h_i(x_i) \ (\forall \boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n), \tag{8}$$

where each $h_i(x_i)$ denotes a polynomial in a single variable $x_i \in \mathbb{R}$ with $\deg(h_i(x_i)) = 2\omega$, we have that $\mathcal{G}_0^* = \mathcal{A}_\omega^N$. See Proposition 5.1 of [20]

3.2 An outline of sparse SOS relaxations

The focus of this subsection is on how we can deal with the weakness of the method in [20] mentioned in Section 3.1. We find a certain structure from the correlative sparsity of POPs, and propose a heuristic method for constructing smaller-sized SOS relaxations of POPs with the correlative sparsity. Using the structure obtained from the correlative sparsity, we generate multiple support sets $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_p \subset \mathbb{Z}_+^n$ such that

$$\mathcal{F}_0 \bigcup \{\boldsymbol{0}\} \subset \bigcup_{\ell=1}^p (\mathcal{G}_\ell + \mathcal{G}_\ell), \tag{9}$$

and replace the SOS constraint (7) by

$$f_0(\boldsymbol{x}) - \zeta \in \sum_{\ell=1}^p \mathbb{R}[\boldsymbol{x}, \mathcal{G}_\ell]^2, \tag{10}$$

where $\displaystyle\sum_{\ell=1}^p \mathbb{R}[\boldsymbol{x}, \mathcal{G}_\ell]^2 = \left\{ \sum_{\ell=1}^p h_\ell : h_\ell \in \mathbb{R}[\boldsymbol{x}, \mathcal{G}_\ell]^2 \ (\ell = 1, 2, \ldots, p) \right\}$. The support of the polynomial $f_0(\boldsymbol{x}) - \zeta$ on the left side of the constraint above is $\mathcal{F}_0 \bigcup \{\boldsymbol{0}\}$, while the support of each polynomial in $\displaystyle\sum_{\ell=1}^p \mathbb{R}[\boldsymbol{x}, \mathcal{G}_\ell]^2$ on the right side is contained in $\displaystyle\bigcup_{\ell=1}^p (\mathcal{G}_\ell + \mathcal{G}_\ell)$. Hence the inclusion relation (9) is necessary for the SOS constraint (10) to be feasible although it is not sufficient. If the size of each $\mathcal{G}_\ell$ is much smaller than the size of $\mathcal{G}_0^*$ and if the number of the support sets p is not large, the size of the SOS constraint (10) is smaller than the size of the SOS constraint (5).

For the problem where the polynomial objective function f_0 is given by (8), we have $p = n$ and $\mathcal{G}_\ell = \left\{ \rho \boldsymbol{e}^\ell : \rho = 0, 1, 2, \ldots, \omega \right\}$ $(\ell = 1, 2, \ldots, n)$. Here $\boldsymbol{e}^\ell \in \mathbb{R}^n$ denotes the ℓth unit vector with 1 at the ℓth coordinate and 0 elsewhere. The resulting SOS optimization problem inherits the separability from the separable polynomial objective function f_0, and is subdivided into n independent subproblems; each subproblem forms an SOS relaxation of the corresponding subproblem of the POP (2), minimizing $h_\ell(x_\ell)$ in a single variable.

3.3 Correlative sparsity pattern matrix

We consider a sparsity from cross terms $x_i x_j$ $(1 \leq i < j \leq n)$ in the objective polynomial f_0 of the unconstrained POP (2). The sparsity considered here is measured by the number of different kinds of the cross terms in the objective polynomial f_0. We will call this type of sparsity *correlative sparsity.* The correlative sparsity is represented with the $n \times n$ (symbolic, symmetric) *correlative sparsity pattern matrix* (abbreviated by *csp matrix*) $\boldsymbol{R}$ whose (i, j)th element R_{ij} is given by

$$R_{ij} \;=\; \begin{cases} \star & \text{if } i = j, \\ \star & \text{if } \alpha_i \geq 1 \;\text{ and } \alpha_j \geq 1 \;\text{ for some } \boldsymbol{\alpha} \in \mathcal{F}_0 = \text{supp}(f_0), \\ 0 & \text{otherwise} \end{cases}$$

$(i = 1, 2, \ldots, n,\ j = 1, 2, \ldots, n)$. Here $\star$ stands for some nonzero element. If the csp matrix $\boldsymbol{R}$ of f_0 is sparse, then f_0 is sparse as defined in [20]. But the converse is not true; for example, the polynomial $f_0(\boldsymbol{x}) = x_1^2 x_2^2 \cdots x_n^2 \in \mathbb{R}[\boldsymbol{x}, \mathcal{A}_{2n}^N]$ $(\forall \boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n)$ is sparse by the definition in [20] while its csp matrix is fully dense. We say that f_0 is *correlatively sparse* if the associated csp matrix is sparse. As was mentioned in Introduction, the correlative sparsity of an objective function $f_0(\boldsymbol{x})$ is equivalent to the sparsity of its Hessian matrix with some additional diagonal elements.

Let us consider a few examples. First, we observe that the csp matrix $\boldsymbol{R}$ becomes an $n \times n$ diagonal matrix in the case of the separable polynomial (8).

Suppose that

$$f_0(\boldsymbol{x}) = \sum_{i=1}^{n-1} \left(a_i x_i^4 + b_i x_i^2 x_{i+1} + c_i x_i x_{i+1} \right) \;\; (\forall \boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n) \tag{11}$$

where a_i, b_i and c_i are nonzero real numbers $(i = 1, 2, \ldots, n-1)$. Then, the csp matrix turns out to be the $n \times n$ tridiagonal matrix

$$\boldsymbol{R} \;=\; \begin{pmatrix} \star & \star & 0 & 0 & \ldots & 0 & 0 \\ \star & \star & \star & 0 & \ldots & 0 & 0 \\ 0 & \star & \star & \star & \ldots & 0 & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ 0 & 0 & & \star & \star & \star & 0 \\ 0 & 0 & & & \star & \star & \star \\ 0 & 0 & \ldots & \ldots & 0 & \star & \star \end{pmatrix}. \tag{12}$$

Next, consider

$$f_0(\boldsymbol{x}) = \sum_{i=1}^{n-1} \left(a_i x_i^4 + b_i x_i^2 x_n + c_i x_i x_n \right) \;\; (\forall \boldsymbol{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n), \tag{13}$$

where a_i, b_i and c_i are nonzero real numbers $(i = 1, 2, \ldots, n-1)$. In this case, we have

$$\boldsymbol{R} \;=\; \begin{pmatrix} \star & 0 & \ldots & 0 & \star \\ 0 & \star & \ldots & 0 & \star \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \ldots & \star & \star \\ \star & \star & \ldots & \star & \star \end{pmatrix}. \tag{14}$$

3.4 Correlative sparsity pattern graphs

We describe a method to determine the sets of supports $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_p$ for the target SOS relaxation (10) of the unconstrained POP (2). The basic idea is to use the structure of the csp matrix $\boldsymbol{R}$ and some results from graph theory.

From the csp matrix $\boldsymbol{R}$, the undirected graph $G(N, E)$ with

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

is called the *correlative sparsity pattern graph* (abbreviated as *csp graph*). Let $C_1, C_2, \ldots, C_p \subset N$ denote the maximal cliques of the csp graph $G(N, E)$. Then, choose the sets of supports $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_p$ such that $\mathcal{G}_\ell = \mathcal{A}^{C_\ell}_\omega$ $(\ell = 1, 2, \ldots, p)$. We show that the relation (9) holds. First, we observe that each $\mathcal{G}_\ell = \mathcal{A}^{C_\ell}_\omega$ contains $\mathbf{0} \in \mathbb{Z}^n_+$ by definition. Suppose that $\boldsymbol{\alpha} \in \mathcal{F}_0$. Then the set $C = \{i \in N : \alpha_i \geq 1\}$ forms a clique of the csp graph $G(N, E)$ since $\{i, j\} \in E$ for every pair i and j from the set C. Hence there exists an ℓ such that $C \subset C_\ell$. On the other hand, we know $\deg(f_0) = 2\omega$ by the assumption; hence $\sum_{i \in C} \alpha_i \leq 2\omega$. Therefore, we obtain that

$$\boldsymbol{\alpha} \in \mathcal{A}^{C}_{2\omega} \subset \mathcal{A}^{C_\ell}_{2\omega} = \mathcal{A}^{C_\ell}_\omega + \mathcal{A}^{C_\ell}_\omega \subset \bigcup_{\ell=1}^{p} (\mathcal{G}_\ell + \mathcal{G}_\ell)\,.$$

However, the method described above for choosing $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_p$ has a critical disadvantage since finding all maximal cliques of a graph is a difficult problem in general. In fact, finding a single maximum clique is an NP hard problem. To resolve this difficulty, we generate a chordal extension $G(N, E')$ of the csp graph $G(N, E)$ and use the extended csp graph $G(N, E')$ instead of $G(N, E)$. See [2] for chordal graphs and their basic properties.

Consequently, we obtain a sparse SOS relaxation of the POP (2):

$$\text{maximize} \quad \zeta \quad \text{subject to} \quad f_0(\boldsymbol{x}) - \zeta \in \sum_{\ell=1}^{p} \mathbb{R}[\boldsymbol{x}, \mathcal{A}^{C_\ell}_\omega]^2, \tag{15}$$

where C_ℓ $(\ell = 1, 2, \ldots, p)$ denote the maximal cliques of a chordal extension $G(N, E')$ of the csp graph $G(N, E)$. Some software packages [1, 16] are available to generate a chordal extension of a graph. One way of computing a chordal extension $G(N, E')$ of the csp graph $G(N, E)$ and the maximal cliques of the extension is to apply the MATLAB functions symmmd (the symmetric minimum degree ordering) or symamd (the symmetric approximate minimum degree permutation), and chol (the Cholesky factorization) to the csp matrix $\boldsymbol{R}$ with replacing the off-diagonal nonzero elements $R_{ij} = R_{ji} = \star$ by small random numbers and the diagonal elements R_{ii} by sufficiently large positive random numbers. We employed the MATLAB functions symamd and chol in the numerical experiments reported in Section 6. It should be noted that the number of the maximal cliques of $G(N, E')$ does not exceed n, which is equivalent to the number of nodes of the graph $G(N, E')$ as well as to the number of variables of the objective polynomial f_0.

In the case (8), the csp graph $G(N, E)$ has no edge, and every maximal clique consist of a single node. Hence, we have $p = n$ and $C_\ell = \{\ell\}$ $(\ell = 1, 2, \ldots, n)$. Either of the csp matrices given in (12) and (14) induces a chordal graph, and there is no need to extend. The maximal cliques are $C_\ell = \{\ell,\ \ell + 1\}$ $(\ell = 1, 2, \ldots, n - 1)$, and $C_\ell = \{\ell,\ n\}$ $(\ell = 1, 2, \ldots, n - 1)$, respectively.

3.5 Quadratic objective functions

In this subsection, we focus on the POP (2) with $\omega = \deg(f_0)/2 = 1$, *i.e.*, the unconstrained minimization of a quadratic objective function:

$$f_0(\boldsymbol{x}) \;=\; \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + 2\boldsymbol{q}^T\boldsymbol{x} + \gamma \; (\forall \boldsymbol{x} \in \mathbb{R}^n), \tag{16}$$

where $\boldsymbol{Q} \in \mathcal{S}^n$, $\boldsymbol{q} \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}$. In this case, we show that the proposed sparse SOS relaxation (15) using any chordal extension of the csp graph $G(N, E)$ attains the same optimal value as the dense SOS relaxation (5). This demonstrates an advantage of using the set of maximal cliques of a chordal extension of the csp graph $G(N, E)$ instead of the set of maximal cliques of $G(N, E)$ itself.

Recall that (5) is equivalent to (6). Suppose that $(\boldsymbol{V}, \zeta) \in \mathcal{S}_+(\mathcal{A}_1^N) \times \mathbb{R}$ is a feasible solution of (6). If we rewrite (6) for the quadratic objective function (16), we have

$$(1, \boldsymbol{x}^T) \begin{pmatrix} \gamma - \zeta & \boldsymbol{q}^T \\ \boldsymbol{q} & \boldsymbol{Q} \end{pmatrix} \begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix} = (1, \boldsymbol{x}^T)\boldsymbol{V} \begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix} \; (\forall \boldsymbol{x} \in \mathbb{R}^n) \;\text{ and } \boldsymbol{V} \in \mathcal{S}_+(\mathcal{A}_1^N).$$

Comparing the both sides of the equality above, we know the coefficient matrices coincide:

$$\begin{pmatrix} \gamma - \zeta & \boldsymbol{q}^T \\ \boldsymbol{q} & \boldsymbol{Q} \end{pmatrix} = \boldsymbol{V} \in \mathcal{S}_+(\mathcal{A}_1^N).$$

Hence $\boldsymbol{Q}$ needs to be positive semidefinite.

Let $\epsilon > 0$, and $\boldsymbol{I}$ denote the $n \times n$ identity matrix. Then, $\boldsymbol{Q} + \epsilon\boldsymbol{I}$ is positive definite and the csp matrix $\boldsymbol{R}$ for the quadratic function (16) has the same sparsity pattern as the matrix $\boldsymbol{Q} + \epsilon\boldsymbol{I}$:

$$R_{ij} = \begin{cases} \star & \text{if } Q_{ij} + \epsilon I_{ij} \neq 0, \\ 0 & \text{if } Q_{ij} + \epsilon I_{ij} = 0. \end{cases}$$

Let $G(N, E')$ be a chordal extension of the csp graph $G(N, E)$ for the quadratic function (16). Then the maximal cliques of the chordal extension $G(N, E')$ determine all possible fill-ins when the Cholesky factorization is applied to $\boldsymbol{Q} + \epsilon\boldsymbol{I}$ under the perfect elimination ordering induced from the chordal graph $G(N, E')$. More specifically, there is an $n \times n$ permutation matrix $\boldsymbol{P}$ corresponding to the perfect elimination ordering and an $n \times n$ lower triangular matrix $\boldsymbol{L}(\epsilon)$ such that

$$\begin{array}{rcl} \boldsymbol{P}\left(\boldsymbol{Q} + \epsilon\boldsymbol{I}\right)\boldsymbol{P}^T & = & \boldsymbol{L}(\epsilon)\boldsymbol{L}(\epsilon)^T \text{ (the Cholesky factorization)}, \\ \{i \in N : L(\epsilon)_{ij} \neq 0\} & \subset & C_j \text{ for some maximal clique } C_j \text{ of } G(N, E') \\ & & (j = 1, 2, \ldots, n). \end{array}$$

We assume without loss of generality that $\boldsymbol{P}$ is the identity matrix, *i.e.*, $1, 2, \ldots, n$ itself is the perfect elimination ordering under consideration.

