TSSOS: A MOMENT-SOS HIERARCHY THAT EXPLOITS TERM SPARSITY

JIE WANG, VICTOR MAGRON, AND JEAN-BERNARD LASSERRE

Abstract. This paper is concerned with polynomial optimization problems. We show how to exploit term (or monomial) sparsity of the input polynomials to obtain a new converging hierarchy of semidefinite programming relaxations. The novelty (and distinguishing feature) of such relaxations is to involve block-diagonal matrices obtained in an iterative procedure performing completion of the connected components of certain adjacency graphs. The graphs are related to the terms arising in the original data and *not* to the links between variables. Our theoretical framework is then applied to compute lower bounds for polynomial optimization problems either randomly generated or coming from the networked systems literature.

1. Introduction

In this paper we provide a new method to handle a certain class of *sparse* polynomial optimization problems. Roughly speaking, for problems in this class the terms (monomials) appearing in the involved polynomials satisfy a certain "sparsity pattern" which is represented by block-diagonal binary matrices. This sparsity pattern concerned with the structure of *monomials* involved in the problem, is different from the structured sparsity pattern already studied in [33] and related to the links between *variables*.

Background. The problem of minimizing a polynomial over a set defined by a finite conjunction of polynomial inequalities (also known as a *basic semialgebraic* set), is known to be NP-hard [15]. The *moment-sum of squares* (moment-SOS) hierarchy by Lasserre [12] is a nowadays established methodology allowing one to handle this problem. Optimizing a polynomial can be reformulated either with a primal infinite-dimensional linear program (LP) over probability measures or with its dual LP over nonnegative polynomials. In a nutshell, the moment-SOS hierarchy is based on the fact that one can consider a sequence of finite-dimensional primal-dual relaxations for the two above-mentioned LPs. At each step of the hierarchy, one only needs to solve a single semidefinite program (SDP). Under mild assumptions (slightly stronger than compactness), the related sequence of optimal values converges to the optimal value of the initial problem. One well-known limitation of this methodology is that the size of the matrices involved in the primal-dual SDP at the d-th step of the hierarchy is proportional to $\binom{n+d}{n}$, where n is the number of variables of the initial problem.

Date: December 20, 2019.
2010 *Mathematics Subject Classification.* Primary, 14P10,90C25; Secondary, 12D15,12Y05.
Key words and phrases. polynomial optimization, moment relaxation, sum of squares, term sparsity, moment-SOS hierarchy, semidefinite programming.

There are several existing ways to overcome these scalability limitations. To compute the SOS decomposition of a given nonnegative polynomial, one can systematically reduce the size of the corresponding SDP matrix by removing the terms (monomials) which cannot appear in the support of the decomposition [29]. One can also exploit (i) the sparsity pattern satisfied by the variables of the initial problem [13, 33] (see also the related SparsePOP solver [34]) as well as (ii) the symmetries [30] of the problem. In particular, sparsity has been successively exploited for specific applications, e.g. for solving optimal powerflow problems [9], roundoff error bound analysis [18, 19], or more recently to approximate the volume of sparse semialgebraic sets [31]. The polynomials involved in these applications have a specific *correlative sparsity pattern*. Sparse polynomial optimization is based on re-indexing the SDP matrices involved in the moment-SOS hierarchy, by considering subsets $I_1, \ldots, I_p \subseteq \{1, \ldots, n\}$ of the input variables. One obtains a sparse variant of the moment-SOS hierarchy with block SDP matrices, each block having a size related to the cardinality of these subsets. Hence if the cardinalities are small with respect to n, then the resulting SDP relaxations yield significant (sometimes drastic) computational savings. Convergence of this sparse version of the Moment-SOS hierarchy is guaranteed if the so-called *running intersection property* (RIP) holds. In particular, RIP is always satisfied when one has two subsets of variables ($p = 2$), or when $I_1, \ldots, I_p$ are the maximal cliques of the chordal extension of the correlative sparsity pattern graph of the initial problem. We refer to [33] for more details on the related algorithmic procedure to compute the cliques of the chordal extension and to [4, 24] for more details on chordal graphs. Recently, this methodology has been extended in [10] to sparse problems with non-commuting variables (for instance matrices). Other SOS-based representations include the bounded sums of squares [14] with its sparse variant [35]. These two latter hierarchies come with same convergence guarantees as the standard ones under the same sparsity pattern assumptions. They involve SDP matrices of smaller size but come with potentially larger sets of linear constraints which may sometimes result in ill-conditioned relaxations.

Other than exploiting sparsity from the perspective of variables, one can also exploit sparsity from the perspective of terms, such as sign-symmetries [17] and minimal coordinate projections [27] in the unconstrained case. More recently, *cross sparsity patterns*, a new attempt in this direction introduced in [32], applies to a wider class of polynomials. By exploiting cross sparsity patterns, a monomial basis used for constructing SOS decompositions is partitioned into blocks. If each block has a small size with respect to the size of the original monomial basis, then the corresponding SDP matrix is block-diagonal with small blocks and this might significantly improve the efficiency and the scalability.