Now, we show that there exist $d \in \mathbb{R}_+$, $\tilde{\boldsymbol{q}} \in \mathbb{R}^n$, an $n \times n$ lower triangular matrix $\widetilde{\boldsymbol{L}}$ and a $(1+n) \times (1+n)$ matrix $\widetilde{\boldsymbol{M}}$ such that

$$\left.\begin{array}{l} \widetilde{\boldsymbol{M}} = \begin{pmatrix} \sqrt{d} & \tilde{\boldsymbol{q}}^T \\ \boldsymbol{0} & \widetilde{\boldsymbol{L}} \end{pmatrix}, \; \boldsymbol{V} = \begin{pmatrix} \gamma - \zeta & \boldsymbol{q}^T \\ \boldsymbol{q} & \boldsymbol{Q} \end{pmatrix} = \widetilde{\boldsymbol{M}}\widetilde{\boldsymbol{M}}^T, \\ \{i \in N : \widetilde{L}_{ij} \neq 0\} \subset C_j \; (j = 1, 2, \ldots, n). \end{array}\right\} \tag{17}$$

For each $\epsilon \in (0,1]$, let

$$\boldsymbol{V}(\epsilon) \;=\; \begin{pmatrix} \gamma-\zeta & \boldsymbol{q}^T \\ \boldsymbol{q} & \boldsymbol{Q}+\epsilon\boldsymbol{I} \end{pmatrix} \text{ and } \boldsymbol{M}(\epsilon) = \begin{pmatrix} \sqrt{d(\epsilon)} & \boldsymbol{q}^T\boldsymbol{L}(\epsilon)^{-T} \\ \boldsymbol{0} & \boldsymbol{L}(\epsilon) \end{pmatrix},$$

where $d(\epsilon) = \gamma - \zeta - \boldsymbol{q}^T(\boldsymbol{Q}+\epsilon\boldsymbol{I})^{-1}\boldsymbol{q}$, which is nonnegative since the matrix $\boldsymbol{V}(\epsilon)$ is positive semidefinite for every $\epsilon \in (0,1]$. Then we observe the identity $\boldsymbol{V}(\epsilon) = \boldsymbol{M}(\epsilon)\boldsymbol{M}(\epsilon)^T$ for every $\epsilon \in (0,1]$. We also see that the norm of jth row vector of the matrix $\boldsymbol{M}(\epsilon)$ coincides with the square root of the ith diagonal element of the matrix $\boldsymbol{V}(\epsilon)$; hence it is bounded uniformly for all $\epsilon \in (0,1]$. Therefore we may take a sequence of $\epsilon \in (0,1]$ converging to zero as the matrix $\boldsymbol{M}(\epsilon)$ converges to some $\widetilde{\boldsymbol{M}}$ satisfying (17).

Now we obtain by (17) that

$$\begin{aligned} f_0(\boldsymbol{x}) &= (1,\boldsymbol{x}^T)\begin{pmatrix} \gamma-\zeta & \boldsymbol{q}^T \\ \boldsymbol{q} & \boldsymbol{Q} \end{pmatrix}\begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix} \\ &= (1,\boldsymbol{x}^T)\widetilde{\boldsymbol{M}}\widetilde{\boldsymbol{M}}^T\begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix} \\ &= \sum_{\ell=1}^{n}\left(\widetilde{\boldsymbol{M}}^T_{.\ell+1}\begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix}\right)^2 + \left(\sqrt{d}\right)^2. \end{aligned}$$

Here $\widetilde{\boldsymbol{M}}_{.\ell+1}$ denotes the $(\ell+1)$st column of $\widetilde{\boldsymbol{M}}$ $(\ell = 1,2,\ldots,n)$. It should be noted that each $\widetilde{\boldsymbol{M}}^T_{.\ell+1}\begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix}$ is an affine function whose support is contained in

$$\mathcal{A}_1^{C_\ell} = \left\{\boldsymbol{\alpha} \in \mathbb{Z}^n_+ : \alpha_i = 0 \ (i \notin C_\ell) \ \sum_{i\in C_\ell}\alpha_i \leq 1\right\}$$

as a polynomial. Therefore we have shown that the dense SOS relaxation (5) with $\omega = 1$ is equivalent to the sparse SOS relaxation (15) with $\omega = 1$, $p = n$ and C_ℓ $(\ell = 1,2,\ldots,p)$; if two maximal cliques C_k and C_ℓ with $k \neq \ell$ coincide, only one of them is necessary in the sparse SOS relaxation (15).

4 SOS relaxations of inequality constrained POPs

We discuss SOS relaxations of inequality constrained POPs using the correlative sparsity of the objective and constraint polynomials. The sets of supports that decide the SOS relaxations are determined using the csp matrices constructed from the correlative sparsity of inequality constrained POPs.

Let $f_k \in \mathbb{R}[\boldsymbol{x}]$ $(k = 0,1,2,\ldots,m)$. Consider the following POP:

$$\text{minimize} \quad f_0(\boldsymbol{x}) \quad \text{subject to} \quad f_k(\boldsymbol{x}) \geq 0 \ (k = 1,2,\ldots,m). \tag{18}$$

Let ζ^* denote the optimal value of this problem; $\zeta^* = \inf\{f_0(\boldsymbol{x}) : f_k(\boldsymbol{x}) \geq 0 \ (k = 1,2,\ldots,m)\}$. With the correlative sparsity of the POP (18), we determine the generalized Lagrangian function with the same sparsity and proper sets of supports in an SOS relaxation. A sparse SOS relaxation is proposed in two steps. In the first step, we convert the POP (18) into an unconstrained minimization of the generalized Lagrangian function according to the paper [17]. In the second step, we apply the sparse SOS relaxation given in the previous section for unconstrained POPs to the resulting minimization problem. A key point of utilizing the correlative sparsity of the POP (18) is that the POP (18) and its generalized Lagrangian function have the same correlative sparsity.

4.1 Correlative sparsity in inequality constrained POPs

We first investigate correlative sparsity in the POP (18). Let

$$F_k \;=\; \{i : \alpha_i \geq 1 \text{ for some } \boldsymbol{\alpha} \in \text{supp}(f_k)\} \;\; (k = 1, 2, \ldots, m).$$

Each F_k is regarded as the index set of variables x_i's which are involved in the polynomial f_k. For example, if $n = 4$ and $f_k(\boldsymbol{x}) = x_1^3 + 3x_1x_4 - 2x_4^2$, then $F_k = \{1, 4\}$. Define the $n \times n$ (symbolic, symmetric) csp matrix $\boldsymbol{R}$ such that

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

When the POP (18) has no inequality constraint or $m = 0$, the above definition of the csp matrix $\boldsymbol{R}$ coincides with the one given in the previous section for the objective polynomial f_0. When the csp matrix $\boldsymbol{R}$ is sparse, we call that the POP (18) is correlatively sparse.

4.2 Generalized Lagrangian duals

The generalized Lagrangian function [17] is defined as

$$L(\boldsymbol{x}, \boldsymbol{\varphi}) \;=\; f_0(\boldsymbol{x}) - \sum_{k=1}^{m} \varphi_k(\boldsymbol{x}) f_k(\boldsymbol{x}) \; (\forall \boldsymbol{x} \in \mathbb{R}^n \;\text{ and } \forall \boldsymbol{\varphi} \in \Phi),$$

where

$$\Phi = \left\{ \boldsymbol{\varphi} = (\varphi_1, \varphi_2, \ldots, \varphi_m) : \begin{array}{l} \varphi_k \in \mathbb{R}[\boldsymbol{x}, \mathcal{A}_\omega^N]^2 \;\text{ for some } \omega \in \mathbb{Z}_+ \\ (k = 1, 2, \ldots, m) \end{array} \right\}.$$

Then, for each fixed $\boldsymbol{\varphi} \in \Phi$, the problem of minimizing $L(\boldsymbol{x}, \boldsymbol{\varphi})$ over $\boldsymbol{x} \in \mathbb{R}^n$ serves as a Lagrangian relaxation problem such that its optimal objective value, which is denoted by $L^*(\boldsymbol{\varphi}) = \inf\{L(\boldsymbol{x}, \boldsymbol{\varphi}) : \boldsymbol{x} \in \mathbb{R}^n\}$, bounds the optimal objective value ζ^* of the POP (18) from below.

If our aim is to preserve the correlative sparsity of the POP (18) in the resulting SOS relaxation, we need to have the Lagrangian function L that inherits the correlative sparsity from the POP (18). Notice that $\boldsymbol{\varphi}$ can be chosen for this purpose. In [17], Kim, *et al.* proposed to choose a polynomial of the same variables as the variables x_i $(i \in F_k)$ in the polynomial f_k for each multiplier polynomial φ_k; $\text{supp}(\varphi_k) \subset \left\{\boldsymbol{\alpha} \in \mathbb{Z}_+^n : \alpha_i = 0 \; (i \notin F_k)\right\}$.

Let $\omega_k = \lceil \deg(f_k)/2 \rceil$ $(k = 0, 1, 2, \ldots, m)$ and $\omega_{\max} = \max\{\omega_k : k = 0, 1, \ldots, m\}$. For every nonnegative integer $\omega \geq \omega_{\max}$, define

$$\Phi_\omega = \left\{ \boldsymbol{\varphi} = (\varphi_1, \varphi_2, \ldots, \varphi_m) : \varphi_k \in \mathbb{R}[\boldsymbol{x}, \mathcal{A}^{F_k}_{\omega-\omega_k}]^2 \ (k = 1, 2, \ldots, m) \right\}.$$

Here the parameter $\omega \in \mathbb{Z}_+$ serves as the (relaxation) order of the SOS relaxation of the POP (18) that is derived in the next subsection. Then a generalized Lagrangian dual (with the Lagrangian multiplier $\boldsymbol{\varphi}$ restricted to Φ_ω) [17] is defined as

$$\text{maximize} \quad \zeta \quad \text{subject to} \quad L(\boldsymbol{x}, \boldsymbol{\varphi}) - \zeta \geq 0 \ (\forall \boldsymbol{x} \in \mathbb{R}^n) \text{ and } \boldsymbol{\varphi} \in \Phi_\omega. \tag{19}$$

Let L^*_ω denote the optimal value of this problem; $L^*_\omega = \sup\{L^*(\boldsymbol{\varphi}) : \boldsymbol{\varphi} \in \Phi_\omega\}$. Then $L^*_\omega \leq \zeta^*$. If the POP (18) includes the box inequality constraint of the form $\rho - x_i^2 \geq 0$ $(i = 1, 2, \ldots, n)$ for some $\rho > 0$, we know by Theorem 3.1 of [17] that L^*_ω converges to ζ^* as $\omega \to \infty$. When the feasible region of the POP (18) is bounded, we can add the box inequality constraint above without destroying the correlative sparsity of the problem.

4.3 Sparse SOS relaxations

We show how a sparse SOS relaxation is formulated using the sets of supports constructed from the csp matrix $\boldsymbol{R}$. Let $\omega \geq \omega_{\max}$ be fixed. Suppose that $\boldsymbol{\varphi} \in \Phi_\omega$. Then $L(\cdot, \boldsymbol{\varphi})$ forms a polynomial in x_i $(i = 1, 2, \ldots, n)$ with $\deg(L(\cdot, \boldsymbol{\varphi})) = 2\omega$. We also observe from the construction of the csp matrix $\boldsymbol{R}$ and Φ_ω that the polynomial $L(\cdot, \boldsymbol{\varphi})$ has the same csp matrix as the csp matrix $\boldsymbol{R}$ that we have constructed for the POP (18). As in Section 3.4, the csp matrix $\boldsymbol{R}$ induces the csp graph $G(N, E)$. By construction, we know that each F_k forms a clique of the csp graph $G(N, E)$. Let $C_1, C_2, \ldots, C_p$ be the maximal cliques of a chordal extension $G(N, E')$ of $G(N, E)$. Then, a sparse SOS relaxation of the POP (18) is written as

$$\text{maximize} \quad \zeta \quad \text{subject to} \quad L(\boldsymbol{x}, \boldsymbol{\varphi}) - \zeta \in \sum_{\ell=1}^{p} \mathbb{R}[\boldsymbol{x}, \mathcal{A}^{C_\ell}_\omega]^2 \ \text{ and } \boldsymbol{\varphi} \in \Phi_\omega. \tag{20}$$

We call the parameter $\omega \in \mathbb{Z}_+$ the *(relaxation) order*. Let ζ_ω denote the optimal objective value of this SOS optimization problem. Then $\zeta_\omega \leq L^*_\omega \leq \zeta^*$ for every $\omega \geq \omega_{\max}$, but the convergence of ζ_ω to ζ^* as $\omega \to \infty$ is not guaranteed in theory.

4.4 Primal approach

We have formulated a sparse SOS relaxation (20) of the inequality constrained POP (18) in the previous subsection. For numerical computation, we convert the SOS optimization problem (20) into an SDP, which serves as an SDP relaxation of the POP (18). We may regard this way of deriving an SDP relaxation from the POP (18) as the dual approach. We briefly mention below the so-called primal approach to the POP (18) whose sparsity is characterized by the csp matrix $\boldsymbol{R}$ and the csp graph $G(N, E)$. The SDP obtained from the primal approach plays an essential role in the Section 5.1 where we discuss how we compute an optimal solution of the POP (18). We use the same symbols and notation as in

Section 4.3. Let $\omega \geq \omega_{\max}$. To derive a primal SDP relaxation, we first transform the POP (18) into an equivalent polynomial SDP

$$\left.\begin{array}{ll} \text{minimize} & f_0(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{F_k}_{\omega-\omega_k})\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{F_k}_{\omega-\omega_k})^T f_k(\boldsymbol{x}) \in \mathcal{S}_+(\mathcal{A}^{F_k}_{\omega-\omega_k}) \ (k = 1, 2, \ldots, m), \\ & \boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{C_\ell}_{\omega})\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{C_\ell}_{\omega})^T \in \mathcal{S}_+(\mathcal{A}^{C_\ell}_{\omega}) \ (\ell = 1, 2, \ldots, p). \end{array}\right\} \tag{21}$$

The matrices $\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{F_k}_{\omega-\omega_k})\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{F_k}_{\omega-\omega_k})^T$ $(k = 1, 2, \ldots, m)$ and $\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{C_\ell}_{\omega})\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{C_\ell}_{\omega})^T$ $(\ell = 1, 2, \ldots, p)$ are positive semidefinite symmetric matrices of rank one for any $\boldsymbol{x} \in \mathbb{R}^n$, and has 1 as the element in its upper left corner. These facts ensure the equivalence between the POP (18) and the polynomial SDP above. Let

$$\begin{aligned} \widetilde{\mathcal{F}} &= \left(\bigcup_{\ell=1}^{p} \mathcal{A}^{C_\ell}_{\omega}\right) \backslash \{\boldsymbol{0}\}, \\ \widetilde{\mathcal{S}} &= \mathcal{S}(\mathcal{A}^{F_1}_{\omega-\omega_1}) \times \cdots \times \mathcal{S}(\mathcal{A}^{F_m}_{\omega-\omega_m}) \times \mathcal{S}(\mathcal{A}^{C_1}_{\omega}) \times \cdots \times \mathcal{S}(\mathcal{A}^{C_p}_{\omega}) \\ & \quad \text{(the set of block diagonal matrices of matrices in } \mathcal{S}(\mathcal{A}^{F_k}_{\omega-\omega_k}) \\ & \quad (k = 1, \ldots, m) \text{ and } \mathcal{S}(\mathcal{A}^{C_\ell}_{\omega}) \ (\ell = 1, \ldots, p) \text{ on their diagonal blocks)}, \\ \widetilde{\mathcal{S}}_+ &= \left\{ \boldsymbol{M} \in \widetilde{\mathcal{S}} : \text{positive semidefinite} \right\}. \end{aligned}$$

Then we can rewrite the polynomial SDP above as

$$\text{minimize} \quad \sum_{\boldsymbol{\alpha} \in \widetilde{\mathcal{F}}} \tilde{c}_0(\boldsymbol{\alpha}) \boldsymbol{x}^{\boldsymbol{\alpha}} \quad \text{subject to} \quad \boldsymbol{M}(\boldsymbol{0}) + \sum_{\boldsymbol{\alpha} \in \widetilde{\mathcal{F}}} \boldsymbol{M}(\boldsymbol{\alpha}) \boldsymbol{x}^{\boldsymbol{\alpha}} \in \widetilde{\mathcal{S}}_+.$$

for some $\tilde{c}_0(\boldsymbol{\alpha}) \in \mathbb{R}$ $(\boldsymbol{\alpha} \in \widetilde{\mathcal{F}})$, $\boldsymbol{M}(\boldsymbol{0}) \in \widetilde{\mathcal{S}}$ and $\boldsymbol{M}(\boldsymbol{\alpha}) \in \widetilde{\mathcal{S}}$ $(\boldsymbol{\alpha} \in \widetilde{\mathcal{F}})$. Now, replacing each monomial $\boldsymbol{x}^{\boldsymbol{\alpha}}$ by a single real variable $y_{\boldsymbol{\alpha}}$, we have an SDP relaxation problem of (18):

$$\text{minimize} \quad \sum_{\boldsymbol{\alpha} \in \widetilde{\mathcal{F}}} \tilde{c}_0(\boldsymbol{\alpha}) y_{\boldsymbol{\alpha}} \quad \text{subject to} \quad \boldsymbol{M}(\boldsymbol{0}) + \sum_{\boldsymbol{\alpha} \in \widetilde{\mathcal{F}}} \boldsymbol{M}(\boldsymbol{\alpha}) y_{\boldsymbol{\alpha}} \in \widetilde{\mathcal{S}}_+. \tag{22}$$

We denote the optimal objective value by $\hat{\zeta}_\omega$.

The SDP (22) derived above is the dual of the SDP from the SOS optimization problem (20). We call the SDP (22) primal and the SDP induced from (20) dual. See the paper [17] for more technical details. If we use the primal-dual interior point method, we can solve both SDPs simultaneously.

4.5 SOS and SDP relaxations of quadratic optimization problems with order 1

Consider a quadratic optimization problem (QOP)

$$\left.\begin{array}{ll} \text{minimize} & \boldsymbol{x}^T \boldsymbol{Q}_0 \boldsymbol{x} + 2\boldsymbol{q}_0^T \boldsymbol{x} \\ \text{subject to} & \boldsymbol{x}^T \boldsymbol{Q}_k \boldsymbol{x} + 2\boldsymbol{q}_k^T \boldsymbol{x} + \gamma_k \geq 0 \ (k = 1, 2, \ldots, m). \end{array}\right\} \tag{23}$$

Here $\boldsymbol{Q}_k$ denotes an $n \times n$ symmetric matrix $(k = 0, 1, 2, \ldots, m)$, $\boldsymbol{q}_k \in \mathbb{R}^n$ $(k = 0, 1, 2, \ldots, m)$ and $\gamma_k \in \mathbb{R}$ $(k = 1, 2, \ldots, m)$. Based on the discussions in Section 3.5, we show that the sparse SOS and SDP relaxations with order 1 for the QOP (23) is as effective as the dense SOS and SDP relaxations for the QOP (23). If we let

$$\gamma_0 = 0 \text{ and } \widetilde{\boldsymbol{Q}}_k = \begin{pmatrix} \gamma_k & \boldsymbol{q}_k^T \\ \boldsymbol{q}_k & \boldsymbol{Q}_k \end{pmatrix} \ (k = 0, 1, 2, \ldots, m),$$

the QOP is rewritten as

$$\begin{array}{ll} \text{minimize} & \widetilde{\boldsymbol{Q}}_0 \bullet \left(\begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix} (1, \ \boldsymbol{x}^T) \right) \\ \text{subject to} & \widetilde{\boldsymbol{Q}}_k \bullet \left(\begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix} (1, \ \boldsymbol{x}^T) \right) \geq 0 \ (k = 1, 2, \ldots, m). \end{array}$$

If we replace the matrix $\begin{pmatrix} 1 \\ \boldsymbol{x} \end{pmatrix} (1, \ \boldsymbol{x}^T)$, which is quadratic in the vector variable $\boldsymbol{x} \in \mathbb{R}^n$, by a positive semidefinite matrix variable $\boldsymbol{X} \in \mathcal{S}_+^{1+n}$ with $X_{00} = 1$, we obtain an SDP relaxation of the QOP (23)

$$\text{minimize} \quad \widetilde{\boldsymbol{Q}}_0 \bullet \boldsymbol{X} \quad \text{subject to} \quad \widetilde{\boldsymbol{Q}}_k \bullet \boldsymbol{X} \geq 0 \ (k = 1, 2, \ldots, m), \ \boldsymbol{X} \in \mathcal{S}_+^{1+n}, \ X_{00} = 1.$$

Here $\mathcal{S}_+^{1+n}$ denotes the set of $(1+n) \times (1+n)$ positive semidefinite symmetric matrices. This type of SDP relaxations is rather classical and has been studied in many literatures [7, 9, etc.]. This is also a special case of the application of the primal SDP relaxation (22) described in Section 5.1 to the QOP (23) where $p = 1$, $C_1 = N$ and $\omega_0 = 1$.

We can formulate this relaxation using sums of squares of polynomials from the dual side as well. First, consider the Lagrangian dual problem of the QOP (23).

$$\text{maximize} \quad \zeta \quad \text{subject to} \quad L(\boldsymbol{x}, \boldsymbol{\varphi}) - \zeta \geq 0 \ (\forall \boldsymbol{x} \in \mathbb{R}^n) \ \text{ and } \boldsymbol{\varphi} \in \mathbb{R}_+^m, \tag{24}$$

where L denotes the Lagrangian function such that

$$L(\boldsymbol{x}, \boldsymbol{\varphi}) = \boldsymbol{x}^T \left(\boldsymbol{Q}_0 - \sum_{k=1}^m \varphi_k \boldsymbol{Q}_k \right) \boldsymbol{x} + 2 \left(\boldsymbol{q}_0 - \sum_{k=1}^m \varphi_k \boldsymbol{q}_k \right)^T \boldsymbol{x} - \sum_{k=1}^m \varphi_k \gamma_k.$$

Then we replace the constraint $L(\boldsymbol{x}, \boldsymbol{\varphi}) - \zeta \geq 0$ $(\forall \boldsymbol{x} \in \mathbb{R}^n)$ by a sum of squares condition $L(\boldsymbol{x}, \boldsymbol{\varphi}) - \zeta \in \mathbb{R}[\boldsymbol{x}, \mathcal{A}_1^N]^2$ to obtain an SOS relaxation.

$$\text{maximize} \quad \zeta \quad \text{subject to} \quad L(\boldsymbol{x}, \boldsymbol{\varphi}) - \zeta \in \mathbb{R}[\boldsymbol{x}, \mathcal{A}_1^N]^2 \ \text{ and } \boldsymbol{\varphi} \in \mathbb{R}_+^m. \tag{25}$$

Now consider the aggregated sparsity pattern matrix $\tilde{\boldsymbol{R}}$ over the coefficient matrices $\boldsymbol{Q}_0, \boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_m$ such that

$$\tilde{R}_{ij} = \begin{cases} \star & \text{if } i = j, \\ \star & \text{if } i \neq j \text{ and } [Q_k]_{ij} \neq 0 \ \text{ for some } k \in \{0, 1, 2, \ldots, m\}, \\ 0 & \text{otherwise} , \end{cases}$$

which coincides with the csp matrix of the Lagrangian function $L(\cdot, \boldsymbol{\varphi})$ with $\boldsymbol{\varphi} \in \mathbb{R}_+^m$. It should be noted that $\tilde{\boldsymbol{R}}$ is different from the csp matrix $\boldsymbol{R}$ of the QOP (23); we use

$\tilde{\boldsymbol{R}}$ instead of $\boldsymbol{R}$ since we are interested only in sparse SOS and SDP relaxations of order $\omega = 1$. Let $G(N, E')$ be a chordal extension of the csp graph $G(N, E)$ from $\tilde{\boldsymbol{R}}$, and C_ℓ ($\ell = 1, 2, \ldots, p$) the maximal cliques of $G(N, E')$. Then we can apply the sparse relaxation (15) to the unconstrained minimization of the Lagrangian function $L(\cdot, \boldsymbol{\varphi})$ with $\boldsymbol{\varphi} \in \mathbb{R}^m_+$. Thus, replacing $\mathbb{R}[\boldsymbol{x}, \mathcal{A}_1^N]^2$ in the dense SOS relaxation (25) by $\sum_{\ell=1}^{p} \mathbb{R}[\boldsymbol{x}, \mathcal{A}_1^{C_\ell}]^2$, we obtain a sparse SOS relaxation:

$$\left.\begin{array}{ll} \text{maximize} & \zeta \\ \text{subject to} & L(\boldsymbol{x}, \boldsymbol{\varphi}) - \zeta \in \displaystyle\sum_{\ell=1}^{p} \mathbb{R}[\boldsymbol{x}, \mathcal{A}_1^{C_\ell}]^2 \ \text{ and } \boldsymbol{\varphi} \in \mathbb{R}^m_+. \end{array}\right\} \tag{26}$$

Note that the Lagrangian function $L(\cdot, \boldsymbol{\varphi})$ is a quadratic function in $\boldsymbol{x} \in \mathbb{R}^n$ which results in the same csp graph $G(N, E)$ for each $\boldsymbol{\varphi} \in \mathbb{R}^m_+$. In view of the discussions given in Section 3.5, the sparse relaxation (26) is equivalent to the dense relaxation (25).

5 Some technical issues

5.1 Computing optimal solutions

We present a technique to compute an optimal solution of the POP (18) that is based on the following lemma.

Lemma 5.1. *Assume that*

(a) the POP (18) has an optimal solution,

(b) the SDP (22) with the parameter $\omega \geq \omega_{\max}$ has an optimal solution $(\hat{y}_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \widetilde{\mathcal{F}})$,

(c) if $(\hat{y}^1_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \widetilde{\mathcal{F}})$ and $(\hat{y}^2_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \widetilde{\mathcal{F}})$ are optimal solutions of the SDP (22) then $\hat{y}^1_{\boldsymbol{e}^i} = \hat{y}^2_{\boldsymbol{e}^i}$ $(i = 1, 2, \ldots, n)$.

Define

$$\hat{\boldsymbol{x}} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_n), \ \hat{x}_i = \hat{y}_{\boldsymbol{e}^i} \ (i = 1, 2, \ldots, n). \tag{27}$$

Then the following two assertions are equivalent.

(d) $\hat{\zeta}_\omega = \zeta^*$,

(e) $\hat{\boldsymbol{x}}$ is a feasible solution of the POP (18) and $f_0(\hat{\boldsymbol{x}}) = \hat{\zeta}_\omega$; hence $\hat{\boldsymbol{x}}$ is an optimal solution of the POP (18).

Proof: Since $\hat{\zeta}_\omega \leq \zeta^* \leq f_0(\boldsymbol{x})$ for any feasible solution $\boldsymbol{x}$ of the POP (18), (e) $\Rightarrow$ (d) follows. Now suppose that (d) holds. Let $(\bar{y}_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \widetilde{\mathcal{F}})$ be such that

$$\bar{y}_{\boldsymbol{\alpha}} = \bar{\boldsymbol{x}}^{\boldsymbol{\alpha}} \ (\boldsymbol{\alpha} \in \widetilde{\mathcal{F}}), \tag{28}$$

where $\bar{\boldsymbol{x}}$ denotes an optimal solution of the POP (18) whose existence is ensured by (a). Then $(\bar{y}_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \widetilde{\mathcal{F}})$ is a feasible solution of the SDP (22) having the same objective value

as $\zeta^* = f_0(\bar{\boldsymbol{x}})$. Since $\hat{\zeta}_\omega = \zeta^*$ is the optimal value of the SDP (22) by (d), $(\bar{y}_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \widetilde{\mathcal{F}})$ must be an optimal solution of the SDP (22). By the assumptions (b) and (c), we see that $\hat{y}_{\boldsymbol{e}^i} = \bar{y}_{\boldsymbol{e}^i}$ $(i = 1, 2, \ldots, n)$, which together with (27) and (28) imply that $\hat{x}_i = \hat{y}_{\boldsymbol{e}^i} = \bar{y}_{\boldsymbol{e}^i} = \bar{x}_i$ $(i = 1, 2, \ldots, n)$. Hence $\hat{\boldsymbol{x}}$ is an optimal solution of the POP (18) and (e) follows.
∎

If (a), (b) and (c) are satisfied, Lemma 5.1 shows how we compute an optimal solution of the POP (18). That is, if (e) holds for $\hat{\boldsymbol{x}}$ given by (27), then $\hat{\boldsymbol{x}}$ is an optimal solution of the POP (18). Otherwise, we have $\hat{\zeta}_\omega < \zeta^*$ although ζ^* is unknown. In the latter case, we replace ω by $\omega + 1$ and solve the SDP (22) with a lager relaxation order ω to compute a tighter lower bound for the objective value of the POP (18) as well as an optimal solution of the POP (18). Notice that (e) provides a certificate that the lower bound $\hat{\zeta}_\omega$ for the objective values of the POP (18) attains the exact optimal value ζ^* of the POP (18).

It is fair to assume the uniqueness of optimal solutions of the POP (18) mathematically, which implies (c) of Lemma 5.1. When the feasible region of the POP (18) is bounded, perturbing the objective function slightly with small random numbers makes its solution unique as we see below. In practice, however, the SDP (22) may have multiple optimal solutions, even if the POP (18) has a unique optimal solution.

We may assume without loss of generality that the objective polynomial function f_0 of the POP (18) is linear. If f is not linear, we may replace $f_0(\boldsymbol{x})$ by a new variable x_0 and add the inequality constraint $f_0(\boldsymbol{x}) \leq x_0$. Then, for any $\boldsymbol{p} \in \mathbb{R}^n$, the problem of minimizing the perturbed objective function $f_0(\boldsymbol{x}) + \boldsymbol{p}^T\boldsymbol{x}$ subject to the inequality constraints of the POP (18) has a unique optimal solution if and only if the perturbed problem with a linear objective function

$$\text{minimize } x_0 + \boldsymbol{p}^T\boldsymbol{x} \text{ subject to } x_0 - f_0(\boldsymbol{x}) \geq 0,\ f_k(\boldsymbol{x}) \geq 0\ (k = 1, 2, \ldots, m)$$

has a unique solution. Define

$$D \;=\; \left\{ (y_{\boldsymbol{e}^1}, y_{\boldsymbol{e}^2}, \ldots, y_{\boldsymbol{e}^n}) \in \mathbb{R}^n : (y_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \widetilde{\mathcal{F}}) \text{ is a feasible solution of (22)} \right\}.$$

Note that D is a convex subset of $\mathbb{R}^n$ since it is a projection of the feasible region of (22) which is convex. By construction, the SDP (22) is equivalent to the convex program

$$\text{minimize } \; f_0(\boldsymbol{x}) \text{ subject to } \boldsymbol{x} \in D.$$

We also assume that D is compact. Let $\epsilon > 0$. Then, for almost every $\boldsymbol{p} \in \{\boldsymbol{r} \in \mathbb{R}^n : |r_j| < \epsilon\ (j = 1, 2, \ldots, n)\}$, the perturbed convex minimization

$$\text{minimize} \quad f_0(\boldsymbol{x}) + \boldsymbol{p}^T\boldsymbol{x} \quad \text{subject to } \boldsymbol{x} \in D$$

has a unique minimizer. See the paper [6] for a proof of this assertion. Therefore, if we replace the objective function $f_0(\boldsymbol{x})$ by $f_0(\boldsymbol{x}) + \boldsymbol{p}^T\boldsymbol{x}$ in the POP (18), the corresponding SDP relaxation (22) satisfies the assumption (a) of Lemma 5.1.