The present paper can be viewed as a comprehensive extension of the idea in [32] *to the constrained case and in a more general perspective.*

All the above-mentioned hierarchies require to solve a sequence of SDP relaxations. However in other convex programming frameworks, there exist alternative classes of positivity certificates also based on term sparsity. This includes sums of nonnegative *circuit polynomials* (SONC) and sums of *arithmetic-geometric-exponential-means (SAGE) polynomials*. A circuit polynomial is a polynomial with support containing only monomial squares, except at most one term, whose exponent is a strict convex combination of the other exponents. An AGE polynomial is a composition of weighted sums of exponentials with linear functionals of the variables, which is nonnegative and contains also at most one negative coefficient. Existing frameworks [2, 5, 8] allow one to compute sums of nonnegative circuits and sums of AGE by relying on geometric programming and signomial programming, respectively. In [1], the authors introduce alternative decompositions of nonnegative polynomials as *diagonal sum of squares* (DSOS) and *scaled diagonal sum of squares* (SDSOS). Such decompositions can be computed via linear programming and second order cone programming, respectively, a potential advantage with respect to standard SOS-based decompositions. For these frameworks based on SAGE/SONC/SDSOS decompositions, one can also handle constrained problems and derive a corresponding converging hierarchy of lower bounds. However the underlying relaxations share the same drawback, namely their implementation and the computation of resulting lower bounds is not easy in practice.

Very recently, a combination of correlative sparsity and SDSOS has been proposed in [23]. This method does not provide a guarantee of convergence and, in its current state, is only applicable to the case of unconstrained polynomial optimization problems.

Contribution. We provide a new sparse moment-SOS hierarchy (TSSOS) based on term sparsity rather than correlative sparsity. This is in deep contrast with the sparse variant of the moment-SOS hierarchy developed in [13, 33].

• In Sec. 3, we describe an iterative procedure to exploit the term sparsity in polynomials that describe the problem on hand. Each iteration consists of two steps, a support-extension operation followed by a block-closure operation on certain binary matrices. This iterative procedure is then applied to obtain a block moment-SOS hierarchy to handle unconstrained polynomial optimization in Sec. 4 and constrained polynomial optimization in Sec. 5.

• In both cases this iterative procedure yields a converging moment-SOS hierarchy of primal-dual relaxations involving *block-diagonal* SDP matrices. In the unconstrained case we prove that the optimal value of the first SDP relaxation of the resulting hierarchy is always equal to or better than the one obtained with the SDSOS-based decompositions [1].

• We also provide in Sec. 6 a new sparse variant of Putinar's Positivstellensatz [28] for positive polynomials over basic compact semialgebraic sets. In this representation, the support of the SOS multipliers depends on the sign-symmetries binary matrix related to the support of the input data.

• In Sec. 7, we compare the efficiency and scalability of our new hierarchy with existing frameworks on randomly generated examples as well as on problems arising from the networked systems literature. The numerical results demonstrate that our approach has a significantly better performance in both efficiency and scalability. In addition, and although it is not guaranteed in theory, we observe in our numerical results that the optimal value obtained at the first step of our block hierarchy is the same as the one obtained from the dense moment-SOS hierarchy, a very encouraging sign of efficiency.

At last but not least, we emphasize that in all numerical examples (except the Broyden banded function from [33]), the usual correlative sparsity pattern is dense or almost dense and so yields no or little computational savings (or cannot even be implemented). Note that this hierarchy depends on two parameters $\hat{d}$ and k. At fixed $\hat{d}$, one can increase the value of k and after finitely many steps, only two things can happen: either (1) we retrieve a single block SDP matrix corresponding to the step $\hat{d}$ from the classical moment-SOS hierarchy, or (2) we obtain a nontrivial block SDP matrix, yielding the same value as the classical moment-SOS hierarchy but at a cheaper computational cost. Note that this hierarchy allows one more level of flexibility by playing with the two parameters k and $\hat{d}$ and still obtain a sequence of lower bounds.

2. Notations and Preliminaries

2.1. **SOS polynomials.** Let $\mathbf{x} = (x_1, \ldots, x_n)$ be a tuple of variables and $\mathbb{R}[\mathbf{x}] = \mathbb{R}[x_1, \ldots, x_n]$ be the ring of real n-variate polynomials. For a subset $\mathscr{A} \subseteq \mathbb{N}^n$, we denote by $\mathrm{conv}(\mathscr{A})$ the convex hull of $\mathscr{A}$. A polynomial $f \in \mathbb{R}[\mathbf{x}]$ can be written as $f(\mathbf{x}) = \sum_{\boldsymbol{\alpha} \in \mathscr{A}} f_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}$ with $f_{\boldsymbol{\alpha}} \in \mathbb{R}, \mathbf{x}^{\boldsymbol{\alpha}} = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. The support of f is defined by $\mathrm{supp}(f) = \{\boldsymbol{\alpha} \in \mathscr{A} \mid f_{\boldsymbol{\alpha}} \neq 0\}$, and the Newton polytope of f is defined as $\mathrm{New}(f) = \mathrm{conv}(\{\boldsymbol{\alpha} : \boldsymbol{\alpha} \in \mathrm{supp}(f)\})$. We use $|\cdot|$ to denote the cardinality of a set.