The assumption that D is compact is not considered to be much restrictive when the feasible region of the POP (18) is bounded. For example, if the feasible region of the POP (18) lies in a unit box $\{\boldsymbol{x} \in \mathbb{R}^n : 0 \leq x_i \leq 1\ (i = 1, 2, \ldots, n)\}$, we can put additional inequality constraints $0 \leq y_{\boldsymbol{e}^i} \leq 1$ $(i = 1, 2, \ldots, n)$, which ensure the boundedness of D. In this case, we can further add $0 \leq y_{\boldsymbol{\alpha}} \leq 1$ ($\boldsymbol{\alpha} \in \widetilde{\mathcal{F}}$) so that the resulting feasible region of the SDP (22) with these inequalities is bounded and that D is guaranteed to be compact.

In the numerical experiments in Section 6, we add a perturbation $\boldsymbol{p}^T\boldsymbol{x}$, where $\boldsymbol{p} = (p_1, p_2, \ldots, p_n)^T \in \mathbb{R}^n$ and each p_i denotes a randomly generated and sufficiently small number, to the objective function $f_0(\boldsymbol{x})$ of the unconstrained POP (2) and the constrained POP (18), and apply the SOS and SDP relaxations described in Sections 3 and 4 to the perturbed unconstrained and constrained POPs, respectively.

5.2 Equality constraints

In this subsection, we deal with a POP (18) with additional equality constraints. Consider the POP

$$\left.\begin{array}{ll} \text{minimize} & f_0(\boldsymbol{x}) \\ \text{subject to} & f_k(\boldsymbol{x}) \geq 0 \ (k = 1, 2, \ldots, m), \ \ h_j(\boldsymbol{x}) = 0 \ (j = 1, 2, \ldots, q). \end{array}\right\} \tag{29}$$

Here $f_k \in \mathbb{R}[\boldsymbol{x}]$ $(k = 0, 1, 2, \ldots, m)$ and $h_j \in \mathbb{R}[\boldsymbol{x}]$ $(j = 1, 2, \ldots, q)$. We can replace each equality constraints $h_j(\boldsymbol{x}) = 0$ by two inequality constraints $h_j(\boldsymbol{x}) \geq 0$ and $-h_j(\boldsymbol{x}) \geq 0$. Hence we reduce the POP (29) to the inequality constrained POP of the form

$$\left.\begin{array}{ll} \text{minimize} & f_0(\boldsymbol{x}) \\ \text{subject to} & f_k(\boldsymbol{x}) \geq 0 \ (k = 1, 2, \ldots, m), \\ & h_j(\boldsymbol{x}) \geq 0, \ -h_j(\boldsymbol{x}) \geq 0 \ (j = 1, 2, \ldots, q). \end{array}\right\} \tag{30}$$

Let

$$\begin{array}{rcl} \omega_k & = & \lceil \deg(f_k)/2 \rceil \ (k = 0, 1, 2, \ldots, m), \\ \chi_j & = & \lceil \deg(h_j)/2 \rceil \ (j = 1, 2, \ldots, q), \\ \omega_{\max} & = & \max\{\omega_k \ (k = 0, 1, 2, \ldots, m), \ \chi_j \ (j = 1, 2, \ldots, q)\}, \\ F_k & = & \{i : \alpha_i \geq 1 \text{ for some } \boldsymbol{\alpha} \in \operatorname{supp}(f_k)\} \ (k = 1, 2, \ldots, m), \\ H_j & = & \{i : \alpha_i \geq 1 \text{ for some } \boldsymbol{\alpha} \in \operatorname{supp}(h_j)\} \ (j = 1, 2, \ldots, q). \end{array}$$

We construct the csp matrix $\boldsymbol{R}$ and the csp graph $G(N, E)$ of the POP (30). Let $C_1, C_2, \ldots, C_p$ be the maximal cliques of a chordal extension of $G(N, E)$, and $\omega \geq \omega_{\max}$. Applying the SOS relaxation given for the inequality constrained POP (18) in Section 4 to the POP (30), we have the SOS optimization problem

$$\begin{array}{ll} \text{maximize} & \zeta \\ \text{subject to} & f_0(\boldsymbol{x}) - \displaystyle\sum_{k=1}^{m} \varphi_k(\boldsymbol{x}) f_k(\boldsymbol{x}) - \sum_{j=1}^{q} \left(\psi_j^+(\boldsymbol{x}) - \psi_j^-(\boldsymbol{x})\right) h_j(\boldsymbol{x}) - \zeta \ \in \sum_{\ell=1}^{p} \mathbb{R}[\boldsymbol{x}, \mathcal{A}_\omega^{C_\ell}]^2, \\ & \boldsymbol{\varphi} \in \Phi_\omega, \ \psi_j^+, \psi_j^- \in \mathbb{R}[\boldsymbol{x}, \mathcal{A}_{\omega-\chi_j}^{H_j}]^2 \ (j = 1, 2, \ldots, q). \end{array}$$

Since $\mathbb{R}[\boldsymbol{x}, \mathcal{A}_{\omega-\chi_j}^{H_j}]^2 - \mathbb{R}[\boldsymbol{x}, \mathcal{A}_{\omega-\chi_j}^{H_j}]^2 = \mathbb{R}[\boldsymbol{x}, \mathcal{A}_{2(\omega-\chi_j)}^{H_j}]$, this problem is equivalent to the SOS optimization problem

$$\left.\begin{array}{ll} \text{maximize} & \zeta \\ \text{subject to} & f_0(\boldsymbol{x}) - \displaystyle\sum_{k=1}^{m} \varphi_k(\boldsymbol{x}) f_k(\boldsymbol{x}) - \sum_{j=1}^{q} \psi_j(\boldsymbol{x}) h_j(\boldsymbol{x}) - \zeta \in \sum_{\ell=1}^{p} \mathbb{R}[\boldsymbol{x}, \mathcal{A}_\omega^{C_\ell}]^2, \\ & \boldsymbol{\varphi} \in \Phi_\omega, \ \psi_j \in \mathbb{R}[\boldsymbol{x}, \mathcal{A}_{2(\omega-\chi_j)}^{H_j}] \ (j = 1, 2, \ldots, q). \end{array}\right\} \tag{31}$$

We can solve the SOS optimization problem (31) as an SDP with free variables.

When we apply the primal approach to the POP (30), the polynomial SDP (21) is replaced by

$$\left.\begin{array}{ll} \text{minimize} & f_0(\boldsymbol{x}) \\ \text{subject to} & \boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{F_k}_{\omega-\omega_k})\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{F_k}_{\omega-\omega_k})^T f_k(\boldsymbol{x}) \in \mathcal{S}_+(\mathcal{A}^{F_k}_{\omega-\omega_k}) \ (k = 1, 2, \ldots, m), \\ & \boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{H_j}_{\omega-\chi_j})\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{H_j}_{\omega-\chi_j})^T h_j(\boldsymbol{x}) = \boldsymbol{O} \in \mathcal{S}(\mathcal{A}^{H_j}_{\omega-\chi_j}) \ (j = 1, 2, \ldots, q), \\ & \boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{C_\ell}_{\omega})\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{C_\ell}_{\omega})^T \in \mathcal{S}_+(\mathcal{A}^{C_\ell}_{\omega}) \ (\ell = 1, 2, \ldots, p). \end{array}\right\} \quad (32)$$

Since some elements of the symmetric matrix $\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{H_j}_{\omega-\chi_j})\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{H_j}_{\omega-\chi_j})^T$ coincide, the system of nonlinear equations $\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{H_j}_{\omega-\chi_j})\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{H_j}_{\omega-\chi_j})^T h_j(\boldsymbol{x}) = \boldsymbol{O}$ contains multiple identical equations ($j = 1, 2, \ldots, q$), and hence so does its linearization. To avoid the degeneracy caused by multiplying of identical equations, we replace it by an equivalent system of nonlinear equations

$$\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{H_j}_{2(\omega-\chi_j)})h_j(\boldsymbol{x}) = \boldsymbol{0} \ (j = 1, 2, \ldots, q). \quad (33)$$

Linearizing the resulting polynomial SDP as in Section 5.1 provides a primal SDP relaxation of the POP (29). This SDP relaxation problem and the SDP relaxation problem induced from the SOS optimization problem (31) have the primal-dual relationship.

Since it is not necessary to multiply a positive semidefinite polynomial matrix $\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{H_j}_{\omega-\chi_j})\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{H_j}_{\omega-\chi_j})^T$ to the equality constraint $h_j(\boldsymbol{x}) = 0$ as we have observed, we can further modify the primal approach mentioned above. We replace the system of nonlinear equations (33) by

$$\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{H_j}_{2\omega-\kappa_j})h_j(\boldsymbol{x}) = \boldsymbol{0} \ (j = 1, 2, \ldots, q), \quad (34)$$

where $\kappa_j = \deg(h_j)$. By the definition of χ_j, we know that $2(\omega-\chi_j) = 2\omega-\kappa_j-1$ if $\deg(h_j)$ is odd and $2(\omega - \chi_j) = 2\omega - \kappa_j$ if $\deg(h_j)$ is even. Hence, in the former case, the system of nonlinear equations (34) is a stronger constraint than the system of nonlinear equations (33), and the degree of the component polynomials in (33) is bounded by $2\omega - 1$. Note that the maximum degree of the component polynomials in (34) is 2ω, the same degree as $\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{C_\ell}_{\omega})\boldsymbol{u}(\boldsymbol{x}, \mathcal{A}^{C_\ell}_{\omega})^T$ ($\ell = 1, 2, \ldots, p$) of (32), in both odd and even cases. Thus this modification is valid. Even when the original system of polynomial equations $h_j(\boldsymbol{x}) = 0$ ($j = 1, 2, \ldots, q$) is linearly independent, the resulting system (34) can be linearly dependent; hence so is its linearization. Here we say that a system of polynomial equations $g_j(\boldsymbol{x}) = 0$ ($j = 1, 2, \ldots, r$) is linearly dependent if there exists a nonzero $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_r) \in \mathbb{R}^r$ such that $\sum_{j=1}^{r} \lambda_j g_j(\boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathbb{R}^n$, and linearly independent otherwise. In such a linearly dependent case, we eliminate some redundant equations from the system (34) or from its linearization.

In the dual approach to the POP (29) having equality constraints, we can replace the condition $\psi_j \in \mathbb{R}[\boldsymbol{x}, \mathcal{A}^{H_j}_{2(\omega-\chi_j)}]$ ($j = 1, 2, \ldots, q$) by $\psi_j \in \mathbb{R}[\boldsymbol{x}, \mathcal{A}^{H_j}_{2\omega-\kappa_j}]$ ($j = 1, 2, \ldots, q$) in its SOS relaxation (31).

5.3 Reducing sizes of SOS relaxations

The method proposed in the paper [20] can be used to reduce the size of the dense SOS relaxation. It consists of two phases. Let $\mathcal{F}$ be the support of a polynomial f. We represent f as a sum of squares of unknown polynomials $\phi_i \in \mathbb{R}[\boldsymbol{x}, \mathcal{G}]$ ($i = 1, 2, \ldots, k$) with some support $\mathcal{G}$ such that $f = \sum_{i=1}^{k} \phi^2$. For numerical efficiency, we want to choose a smaller $\mathcal{G}$. In phase 1, we compute

$$\mathcal{G}^0 = \left(\text{the convex hull of } \left\{\frac{\boldsymbol{\alpha}}{2} : \boldsymbol{\alpha} \in \mathcal{F}^e\right\}\right) \bigcap \mathbb{Z}^n_+,$$

where $\mathcal{F}^e = \{\boldsymbol{\alpha} \in \mathcal{F} : \alpha_i \text{ is even } (i = 1, 2, \ldots, n)\}$. It is known that $\mathrm{supp}(\phi_i) \subset \mathbb{R}[\boldsymbol{x}, \mathcal{G}^0]$ $(i = 1, 2, \ldots, k)$ for any sum of squares representation of $f = \sum_{i=1}^{k} \phi_i^2$. In phase 2, we eliminate redundant elements from $\mathcal{G}^0$ that are unnecessary in any sum of squares representation of f.

In the sparse SOS relaxations (15) and (20), we can apply phase 2 of the method with some modification to eliminate redundant elements from $\mathcal{A}^{C_\ell}_{\omega}$ $(\ell = 1, 2, \ldots, p)$. Let $\mathcal{F}$ denote the support of a polynomial f which we want to represent as

$$f = \sum_{\ell=1}^{p} \psi_\ell \text{ for some } \psi_\ell \in \mathbb{R}[\boldsymbol{x}, \mathcal{G}_\ell]^2 \ (\ell = 1, 2, \ldots, p). \tag{35}$$

The polynomial f corresponds to $f_0 - \zeta$ in the sparse SOS relaxation of the problem (3), or equivalently, to the unconstrained POP (2), and it also corresponds to $L(\cdot, \boldsymbol{\varphi}) - \zeta$ with $\boldsymbol{\varphi} \in \Phi_\omega$ in the problem (19) that is equivalent to the constrained POP (18). In both cases, we assume that the family of supports $\mathcal{G}_\ell = \mathcal{A}^{C_\ell}_{\omega}$ $(\ell = 1, 2, \ldots, p)$ is sufficient to represent f as in (35); hence phase 1 is not implemented. Let $\mathcal{F}^e = \{\boldsymbol{\alpha} \in \mathcal{F} : \alpha_i \text{ is even } (i = 1, 2, \ldots, n)\}$. For each $\boldsymbol{\alpha} \in \bigcup_{\ell=1}^{p} \mathcal{G}_\ell$, we check whether the following relations are true.

$$2\boldsymbol{\alpha} \notin \mathcal{F}^e \ \text{ and } 2\boldsymbol{\alpha} \notin \bigcup_{\ell=1}^{p} \{\boldsymbol{\beta} + \boldsymbol{\gamma} : \boldsymbol{\beta} \in \mathcal{G}_\ell,\ \boldsymbol{\gamma} \in \mathcal{G}_\ell,\ \boldsymbol{\beta} \neq \boldsymbol{\alpha}\}$$

If an $\boldsymbol{\alpha} \in \mathcal{G}_\ell$ satisfies these two relations, we can eliminate $\boldsymbol{\alpha}$ from $\mathcal{G}_\ell$ and continue this process until no $\boldsymbol{\alpha} \in \bigcup_{\ell=1}^{p} \mathcal{G}_\ell$ satisfies these two relations. See the paper [20] for more details.

5.4 Supports for Lagrange multiplier polynomials φ_k $(k = 1, 2, \ldots, m)$

When the generalized Lagrangian dual (19) and the sparse SOS relaxation (20) are described in Section 4, each multiplier polynomial φ_k is chosen from SOS polynomials with the support $\mathcal{A}^{F_k}_{\omega-\omega_k}$ to inherit the correlative sparsity from the original constrained POP (18). We show a way to take a larger support to strengthen the SOS relaxation (20) while maintaining the same correlative sparsity. For each k, let $J_k = \{\ell : F_k \subset C_\ell\}$ $(k = 1, 2, \ldots, m)$, where $C_1, C_2, \ldots, C_p$ denote the maximal cliques of a chordal extension $G(N, E')$ of the csp graph $G(N, E)$ induced from the POP (18). We know by construction that $F_k \subset C_\ell$ for some $\ell = 1, 2, \ldots, p$. Hence $J_k \neq \emptyset$. Then we may replace the support $\mathcal{A}^{F_k}_{\omega-\omega_k}$ of SOS polynomials for φ_k by $\mathcal{A}^{C_\ell}_{\omega-\omega_k}$ for some $\ell \in J_k$ in the sparse SOS relaxation (20). By this modification, we strengthen the SOS relaxation (20) while keeping the same sparsity as the chordal extension $G(N, E')$ of the csp graph $G(N, E)$; the csp graph of the modified Lagrangian function may change but $G(N, E')$ remains to be a chordal extension of the csp graph induced from the modified Lagrangian function. We may replace $\mathcal{A}^{F_k}_{\omega-\omega_k}$ by $\mathcal{A}^{C}_{\omega-\omega_k}$ for some union C of sets C_ℓ $(\ell \in J_k)$. This modification may destroy the correlative sparsity of $G(N, E')$ but the resulting SOS relaxation is still spare whenever C is a small subset of N.

5.5 Polynomial valid inequalities and their linearization

By adding appropriate polynomial valid inequalities to the constrained POP (18), we can strengthen its SDP relaxation (22). This idea has been used in many convex relaxation methods. See the paper [19] and the references therein. We consider two types of polynomial valid inequalities that occur frequently in practice. These inequalities are used for some test problems in the numerical experiments in Section 6. Suppose that (18) involves the nonnegative and upper bound constraints on all variables: $0 \leq x_i \leq \rho_i$ $(i = 1, 2, \ldots, n)$, where ρ_i denotes a nonnegative number $(i = 1, 2, \ldots, n)$. In this case, $0 \leq \boldsymbol{x}^{\boldsymbol{\alpha}} \leq \boldsymbol{\rho}^{\boldsymbol{\alpha}}$ $(\boldsymbol{\alpha} \in \widetilde{\mathcal{F}})$ forms valid inequalities, where $\boldsymbol{\rho} = (\rho_1, \rho_2, \ldots, \rho_n) \in \mathbb{R}^n$. Therefore we can add their linearizations $0 \leq y_{\boldsymbol{\alpha}} \leq \boldsymbol{\rho}^{\boldsymbol{\alpha}}$ to the primal SDP relaxation (22). The complementarity condition $x_i x_j = 0$ is another example. If $\alpha_i \geq 1$ and $\alpha_j \geq 1$ in this case for some $\boldsymbol{\alpha} \in \mathbb{Z}^n_+$, then $\boldsymbol{x}^{\boldsymbol{\alpha}} = 0$ forms a valid equality; hence we can add $y_{\boldsymbol{\alpha}} = 0$ to the primal SDP relaxation or we can reduce the size of the primal SDP relaxation by eliminating the variable $y_{\boldsymbol{\alpha}} = 0$.

5.6 Scaling

High degree of polynomials in POPs can cause numerical problems. Introducing appropriate scaling techniques may resolve numerical difficulty. We explain how a proper scaling of objective and constrained polynomials helps achieve numerically stability in the SOS and SDP relaxations. Notice that the polynomial SDP (21) is equivalent to the unconstrained POP (18) and induces the primal SDP relaxation (22). Even when the degrees of objective and constrained polynomials are small, (21) involves high degree monomials as the order ω gets larger; for example, monomial $\boldsymbol{x}^{\boldsymbol{\alpha}}$ of degree 8 appears in (22) if $\omega = 4$. Note that each variable $y_{\boldsymbol{\alpha}}$ corresponds to a monomial $\boldsymbol{x}^{\boldsymbol{\alpha}}$. More precisely, if $\boldsymbol{x}$ is a feasible solution of the POP (18), then $(y_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \widetilde{\mathcal{F}})$ is a feasible solution of the primal SDP relaxation (22) with the same objective value as (18). Therefore, if the magnitudes of some nonzero components of a feasible (or optimal) solution $\boldsymbol{x}$ of (18) are much larger (or smaller) than 1, the magnitude of some components of the corresponding solution $(y_{\boldsymbol{\alpha}} : \boldsymbol{\alpha} \in \widetilde{\mathcal{F}})$ can be huge (or tiny). This may be the source of numerical difficulties; for example, if $n = 3$, $x_1 = 1000$, $x_2 = 1000$, $x_3 = 0.1$, $\boldsymbol{\alpha} = (2, 2, 0)$ and $\boldsymbol{\beta} = (0, 0, 4)$ then $y_{\boldsymbol{\alpha}} = 10^{12}$ and $y_{\boldsymbol{\beta}} = 10^{-4}$. To avoid such unbalanced magnitudes in the components of feasible (or optimal) solutions of the primal SDP relaxation (22), it would be ideal to scale the POP (18) so that the magnitudes of all nonzero components of optimal solutions of the scaled problem are near 1. Practically such an ideal scaling is impossible unless we know optimal solutions in advance.

Here we restrict our discussion to a POP of the form (18) with additional finite lower and upper bound constraints on variables x_i $(i = 1, 2, \ldots, n)$:

$$\eta_i \leq x_i \leq \rho_i \ (i = 1, 2, \ldots), \tag{36}$$

where η_i and ρ_i denote real numbers such that $\eta_i < \rho_i$ $(i = 1, 2, \ldots, n)$. In this case, we can perform a linear transformation to the variables x_i $(i = 1, 2, \ldots, n)$ such that

$$z_i = (x_i - \eta_i)/(\rho_i - \eta_i) \ (i = 1, 2, \ldots, n).$$

Then we have objective and constrained polynomials $g_k \in \mathbb{R}[\boldsymbol{z}]$ $(k = 0, 1, 2, \ldots, m)$ such that

$$\begin{aligned} & g_k(z_1, z_2, \dots, z_n) \\ = \; & f_k((\rho_1 - \eta_1)z_1 + \eta_1, (\rho_2 - \eta_2)z_2 + \eta_2, \dots, (\rho_n - \eta_n)z_n + \eta_n) \\ & \text{for every } \boldsymbol{z} = (z_1, z_2, \dots, z_n) \in \mathbb{R}^n. \end{aligned}$$

We further normalize the coefficients of each polynomial $g_k \in \mathbb{R}[\boldsymbol{z}]$ such that

$$g'_k(\boldsymbol{z}) \;\; = \;\; g_k(\boldsymbol{z})/\nu_k \text{ for every } \boldsymbol{z} = (z_1, z_2, \dots, z_n) \in \mathbb{R}^n.$$

Here ν_k denotes the maximum magnitude of the coefficients of the polynomial $g_k \in \mathbb{R}[\boldsymbol{z}]$ ($k = 0, 1, 2, \dots, m$). Conequently, we obtain a scaled POP which is equivalent to the POP (18) with the additional bounding constraint (36) on variables x_i ($i = 1, 2, \dots, n$).

$$\left. \begin{array}{ll} \text{minimize} & g'_0(\boldsymbol{z}) \\ \text{subject to} & g'_k(\boldsymbol{z}) \geq 0 \;(k = 1, 2, \dots, m), \;\; 0 \leq z_i \leq 1 \;(i = 1, 2, \dots, n). \end{array} \right\} \tag{37}$$

We note that the scaled POP (37) provides the same csp matrix as the original POP (18). Furthermore, we can add the constraints $0 \leq y_{\boldsymbol{\alpha}} \leq 1$ ($\boldsymbol{\alpha} \in \widetilde{\mathcal{F}}$) to its primal SDP (22) to strengthen the relaxation.

6 Numerical results

In this section, we present numerical results obtained from implementing the proposed sparse relaxation for unconstrained and constrained problems. The focus is on verifying the efficiency of the proposed sparse relaxation compared with the dense relaxation in [23]. The sparse and dense relaxations were implemented with MATLAB for constructing SDP problems and then a software package SeDuMi was used to solve the SDP problems. All the experiments were done on 2.4GHz Xeon cpu with 6.0 GB memory.

Various unconstrained and constrained optimization problems are used as test problems. Unconstrained problems that we dealt with, as shown in Section 6.1, are benchmark test problems from [4, 22, 27] and randomly generated test problems with artificial correlative sparsity. Constrained test problems whose results are presented in Section 6.2 are some problems from [10], optimal control problems [3], the maxcut problems, and randomly generated problems with artificial correlative sparsity.

We employ the techniques described in Section 5.1 for finding an optimal solution when testing the problems. In particular, we use the random perturbation techniques with the parameter $\epsilon = 10^{-5}$ in all the experiments presented here. After an optimal solution $\hat{\boldsymbol{y}}$ of an SDP relaxation of the POP is found by SeDuMi, the linear part $\hat{\boldsymbol{x}}$ is considered as a candidate of an optimal solution of the POP based on lemma 5.1.

With regard to computing the accuracy of an obtained solution, we use the following for an unconstrained POP with an objective function f_0.

$$\epsilon_{\text{obj}} = \frac{|\text{the optimal value of SDP} - (f_0(\hat{\boldsymbol{x}}) + \boldsymbol{p}^T\hat{\boldsymbol{x}})|}{\max\{1, |f_0(\hat{\boldsymbol{x}}) + \boldsymbol{p}^T\hat{\boldsymbol{x}}|\}}.$$

Here $\boldsymbol{p} \in \mathbb{R}^n$ denotes a randomly generated perturbation vector such that $|p_j| < \epsilon = 10^{-5}$ ($j = 1, 2, \dots, n$). If this value is close to 0, we decide that an optimal solution of the original unconstrained POP is obtained, and the POP is solved. For an inequality and equality constrained POP of the form (29), we need another measure for feasibility in addition to $\epsilon_{\rm obj}$ defined above. The following feasibility measure is used.

$$\epsilon_{\rm feas} = \min \{ f_k(\hat{\boldsymbol{x}}) \ (k = 1, \ldots, m), \ -|h_j(\hat{\boldsymbol{x}}) \ (j = 1, \ldots, q)| \}.$$

We regard $\hat{\boldsymbol{x}}$ as feasible for the original POP if this value is nonnegative, or close to 0.

We use the technique given in Section 5.2 for equality constraints and the technique in Section 5.3 for reducing the size of an SOS relaxation in all test problems. In addtion, we apply the rest of the techniques presented in Sections 5.4, 5.5 and 5.6 to constrained test problems from the literature [10]. Some of the problems are badly scaled, and some others involve the complementarity condition in their constraints. The techniques in Sections 5.4, 5.5 and 5.6 are actually motivated by resolving severe numerical difficulties arised from solving the dense and sparse relaxations of those problems.

Table 1 shows notation used in the description of numerical experiments in the following subsections. The notation cl.str shows the structure of the maximal cliques obtained by applying MATLAB functions 'symamd' and 'chol' to the csp matrix. For example, 4*3+5*2 means three cliques of size 4 and two cliques of size 5.

n	the number of variables of a POP
d	the degree of a POP
sparse	cpu time in seconds consumed by the proposed sparse relaxation
dense	cpu time in seconds consumed by the dense relaxation [23]
cl.str	the structure of the maximal cliques
#clique	the average number of cliques found in randomly generated problems
#solved	the number of problems that could be solved among randomly generated problems
#notSol	the number of problems that could not be solved among randomly generated problems
max.cl	the number of the maximal cliques
max	the maximum of cpu time consumed by randomly generated problems
avr	the average of cpu time consumed by randomly generated problems
min	the minimum of cpu time consumed by randomly generated problems
cpu	cpu time in seconds
ω	the relaxation order

Table 1: Notation

6.1 Unconstrained cases

We show the numerical results for unconstrained problems. The problems presented here are from the literatures [4, 22, 27] and randomly generated problems.

Table 2 displays the numerical results of the following two functions.

- The chained singular function [4]:

$$f_{\rm cs}(\boldsymbol{x}) = \sum_{i \in J} \left((x_i + 10x_{i+1})^2 + 5(x_{i+2} - x_{i+3})^2 + (x_{i+1} - 2x_{i+2})^4 + 10(x_i - 10x_{i+3})^4 \right)$$

where $J = \{1, 3, 5, \ldots, n-3\}$ and n is a multiple of 4.

- The Broyden banded function [22]:

$$f_{\rm Bb}(\boldsymbol{x}) = \sum_{i=1}^{n} \left(x_i(2+5x_i^2) + 1 - \sum_{j \in J_i} (1+x_j)x_j \right)^2$$

where $J_i = \{j \mid j \neq i, \max(1, i-5) \leq j \leq \min(n, i+1)\}$.

The above two problems of relatively small size could be solved by the dense relaxation as shown in Table 2, and their results can be used for the comparison of the performance of the sparse and dense relaxations. In the case of the chained singular function $f_{\rm cs}$, its csp matrix $\boldsymbol{R}$ has nonzero elements near the diagonal, *i.e.,* $R_{ij} = 0$ if $|j - i| > 3$. This means that $f_{\rm cs}$ is correlatively sparse. The 'cl.str' column of Table 2 proves that the sparsity can be detected correctly. As a result, the sparse relaxation is much more efficient than the dense relaxation. We could successfully solve the problem of 100 variables in a few seconds, while the dense relaxation could not handle the problem of 20 variables.

If we look at the result of the Broyden banded function $f_{\rm Bb}$ in Table 2, we observe that there is virtually no difference in performance between the proposed sparse and dense relaxations for $n = 6$ and $n = 7$. Because the csp matrix of this function has the bandwidth 7, it is fully dense when $n = 6$ and $n = 7$; the sparse relaxation is identical to the dense relaxation in these cases.

As n increases, however, a sparse structure such as 7*2 for $n = 8$ can be found and the sparse relaxation takes advantage of the structured sparsity providing an optimal solution faster than the dense relaxation. We could not obtain an optimal solution for $n = 9, 10$ since SeDuMi failed to solve the SDP problem of the sparse relaxation. In this case, we applied SDPA to the SDP problem and obtained an optimal solution in 200 and 600 seconds, respectively. The reason of failure in SeDuMi is unknown. SeDuMi also failed to solve the dense relaxation for $n = 10$ as a result of out of memory.

chained singular					Broyden banded				
n	cl.str	$\epsilon_{\rm obj}$	sparse	dense	n	cl.str	$\epsilon_{\rm obj}$	sparse	dense
12	3*10	1.1e-09	0.7	404.2	6	6*1	9.1e-10	20.5	20.4
16	3*14	9.0e-10	0.9	7523.1	7	7*1	2.4e-09	127.7	127.5
40	3*38	1.7e-09	2.1	—	8	7*2	4.2e-09	255.1	620.4
80	3*78	2.9e-04	1.8	—	9	7*3	1.0e+00	117.3.	3408.2
100	3*98	3.6e-04	2.2	—	10	7*4	1.0e+00	155.4	—

Table 2: Numerical results of the chained singular function and the Broyden banded function

In Tables 3, we present the numerical results of the following functions:

- The Broyden tridiagonal function [22]

$$\begin{aligned} f_{\rm Bt}(\boldsymbol{x}) &= ((3-2x_1)x_1 - 2x_2 + 1)^2 + \sum_{i=2}^{n-1} ((3-2x_i)x_i - x_{i-1} - 2x_{i+1} + 1)^2 \\ &\quad + ((3-2x_n)x_n - x_{n-1} + 1)^2 . \end{aligned}$$

- The chained wood function [4]:

$$\begin{aligned} f_{\rm cw}(\boldsymbol{x}) &= 1 + \sum_{i \in J} \big(100(x_{i+1} - x_i^2)^2 + (1-x_i)^2 + 90(x_{i+3} - x_{i+2}^2)^2 + (1-x_{i+2})^2 \\ & \quad +10(x_{i+1} + x_{i+3} - 2)^2 + 0.1(x_{i+1} - x_{i+3})^2\big), \end{aligned}$$

  where $J = \{1, 3, 5, \ldots, n-3\}$ and n is a multiple of 4.

- The generalized Rosenbrock function [27]:

$$f_{\rm gR}(\boldsymbol{x}) = 1 + \sum_{i=2}^{n} \left\{ 100\left(x_i - x_{i-1}^2\right)^2 + (1-x_i)^2 \right\}.$$

Each of the above three functions has a band structure in its csp matrix, and therefore, the problems of large sizes can be handled efficiently. For example, the Broyden tridiagonal function $f_{\rm Bt}$ with 500 variables could be solved in 11.2 seconds with the accuracy of 6.3e-09. Without utilizing the correlative sparsity of the functions, it is not possible to obtain optimal solutions of the smallest-sized problem in Table 3, as we experienced with the dense relaxation in the previous two problems. Note that the solutions are accurate in all tested cases.

	Broyden tridiagonal			chained wood			generalized Rosenbrock		
n	cl.str	$\epsilon_{\rm obj}$	sparse	cl.str	$\epsilon_{\rm obj}$	sparse	cl.str	$\epsilon_{\rm obj}$	sparse
100	3*98	4.1e-9	2.4	2*99	3.9e-7	0.4	2*99	2.6e-5	0.9
200	3*198	5.4e-6	4.7	2*199	8.1e-7	0.7	2*199	1.6e-5	1.8
300	3*298	5.0e-9	6.6	2*299	1.2e-6	1.1	2*299	3.0e-5	2.5
400	3*398	1.1e-8	11.0	2*399	1.6e-06	1.4	2*399	1.2e-4	3.3
500	3*498	6.3e-9	10.2	2*499	2.1e-6	1.7	2*499	4.3e-5	4.5

Table 3: Numerical results of Broyden tridiagonal function, the chained wood function and the generalized Rosenbrock function

Next, we present the numerical results of randomly generated problems. The aim of the test using randomly generated problems is to observe the effects of increasing the number of variables, the degree of the polynomials as well as the maximal size of cliques of the csp graph of a POP. The dense relaxation could not handle the randomly generated problems of the sizes reported here, and we include only the numerical results from the sparse relaxation.

Let us describe how an unconstrained problem with artificial correlative sparsity is generated randomly. We begin by constructing a chordal graph randomly such that the size of every maximal clique is not less than 2 and not greater than max.cl. From the chordal graph, we derive the set of maximal cliques $\{C_1, \ldots, C_\ell\}$ with $2 \leq |C_i| \leq$ max.cl $(i = 1, \ldots, \ell)$. We let $\boldsymbol{v}_{C_i}(\boldsymbol{x}) = (x_k^d \colon k \in C_i)$ where $2d$ is the degree of the polynomial, and generate a positive definite matrix $\boldsymbol{V}_i \in \mathcal{S}_{++}(C_i)$ and a vector $\boldsymbol{g}_i \in [-1, 1]^{\#\mathcal{A}_{2d-1}^{C_i}}$ $(i = 1, 2, \ldots, \ell)$ randomly such that the minimum eigenvalue σ of $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_\ell$ satisfies the following relation:

$$\sigma \geq \sum_{i=1}^{\ell} \left(\|\boldsymbol{g}_i\|_2 \sqrt{\#\mathcal{A}_{2d-1}^{C_i}} \right).$$

By using $\boldsymbol{V}_i$ and $\boldsymbol{g}_i$, we define the objective function:

$$f_{\text{rand}}(\boldsymbol{x}) = \sum_{i=1}^{\ell} \left(\boldsymbol{v}_{C_i}(\boldsymbol{x})^T \boldsymbol{V}_i \boldsymbol{v}_{C_i}(\boldsymbol{x}) + \boldsymbol{g}_i^T \boldsymbol{u}(\boldsymbol{x}, \mathcal{A}_{2d-1}^{C_i}) \right).$$

It is easy to see that this unconstrained POP is guaranteed to have an optimal solution in the compact set $\{\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n \mid \max_{i=1,\ldots,n} |x_i| \leq 1\}$. A scaling with the maximum of the absolute values of the coefficients of $f_{\text{rand}}(\boldsymbol{x})$ is used in numerical experiments.

The numerical results are shown in Tables 4, 5 and 6. Table 4 exhibits how the sparse relaxation performs for varying number of variables, Table 5 for raising the degree of the unconstrained problems, and Table 6 for increasing bounds of sizes of the cliques. For each choice of n, d and max.cl, we generated 50 problems. Each column of #solved indicates the number of the problems whose optimal solutions were obtained with $\epsilon_{\text{obj}} \leq 10^{-5}$ out of 50 problems. We mention that all problems tested were solved.

n	#clique	max	avr	min	#solved
20	14.0	1.4	0.5	0.3	50/50
40	28.7	3.1	1.3	0.7	50/50
60	43.0	6.0	2.5	1.2	50/50
80	57.1	30.4	6.1	2.0	50/50
100	71.7	19.1	6.5	2.7	50/50

Table 4: Randomly generated polynomials with max.cl= 4 and $d = 4$

d	#clique	max	avr	min	#solved
4	21.3	2.0	0.9	0.5	50/50
6	21.0	168.4	15.2	1.9	50/50
8	21.4	1693.4	128.9	3.0	50/50

Table 5: Randomly generated polynomials with max.cl= 4, and $n = 30$

max.cl	#clique	max	avr	min	#solved
4	21.3	2.0	0.9	0.5	50/50
6	18.3	91.1	8.4	1.3	50/50
8	16.9	825.8	121.4	4.4	50/50

Table 6: Randomly generated polynomials with $d = 4$ and $n = 30$

In Table 4, we notice that the number of cliques increases with n. For problems of large numbers of variables and cliques such as $n = 100$ and #clique= 71.74, the sparse relaxation provides optimal solutions in most cases. The rate of success in obtaining an optimal solution remains relatively unchanged for increasing n.

The numerical results in Table 5 displays the performance of the sparse relaxation for the problem of $n = 30$ with degrees up to 8. The maximum size of cliques is fixed to 4. As mentioned before, the size of the SDP relaxation of the POP of increasing degree becomes large rapidly even if the POP remains correlatively sparse. When $d = 8$, the average cpu time is 128.9 and the maximum is 1693.4.

A large size of cliques used when a problem is generated also increases the complexity of the problem as shown in Table 6. We tested with the maximum size of cliques 4, 6, and 8, and observe that cpu time to solve the corresponding problems grows very rapidly, *e.g.* 121.4 average cpu seconds, 825.8 maximum cpu seconds for max.cl = 8. From the increase of work measured by cpu time, we mention that the impact of the maximum size of cliques is comparable to that of degree, and bigger than that of the number of variables.

6.2 Constrained cases

In this subsection, we deal with the following constrained POPs:

- Small-sized POPs from the literature [10].
- Optimal control problems.
- Randomly generated maxcut problems.
- Randomly generated POPs.

The numerical results are presented in Tables 7 and 8. All the problems are quadratic optimization problems (QOPs) except 'alkyl' which involves polynomials of degree 3 in its equality constraints. In preliminary numerical experiment for some of the test problems, we encountered severe numerical difficulties for badly-scaled problems and/or problems with the complementarity condition. We incorporate all the techniques in Sections 5.4, 5.5 and 5.6 into the dense and sparse relaxations for these problems. Specifically, we replaced each support $\mathcal{A}^{F_k}_{\omega-\omega_k}$ for the Lagange multiplier polynomial φ_k by the union C of all cliques C_ℓ containing F_k as mentioned in Section 5.5, and added finite lower and upper bounds to all the variables of each problem so that the scaling technique and the valid inequalities of the form $0 \leq y_{\boldsymbol{\alpha}} \leq 1$ given in Section 5.6 can work effectively. In addition, all the equality constraints were converted to two inequality constraints such that

$$f(\boldsymbol{x}) = 0 \implies f(\boldsymbol{x}) \geq 0 \text{ and } -f(\boldsymbol{x}) + \kappa \geq 0,$$

where $\kappa = 10^{-5}$. Without these techniques, we could not solve many of the problems in Tables 7 and 8.

The problems in Table 7 have known optimal values, hence, we compare the lower bounds obtained by the sparse or dense relaxation to their optimal value in column $\epsilon_{\rm opt}$, which denotes

$$\epsilon_{\rm opt} = \frac{|\text{the optimal value of SDP} - \text{the known optimal value of POP}|}{\max\{1,\ |\text{the optimal value of POP}|\}}.$$

In all cases, the sparse and dense relaxations attain reasonable accuracy with $\epsilon_{\rm opt} \leq$ 5.0e-2. $\epsilon'_{\rm feas}$ denotes the feasibility for the scaled problems at the approximate optimal solutions obtained by the sparse and dense relaxations. We see that $\epsilon'_{\rm feas}$ is small in most of the problems while the feasibility $\epsilon_{\rm feas}$ for the original problems at the approximate optimal solutions becomes larger. The lower bounds obtained by the sparse relaxation are as good as the ones by the dense relaxation except the four problems ex5_2_2_cases1, 2, 3 and ex5_3_2. In the former three cases, the dense relaxation succeeds in computing accurate bounds while the sparse relaxation with order $\omega = 3$ computes accurate bounds with the same quality.

The problems in Table 8 have no known optimal value or their best known optimal values are turned out to be inaccurate. The column SDPval denotes the lower bounds obtained by the sparse or dense relaxation. In all cases, the attained accuracy $\epsilon_{\rm obj}$ and feasibility $\epsilon'_{\rm feas}$ for the scaled problems are very small. Hence we can conclude that the column SDPval provides tight lower bounds for the optimal values of the test problems.

When we compare the performance of the sparse relaxation with the dense relaxation using these problems in Tables 7 and 8, we observe that the sparse relaxation are much faster than the dense relaxation in large dimensional problems.

We should also mention that the technique given in Section 5.3 for reducing sizes of relaxations worked very effectively. For example, the sparse and dense relaxations of ex2_1_3 without incorporating this technique took 2.6 and 351.6 seconds, respectively, while 0.9 and 13.6 seconds were consumed with this technique, respectively, as shown in Table 7.

We present numerical results from the discrete-time optimal control problems in [3]. The problem tested first (5 of [3]) is

$$\left.\begin{array}{ll} \min & \displaystyle\sum_{i=1}^{M-1}\left(\sum_{j=1}^{n_y}\left(y_{i,j}+\frac{1}{4}\right)^4+\sum_{j=1}^{n_x}\left(x_{i,j}+\frac{1}{4}\right)^4\right)+\sum_{j=1}^{n_y}\left(y_{M,j}+\frac{1}{4}\right)^4 \\ \text{subject to} & \boldsymbol{y}_{i+1}=\boldsymbol{A}\boldsymbol{y}_i+\boldsymbol{B}\boldsymbol{x}_i+(\boldsymbol{y}_i^T\boldsymbol{C}\boldsymbol{x}_i)\boldsymbol{e}\ (i=1,\ldots,M-1),\ \boldsymbol{y}_1=\boldsymbol{0} \\ & \boldsymbol{y}_i\in\mathbb{R}^{n_y},\quad (i=1,\ldots,M),\ \boldsymbol{x}_i\in\mathbb{R}^{n_x},\quad (i=1,\ldots,M-1) \end{array}\right\} \quad (38)$$

where $\boldsymbol{A} \in^{n_y\times n_y}$, $\boldsymbol{B}\in\mathbb{R}^{n_y\times n_x}$, and $\boldsymbol{C}\in\mathbb{R}^{n_y\times n_x}$ are given by:

$$A_{i,j}=\left\{\begin{array}{ll} 0.5 & \text{if } j=i, \\ 0.25 & \text{if } j=i+1, \\ -0.25 & \text{if } j=i-1, \end{array}\right. \qquad B_{i,j}=\frac{i-j}{n_y+n_x} \quad C_{i,j}=\mu\frac{i+j}{n_y+n_x},$$

respectively. Here, $\boldsymbol{e}$ denotes a vector of ones in $\mathbb{R}^{n_y}$.

The numerical results of the problem (38) are shown in Table 9, 10 and 11, which display the results of the problem (38) with $(n_x, n_y, \mu) = (2, 4, 0)$, the problem (38) with $(n_x, n_y, \mu) = (2, 4, 0.5)$ and the problem (38) with $(n_x, n_y, \mu) = (1, 2, 1)$, respectively. The values of n_x, n_y and M determine the size of the problem and μ is a parameter in $C_{i,j}$. The relaxation order 2 was used for all cases. As we increase M from 6 to 30, the number of variables becomes bigger as indicated in the column of n. In all cases, the optimal solutions are obtained with good accuracy.

Depending on the choice of μ, it results in different clique structures and the size of the resulting SDP varies. This size can affect greatly the performance of the relaxations. When we take $\mu = 0$ in (38), the constraints are linear since $\boldsymbol{C} = \boldsymbol{O}$. Then, the cliques has smaller number of elements than the ones from the constraints with nonlinear terms, which enables the sparse relaxation to perform better in terms of cpu time. To see this, compare the column of cl.str of Table 9 with that of Table 10. For example, when $M = 30$ in Table 9, it took 135.9 cpu seconds to have an optimal solution whereas 712.0 seconds in Table 10. Similarly, if we compare Table 10 and 11, we notice that the size of the cliques in Table 11 is half the size in Table 10, while the cpu time of Table 11 is less than 1/100 times than the

			sparse					dense				
problem	n	ω	$\epsilon_{\rm opt}$	$\epsilon_{\rm obj}$	$\epsilon_{\rm feas}$	$\epsilon'_{\rm feas}$	cpu	$\epsilon_{\rm opt}$	$\epsilon_{\rm obj}$	$\epsilon_{\rm feas}$	$\epsilon'_{\rm feas}$	cpu
ex2_1_1	5	2	0.5e-01	1.9e+00	0.0e+00	0.0e+00	0.2	0.5e-01	1.9e+00	0.0e+00	0.0e+00	0.2
ex2_1_1	5	3	9.3e-10	3.0e-06	0.0e+00	0.0e+00	1.8	6.6e-06	3.0e-06	0.0e+00	0.0e+00	1.9
ex2_1_2	6	2	4.6e-06	3.3e-11	-3.2e-10	1.6e-11	0.2	4.6e-06	4.4e-11	0.0e+00	0.0e+00	0.3
ex2_1_3	13	2	1.3e-06	5.1e-09	-3.5e-09	-4.4e-10	0.5	1.3e-06	1.6e-09	-1.5e-09	-1.8e-10	7.7
ex2_1_4	6	2	2.2e-06	2.4e-12	-4.7e-11	-2.9e-12	0.3	2.2e-06	2.4e-12	-4.6e-11	-2.9e-12	0.3
ex2_1_5	10	2	3.7e-09	4.4e-11	-5.4e-10	-6.2e-11	1.9	3.7e-09	4.4e-11	-5.4e-10	-6.2e-11	1.8
ex3_1_1	8	2	0.7e-01	0.0e+00	-4.7e+04	-5.8e-02	0.6	0.7e-01	0.0e+00	-4.6e+04	-5.8e-02	2.6
ex3_1_1	8	3	6.3e-09	0.0e+00	-6.5e-02	-2.2e-08	5.5	2.2e-07	0.0e+00	-2.0e-01	-6.9e-08	597.8
ex3_1_2	5	2	3.0e-06	9.9e-11	-1.4e-07	-4.6e-09	0.7	3.0e-06	9.9e-11	-1.4e-07	-4.6e-09	0.7
ex5_2_2_case1	9	2	0.1e-01	0.0e+00	-3.2e+01	-1.3e-04	1.8	1.6e-05	0.0e+00	-2.1e-01	-8.4e-07	3.4
ex5_2_2_case1	9	3	6.6e-04	0.0e+00	-2.3e-01	-9.1e-07	295.9	—	—	—	—	—
ex5_2_2_case2	9	2	0.1e-01	0.0e+00	-7.2e+01	-2.9e-04	2.1	1.3e-04	0.0e+00	-2.7e-01	-1.1e-06	3.5
ex5_2_2_case2	9	3	5.8e-04	0.0e+00	-8.9e-01	-3.6e-06	332.9	—	—	—	—	—
ex5_2_2_case3	9	2	0.3e-01	0.0e+00	-6.7e+01	-2.7e-04	1.9	1.6e-05	0.0e+00	-1.1e-01	-4.4e-07	2.6
ex5_2_2_case3	9	3	2.8e-04	0.0e+00	-6.0e+00	-2.4e-05	336.5	—	—	—	—	—
ex5_3_2	22	2	0.3e-01	0.0e+00	-4.1e+00	-1.4e-02	53.2	2.7e-05	0.0e+00	-1.7e-06	-5.7e-09	2120.6
ex5_4_2	8	2	0.5e+00	0.0e+00	-4.3e+05	-4.3e-02	0.6	0.5e+00	0.0e+00	-4.3e+05	-4.3e-02	2.4
ex5_4_2	8	3	5.2e-06	0.0e+00	-3.2e-01	-3.2e-08	8.3	5.8e-06	0.0e+00	-8.1e-01	-8.1e-08	757.42
ex9_1_1	13	2	6.2e-06	0.0e+00	-4.5e-06	-1.1e-08	1.5	6.2e-06	0.0e+00	-9.2e-07	-2.3e-09	7.7
ex9_1_2	10	2	5.6e-06	0.0e+00	-4.1e-06	-1.7e-07	2.8	5.6e-06	0.0e+00	-2.6e-07	-1.0e-08	2.1
ex9_1_5	13	2	2.3e-04	0.0e+00	-7.2e-05	-2.9e-06	1.0	2.3e-04	0.0e+00	-5.0e-05	-2.0e-06	7.6
ex9_1_8	14	2	7.9e-06	0.0e+00	-4.1e-01	-4.1e-02	1.3	7.9e-06	0.0e+00	-4.1e-01	-4.1e-02	20.2
ex9_2_1	10	2	2.2e-05	1.3e-06	-4.9e-05	-3.5e-07	1.2	2.2e-05	6.8e-08	-3.2e-06	-2.1e-08	2.5
ex9_2_2	10	2	3.0e-04	1.0e-05	-5.2e+00	-1.3e-02	1.6	2.5e-04	8.6e-06	-3.2e+01	-7.9e-02	6.0
ex9_2_2	10	3	2.1e-04	5.3e-06	-8.0e-01	-2.0e-03	36.0	2.9e-04	7.6e-06	-2.5e+01	-6.4e-02	1592.4
ex9_2_5	8	2	2.3e-06	6.2e-07	-8.0e-06	-8.0e-08	0.6	2.5e-06	7.2e-08	-1.2e-06	-1.2e-08	0.9
ex9_2_6	16	2	2.0e-04	1.2e-05	-1.6e+03	-4.1e-02	0.9	2.0e-04	7.2e-06	-3.8e+03	-9.5e-02	69.1
ex9_2_6	16	3	2.0e-04	9.8e-06	-9.3e+02	-2.3e-02	8.6	—	—	—	—	—
ex9_2_7	10	2	2.3e-05	4.7e-07	-2.0e-05	-1.3e-07	1.3	2.2e-05	2.2e-09	-4.0e-07	-2.7e-09	2.6
ex9_2_8	6	2	7.7e-06	7.9e-07	-4.2e-06	-1.0e-08	0.4	8.2e-06	2.5e-07	-3.6e-06	-9.0e-09	0.4

Table 7: The results on some problems in [10] whose optimal value is known.

			sparse					dense				
problem	n	ω	SDPval	ϵ_{obj}	ϵ_{feas}	ϵ'_{feas}	cpu	SDPval	ϵ_{obj}	ϵ_{feas}	ϵ'_{feas}	cpu
alkyl	14	2	-2.42e+00	2.0e-03	-2.5e-01	-1.1e-02	6.7	-2.41e+00	7.3e-06	-3.2e-02	-1.3e-03	65.7
alkyl	14	3	-2.41e+00	9.0e-09	-3.0e-08	-8.3e-10	5216.2	—	—	—	—	—
ex2_1_8	24	2	1.56e+04	1.0e-05	0.0e+00	0.0e+00	304.6	1.56e+04	3.4e-06	0.0e+00	0.0e+00	1946.6
ex9_2_3	16	2	1.85e+01	0.0e+00	-5.7e-06	-9.6e-10	2.3	1.85e+01	0.0e+00	-7.5e-06	-1.2e-09	49.7
ex9_2_4	8	2	3.96e+00	7.5e-04	-3.1e-02	-7.8e-07	1.5	3.97e+00	2.7e-04	-1.6e-02	-4.0e-07	1.8
st_bpaf1a	10	2	-4.54e+01	5.5e-08	-8.5e-09	-4.7e-11	1.4	-4.54e+01	7.6e-09	-2.8e-09	-1.5e-11	1.9
st_bpaf1b	10	2	-4.30e+01	3.8e-08	-2.8e-08	-2.0e-10	1.0	-4.30e+01	4.6e-09	-7.2e-10	-1.7e-11	1.7
st_e05	5	2	7.05e+03	0.0e+00	-2.2e+00	-1.9e-09	0.3	7.05e+03	0.0e+00	-2.3e-01	-2.0e-10	0.4
st_e07	10	2	-1.81e+03	0.0e+00	-8.1e-05	-4.0e-09	0.4	-1.81e+03	0.0e+00	-8.8e-06	-4.4e-10	3.0
st_jcbpaf2	10	2	-7.95e+02	1.1e-07	0.0e+00	0.0e+00	2.1	-7.95e+02	1.1e-07	0.0e+00	0.0e+00	2.0

Table 8: Numerical results for small-sized problems with unknown optimal values from the literatures [10].

cpu time of Table 10. This shows how much efficiency can be improved using appropriate clique structures.

M	n	cl.str	ϵ_{obj}	ϵ_{feas}	cpu
6	30	4*1+5*4+6*2+7*3+8*6	2.5e-10	-6.9e-11	15.7
12	66	4*1+5*4+6*2+7*9+8*18	9.0e-10	-8.2e-11	45.4
18	102	4*1+5*4+6*2+7*15+8*30	5.0e-09	-2.1e-10	73.6
24	138	4*1+5*4+6*2+7*21+8*42	1.0e-08	-2.8e-10	96.1
30	174	4*1+5*4+6*2+7*27+8*54	7.9e-09	-1.2e-10	135.7

Table 9: Numerical results for the problem (38) with $(n_x, n_y, \mu) = (2, 4, 0)$. The relaxation order $\omega = 2$.

M	n	cl.str	ϵ_{obj}	ϵ_{feas}	cpu
6	30	5*2+7*4+10*3	1.8e-09	-6.3e-10	76.7
12	66	5*2+7*4+10*9	2.9e-09	-7.7e-10	223.5
18	102	5*2+7*4+10*15	1.1e-09	-3.5e-10	369.4
24	138	5*2+7*4+10*21	3.2e-09	-1.0e-09	518.0
30	174	5*2+7*4+10*27	3.5e-09	-8.1e-10	663.7

Table 10: Numerical results for the problem (38) with $(n_x, n_y, \mu) = (2, 4, 0.5)$. The relaxation order $\omega = 2$.

The second problem (5 of [3]) is

$$\left.\begin{array}{ll} \text{min} & \dfrac{1}{M}\displaystyle\sum_{i=1}^{M-1}\left(y_i^2 + x_i^2\right) \\ \text{subject to} & y_{i+1} = y_i + \frac{1}{M}(y_i^2 - x_i), \quad (i = 1, \ldots, M-1), \quad y_1 = 1. \end{array}\right\} \tag{39}$$

Table 12 shows the results of (39) for various M. From the column cl.str, we notice that the

M	n	cl.str	ϵ_{obj}	ϵ_{feas}	cpu
6	15	2*1+4*2+5*3	1.3e-09	-3.9e-11	0.5
12	33	2*1+4*2+5*9	1.8e-09	-3.0e-11	1.1
18	51	2*1+4*2+5*15	6.9e-09	-7.2e-11	1.7
24	69	2*1+4*2+5*21	1.0e-08	-1.1e-10	2.4
30	87	2*1+4*2+5*27	1.2e-08	-9.9e-11	3.0

Table 11: Numerical results for the problem (38) with $(n_x, n_y, \mu) = (1, 2, 1)$. The relaxation order $\omega = 2$.

set of cliques has very few elements. The sparse relaxation can solve large-sized problems since they have plenty of correlative sparsity. In fact, the sparse relaxation provides an optimal solutions for the problems with almost 2000 variables, where the size of clique is 2 or 3. It should be noted that the problem (39) is a QOP and that the relaxation order 1 was used. In view of the discussion given in Section 5.4, the sparse relaxation is theoretically guaranteed to provide bounds with the same quality as the dense relaxation in all the cases of Table 12.

M	n	cl.str	ϵ_{obj}	ϵ_{feas}	cpu
600	1198	2*1+3*598	3.4e-08	-2.2e-10	3.4
700	1398	2*1+3*698	2.5e-08	-8.1e-10	3.3
800	1598	2*1+3*798	5.9e-08	-1.6e-10	3.8
900	1798	2*1+3*898	1.4e-07	-6.8e-10	4.5
1000	1998	2*1+3*998	6.3e-08	-2.7e-10	5.0

Table 12: Numerical results from problem (39). The relaxation order $\omega = 1$.

We now consider the maxcut problem. Suppose that an undirected graph $G = (V, E)$ is given, where $V = \{1, \ldots, n\}$ is the set of nodes and E is the set of edges, respectively. For a subset $S \subseteq V$, a set of edges $\{\{i, j\} \in E | i \in S, j \notin S\}$ is called a *cut*. Then the maxcut problem is to find a cut that has the maximum number of edges. We formulate the maxcut problem as an equality constrained POP:

$$\left.\begin{array}{ll} \text{maximize} & \displaystyle\sum_{i\in V, j\in V} (1 - x_i x_j)/2 \\ \text{subject to} & x_i^2 = 1 \quad (i \in V). \end{array}\right\} \tag{40}$$

See [9] or [26] for details of this formulation.

In [24] and [25], it is shown that a further reduction of the variables is possible in the SOS relaxation if we deal with the integer equality constraints directly. However, we do not use this technique because the purpose of this experiment is to estimate the effect of using correlative sparsity of POPs. To incorporate techniques only useful for integer programming is not the subject of this paper.

The csp matrix of (40) has the same sparsity pattern as the adjacency matrix of G with the diagonal. Therefore, the sparse relaxation can be applied efficiently when the chordal extension of the graph is sparse. To obtain such a sparse graph, we generate edges in a band of the matrix. Specifically, we use two parameters, band-width b and rate r to generate a random graph. We first connect nodes i and $i + 1$ ($i = 1, 2, \ldots, n - 1$), and then nodes i and j such that $1 < |i - j| \leq b$ with probability r. In this experiment, we choose $b = 7$, and $r = 0.3333$. Each node has about six edges on average with this parameter selection.

Notice that in Tables 13 and 14, n, the number of variables of POP, is equal to the size of nodes. For each size of nodes, we performed 10 trials. A problem is considered to be *solved* when $\epsilon_{\text{obj}} < 10^{-3}$ and $\epsilon_{\text{feas}} < 10^{-3}$. The relaxation order is fixed to 2, which was decided by some preliminary experiments on what value of relaxation order should be used to obtain solutions. Most of the small-sized problems in Table 13 could be solved to the optimum with the relaxation order 2 and none of the problems was solved with the relaxation order 1.

We observe in Table 13 that the cpu time is significantly reduced for the sparse relaxation compared with the dense relaxation. In all problems, the optimal values of the dense and sparse relaxations coincide up to 9 digits, although theoretically this is not ensured for the sparse relaxation with the relaxation order 2.

We can also see that the sparse relaxation is less influenced by the size of the graph, while in the dense relaxation, the difference in the size of nodes by just one is critical to the cpu time.

	sparse				dense				
n	#clique	max	ave	min	max	ave	min	#solved	#notSol
10	5.9	1.3	0.7	0.3	32.9	28.4	20.8	10/10	0/10
11	6.5	2.8	1.1	0.4	136.7	84.3	50.8	9/10	1/10
12	7.4	2.0	1.1	0.3	305.4	181.8	127.4	10/10	0/10
13	7.8	4.6	1.8	0.8	863.7	443.3	257.6	9/10	1/10
14	8.9	3.0	1.8	0.8	1627.7	1003.7	540.9	9/10	1/10
15	9.6	4.0	2.3	0.9	2456.3	1975.4	1274.0	10/10	0/10

Table 13: Numerical results from the maxcut problem. The relaxation order $\omega = 2$.

Table 14 shows the results of the sparse relaxation with larger size of nodes. The problem with 120 nodes can be solved in less than 300 seconds on average. The average cpu time increases by $O(n^{1.6})$ approximately. However, the accuracy of the solution deteriorates as the size of nodes increases.

n	#clique	max	avr	min	#solved	#notSol
60	47.2	15.9	41.5	15.9	7/10	3/10
80	64.0	136.3	71.6	29.6	2/10	8/10
100	79.5	213.2	96.1	49.2	1/10	9/10
120	95.3	218.0	134.4	75.8	1/10	9/10

Table 14: Sparse relaxation with increasing number of nodes in the maxcut problem. The relaxation order $\omega = 2$.

Finally, we present the numerical results from randomly generated constrained problems:

$$\left.\begin{array}{ll} \min & \sum_{j=1}^{\ell} \boldsymbol{f}_j^T \boldsymbol{u}(\boldsymbol{x}, \mathcal{A}_{2d}^{C_j}) \\ \text{subject to} & \boldsymbol{v}_{C_j}(\boldsymbol{x})^T \boldsymbol{V}_j \boldsymbol{v}_{C_j}(\boldsymbol{x}) + \boldsymbol{g}_j^T \boldsymbol{u}(\boldsymbol{x}, \mathcal{A}_{2d-1}^{C_j}) \leq 0 \quad (j = 1, \ldots, \ell), \end{array}\right\} \tag{41}$$

where the set $\{C_1, \ldots, C_\ell\}$ is a set of maximal cliques generated in the same way as randomly generated unconstrained test problems, $\boldsymbol{v}_{C_j}(\boldsymbol{x}) = (x_k^d \colon k \in C_j)$, $\boldsymbol{f}_j$ are randomly generated in $[-1, 1]^{\#\mathcal{A}_{2d}^{C_j}}$, and positive definite matrices $\boldsymbol{V}_j \in \mathcal{S}_{++}(C_j)$ and $\boldsymbol{g}_j$ $(j = 1, \ldots, \ell)$ are randomly generated such that the minimum eigenvalue of $\boldsymbol{V}_j$ is greater than $\|\boldsymbol{g}_j\| \times \sqrt{\#\mathcal{A}_{2d-1}^{C_j}}$.

The value of the relaxation order needs to be chosen prior to solving a problem. However, its proper value is not known in many problems. As mentioned in Secion 5.2, raising the relaxation order gradually provides more accurate solutions, but it takes increasingly longer to solve. In this experiment, the problem is solved with an initial relaxation order, and if it is successful to get a solution within required accuracy, it is regarded as *solved*. Otherwise, the relaxation order is raised by 1 and the problem is tried again. We generated 10 problems for each choice of n. In every experiment, the tolerance for both $\epsilon_{\rm obj}$ and $\epsilon_{\rm feas}$ is set to 10^{-5}, and the initial relaxation order 2. For the problems that could not be solved, we increase the relaxation order by 1, and apply the sparse relaxation again. As shown in Table 17, for $n = 10$, 7 out of 10 problems were tried again with increased relaxation order 3. It is important to solve the problem with low relaxation order because increasing the relaxation order by 1 needs a great deal of additional cpu time. This repetitive strategy is appropriate when proper relaxation order is not known in advance.

n	ω	#clique	max	avr	min	#solved	#notSol
10	2	6.4	0.5	0.3	0.2	4/10	6/10
	3		1.1	0.7	0.3	6/6	0/6
20	2	14.0	1.1	0.7	0.5	0/10	10/10
	3		4.7	3.0	1.8	9/10	1/10
	4		37.3	37.3	37.3	0/1	1/1
30	2	21.6	1.7	1.0	0.6	0/10	10/10
	3		30.1	8.7	2.6	8/10	2/10
	4		67.7	58.1	48.5	0/2	2/2
40	2	28.7	2.1	1.5	0.9	0/10	10/10
	3		43.6	17.2	4.1	9/10	1/10
	4		595.8	595.8	595.8	1/1	0/1
50	2	36.5	3.5	2.0	1.4	0/10	10/10
	3		151.1	40.5	8.0	7/10	3/10
	4		4320.2	3101.3	677.3	1/3	2/3

Table 15: Problem (41) with max.cl = 4 and $d = 4$

It is shown in Tables 15, 16, and 17 that approximately 80 % of the problems were solved by raising the relaxation order except the case $n = 50$ in Table 15. There remained some problems that could not be solved by the sparse relaxation. The main reason of the failure is that the obtained optimal solutions were not within the required accuracy. For example, for $n = 50$ in Table 15, six problems could not be solved with the accuracy 1.0e-5. However, the worst accuracy obtained among the six problems was 4.5e-4. This leads us to say that the sparse relaxation gives at least good lower bounds for these problems.

From all numerical experiments in the previous and this subsections, we have observed that the sparse relaxation is much faster than the dense relaxation while giving relatively accurate solutions. The sparse relaxation can handle large POPs with more than hundred variables, which is not possible for the dense relaxation. The correlative sparsity has been the key to solve such large problems.

7 Concluding discussions

Solving POPs has been investigated by focusing on the sparsity of the POPs. The structured sparsity called correlative sparsity of POPs has been defined as a special type of

			cpu time				
d	ω	#clique	max	avr	min	#solved	#notSol
4	2	10.4	0.5	0.4	0.2	2/10	8/10
	3		3.1	1.5	0.6	7/8	1/8
	4		24.5	24.5	24.5	1/1	0/1
6	3	9.7	8.7	2.7	1.1	0/10	10/10
	4		137.9	28.7	4.8	9/10	1/10
	5		86.7	86.7	86.7	0/1	1/1

Table 16: Problem (41) with max.cl = 4 and $n = 15$

max.cl	ω	#clique	max	avr	min	#solved	#notSol
4	2	10.4	0.5	0.4	0.2	2/10	8/10
	3		3.1	1.5	0.6	7/8	1/8
	4		24.5	24.5	24.5	1/1	0/1
6	3	8.3	6.1	2.8	0.6	2/10	8/10
	4		870.5	344.6	4.2	8/8	0/8

Table 17: Problem (41) with $d = 4$ and $n = 15$

sparsity and utilized to propose efficient sparse SOS and SDP relaxations. Specifically, it is shown that the sparse relaxation of order 1 approximates the optimal value of QOPs with correlative sparsity with the same accuracy as the dense relaxation of order 1. We have also addressed some technical issues such as computing optimal solutions from an attained optimal solutions of the sparse SDP relaxation and how to formulate SDP relaxations for POPs with equality constraints. The performance of the proposed sparse SDP relaxation has been tested with various unconstrained and constrained problems and proved to have computational advantage over the dense relaxation. In particular, the proposed sparse relaxation was successful to efficiently handle some unconstrained and constrained problems that were impossible to obtain optimal solutions with the dense relaxation, *e.g.* the generalized Rosenblock function with dimension $n = 500$ and a QOP with dimension $n = 1998$ arising from discrete optimal control. This demonstrates the efficiency and effectiveness of the proposed sparse relaxation for POPs with correlative sparsity.

The proposed sparse relaxation for a correlative sparse POP leads to an SDP that can maintain the sparsity for primal-dual interior-point methods. This is based on the fact that if a POP to be solved is correlatively sparse, the resulting SDP relaxation inherits the structured sparsity. In each iteration of a primal-dual interior-point method for solving an SDP, a square system of linear equations, which is often called the Schur complement equation, is solved to compute a search direction. The coefficient matrix of this system is positive definite and fully dense in general even when all data matrices of an SDP to be solved are sparse. However, the sparse SDP relaxation of a correlatively sparse POP possesses sparsity in the coefficient matrix. This is an important advantage of the proposed sparse relaxation. Indeed, the coefficient matrix of the Schur complement equation in most of SDPs solved in Section 6 is sparse. Among software packages implementing primal-dual interior-point methods, SeDuMi [33] handles SDPs with this sparsity in the coefficient matrix of the Schur complement equation and provides solutions with efficiency while the current version of SDPA [34] developed by the authors' group is not equipped with the sparse Cholesky factorization for the Schur complement equation, showing slow performance for POPs with the correlative sparsity. This is the main reason that SeDuMi has been a choice of the numerical experiments instead of SDPA.

We encountered severe numerical difficulties during preliminary numerical experiments. The techniques presented in Section 5 were very effective to overcome the difficulties and to enhance the performance of the sparse and dense relaxations. The three problems ex9_1_4, 'haverly' and 'house' from [10], however, could not be solved because of numerical troubles resulted from SeDuMi. The failure has to be investigated more rigorously, but some SDPs generated as relaxations of POPs may be very difficult to solve. Additional techniques to resolve this difficulty are to be developed. The size of the SDP relaxation of a POP continue to play an an important role to obtain a solution successfully, although applying the sparse relaxation reduces the size of the resulting SDP significantly compared with that of the dense relaxation. As mentioned, increasing the dimension n and/or the degree d of polynomials in a POP makes the size of the resulting SDP larger. Furthermore, even when n and d are small, the relaxation order and/or the size of the maximum cliques can increase the size of the SDP significantly. Reducing the size of the SDP further is a key remaining issue to solve more challenging POPs.

References

[1] C. Ashcraft, D. Pierce, D. K. Wah and J. Wu, "The reference manual for SPOOLES, release 2.3: An object oriented software library for solving linear systems of equations," Boeing Shared Services Group, Seattle, WA, January 1999.

[2] J. R. S. Blair and B. Peyton, "An introduction to chordal graphs and lieque trees," In: A. George, J. R. Gilbert and J. W. H. Liu des, *Graph Theory and Sparse Matrix Computation*, Springer, New York, pp.1-29, 1993.

[3] T. F. Coleman and A. Liao, "An efficient trust region method for unconstrained discrete-time optimal control problems", *Comp.Opt.Appl.*, **4** (1995) 47–66

[4] A. R. Conn, N. I. M. Gould and P. L. Toint, "Testing a class of methods for solving minimization problems with simple bounds on the variables", *Math.Comp.*, **50** (1988) 399–430

[5] M. D. Choi, T. Y. Lam, and B. Reznick, "Sums of squares of real polynomials", *Proceedings of Symposia in Pure Mathematics*, **58** (1995) 103-126.

[6] G.Ewald, D.G.Larman and C.A.Rogers, "The directions of the line segments and of the r-dimensional balls on the boundary of a convex body in Euclidean space", *Mathematica* **17** (1970) 1-20.

[7] T. Fujie and M. Kojima, "Semidefinite programming relaxation for nonconvex quadratic programming", *Journal of Global Optimization*, **10** (1997) 367-380.

[8] M. Fukuda, M. Kojima, K. Murota and K. Nakata, "Exploiting sparsity in semidefinite programming via matrix completion I: General framework," *SIAM Journal on Optimization*, **11** (2000) 647-674.

[9] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming", *Journal of Assoc. Compute. Mach.*, **42** (1995) 1115-1145.

[10] GLOBAL Library, `http://www.gamsworld.org/global/globallib.htm`

[11] D. Henrion and J. B. Lasserre, "GloptiPoly: Global optimization over polynomials with Matlab and SeDuMi", Laboratoire d'Analyse et d'Architecture des Syst'emes, Centre National de la Recherche Scientifique, 7 Avenue du Colonel Roche, 31 077 Toulouse, cedex 4, France, February 2002.

[12] D. Henrion and J. B. Lasserre, "Detecting global optimality and extracting solutions in GloptiPoly", LAAS-CNRS Research Report No. 93541, October 2003. (To appear as a contributed chapter in the book D. Henrion, A. Garulli(Editors). Positive polynomials in control. Lecture Notes on Control and Information Sciences, Springer Verlag, to appear in 2004.)

[13] D. Henrion and J. B. Lasserre, "A convergence hierarchy of LMI relaxations for polynomial matrix inequalities", Laboratoire d'Analyse et d'Architecture des Syst'emes, Centre National de la Recherche Scientifique, 7 Avenue du Colonel Roche, 31 077 Toulouse, cedex 4, France, August 2004.

[14] D. Hilbert, "Über die Darstellung definiter Formen als Summe von Formenquadraten", em Math. Ann., **32** (1888) 342-350; see *Ges. Abh.*, **2** (1933) 154-161, Springer, Berlin, reprinted by Chelsea, New York, 1981.

[15] C.W. J. Hol and C.W. Scherer, Sumof squares relaxations for polynomial semidefinite programming, Proc. Symp. onMathematical Theory of Networks and Systems (MTNS), Leuven, Belgium, 2004.

[16] G. Karypis and V. Kumar, "METIS — A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing ordering of sparse matrices, version 4.0 —," Department of Computer Science/Army HPC Research Center, University of Minnesota, Minneapolis, MN, September, 1998.

[17] S. Kim, M. Kojima and H. Waki, "Generalized Lagrangian duals and sums of squares relaxations of sparse polynomial optimization problems", Research Report B-395, Dept. of Mathematical and computing Sciences, Tokyo Institute of Technology, Meguro, Tokyo 152-8552, September 2003. Revised July 2004. To appear in *SIAM Journal on Optimization*.

[18] M. Kojima, "Sums of squares relaxations of polynomial semidefinite programs," Research Report B-397, Dept. of Mathematical and computing Sciences, Tokyo Institute of Technology, Meguro, Tokyo 152-8552, November 2003.

[19] M. Kojima, S. Kim and H. Waki, "A general framework for convex relaxation of polynomial optimization problems over cones", *Journal of Operations Research Society of Japan*, **46** 2 (2003) 125-144.

[20] M. Kojima, S. Kim and H. Waki, "Sparsity in sums of squares of polynomials", Research Report B-391, Dept. of Mathematical and computing Sciences, Tokyo Institute of Technology, Meguro, Tokyo 152-8552, June 2003. Revised June 2004. To appear in *Mathematical Programming.*

[21] M. Kojima and M. Muramatsu, "An extension of sums of squares relaxations to polynomial optimization problems over symmetric cones," Research Report B-406, Dept. of Mathematical and computing Sciences, Tokyo Institute of Technology, Meguro, Tokyo 152-8552, April 2004.

[22] J. J. More, B. S. Garbow and K. E. Hillstrom, "Testing Unconstrained Optimization Software", *ACM Trans. Math. Soft.*, **7**, (1981) 17–41

[23] J. B. Lasserre, "Global optimization with polynomials and the problems of moments", *SIAM Journal on Optimization*, **11** (2001) 796–817.

[24] J. B. Lasserre, "An explicit equivalent positive semidefinite program for nonlinear 0-1 programs", *SIAM Journal on Optimization*, **12** (2002) 756–769.

[25] M. Laurent, "A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0-1 programming", *Mathematics of Operations Research*, **28** (2003) 470–496.

[26] M. Muramatsu and T. Suzuki, "A New Second-Order Cone Programming Relaxation for MAX-CUT Problems", *Journal of Operations Research of Japan* ,**46** (2003) 164–177.

[27] S. G. Nash, "Newton-Type Minimization via the Lanczos method", *SIAM J.Numer. Anal.*,**21** (1984) 770–788.

[28] P. A. Parrilo, "Semidefinite programming relaxations for semialgebraic problems", *Mathematical Programming*, **96** (2003) 293-320.

[29] S. Prajna, A. Papachristodoulou and P. A. Parrilo, "SOSTOOLS: Sum of Squares Optimization Toolbox for MATLAB – User's Guide", Control and Dynamical Systems, California Institute of Technology, Pasadena, CA 91125 USA, 2002.

[30] M. Putinar, "Positive polynomials on compact semi-algebraic sets", *Indiana University Mathematics Journal*, **42** (1993) 969–984.

[31] B. Reznick, "Extremal psd forms with few terms", *Duke Mathematical Journal*, **45** (1978) 363-374.

[32] M. Schweighofer, "Optimizationof polynomials on compact semialgebraic sets", Working paper, Universitat Konstanz, Fachbereich Mathematik und Statstik, 78457 Konstanz, Allemagne, Germany, April 2004.

[33] J. F. Strum, "SeDuMi 1.02, a MATLAB toolbox for optimizatin over symmetric cones", em Optimization Methods and Software, **11 & 12** (1999) 625-653.

[34] M. Yamashita, K. Fujisawa and M. Kojima, "Implementation and evaluation of SDPA 6.0 (SemiDefinite Programming Algorithm 6.0)", *Optimization Methods and Software*, **18** (2003) 491-505.