For a nonempty finite set $\mathscr{A} \subseteq \mathbb{N}^n$, let $\mathscr{P}(\mathscr{A})$ be the set of polynomials in $\mathbb{R}[\mathbf{x}]$ whose supports are contained in $\mathscr{A}$, i.e. $\mathscr{P}(\mathscr{A}) = \{f \in \mathbb{R}[\mathbf{x}] \mid \mathrm{supp}(f) \subseteq \mathscr{A}\}$ and let $\mathbf{x}^{\mathscr{A}}$ be the $|\mathscr{A}|$-dimensional column vector consisting of elements $\mathbf{x}^{\boldsymbol{\alpha}}, \boldsymbol{\alpha} \in \mathscr{A}$ (fix any ordering on $\mathbb{N}^n$). For a positive integer r, the set of $r \times r$ symmetric matrices is denoted by $\mathbb{S}^r$ and the set of $r \times r$ positive semidefinite (PSD) matrices is denoted by $\mathbb{S}_+^r$.

Given a polynomial $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$, if there exist polynomials $f_1(\mathbf{x}), \ldots, f_t(\mathbf{x})$ such that

$$f(\mathbf{x}) = \sum_{i=1}^{t} f_i(\mathbf{x})^2, \tag{1}$$

then we say that $f(\mathbf{x})$ is a *sum of squares* (SOS) polynomial. Clearly, the existence of an SOS decomposition of a given polynomial gives a certificate for its global nonnegativity. For $d \in \mathbb{N}$, let $\mathbb{N}_d^n := \{\boldsymbol{\alpha} = (\alpha_i) \in \mathbb{N}^n \mid \sum_{i=1}^n \alpha_i \leq d\}$ and assume that $f \in \mathscr{P}(\mathbb{N}_{2d}^n)$. If we choose the *standard monomial basis* $\mathbf{x}^{\mathbb{N}_d^n}$, then the SOS condition (1) is equivalent to the existence of a PSD matrix Q (which is called a *Gram matrix* [3]) such that

$$f(\mathbf{x}) = (\mathbf{x}^{\mathbb{N}_d^n})^T Q \mathbf{x}^{\mathbb{N}_d^n}. \tag{2}$$

We say that a polynomial $f \in \mathscr{P}(\mathbb{N}_{2d}^n)$ is *sparse* if the number of elements in its support $\mathscr{A} = \mathrm{supp}(f)$ is much smaller than the number of elements in $\mathbb{N}_{2d}^n$ that forms a support of fully dense polynomials in $\mathscr{P}(\mathbb{N}_{2d}^n)$. When $f(\mathbf{x})$ is a sparse polynomial in $\mathscr{P}(\mathbb{N}_{2d}^n)$, the size of the corresponding SDP problem (2) can be reduced by computing a smaller monomial basis. In fact, the set $\mathbb{N}_d^n$ in (2) can be replaced by the integer points in half of the Newton polytope of f, i.e. by

$$\mathscr{B} = \frac{1}{2} \cdot \mathrm{New}(f) \cap \mathbb{N}^n \subseteq \mathbb{N}_d^n. \tag{3}$$

See [29] for a proof. We refer to this as the *Newton polytope method*. There are also other methods to reduce the size of $\mathscr{B}$ further [11, 26]. Throughout this paper, we will fix a monomial basis, which is either the monomial basis given by the Newton polytope method in the unconstrained case or the standard monomial basis in the constrained case. For convenience, we abuse notation in the sequel and denote the monomial basis $\mathbf{x}^{\mathscr{B}}$ by $\mathscr{B}$.

2.2. **Moment matrices.** With $\mathbf{y} = (y_{\boldsymbol{\alpha}})$ being a sequence indexed by the standard monomial basis $\{\mathbf{x}^{\boldsymbol{\alpha}}\}$ of $\mathbb{R}[\mathbf{x}]$, let $L_{\mathbf{y}} : \mathbb{R}[\mathbf{x}] \to \mathbb{R}$ be the linear functional

$$f = \sum_{\boldsymbol{\alpha}} f_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} \mapsto L_{\mathbf{y}}(f) = \sum_{\boldsymbol{\alpha}} f_{\boldsymbol{\alpha}} y_{\boldsymbol{\alpha}}.$$

Fixing a monomial basis $\mathscr{B}$, the *moment* matrix $M_{\mathscr{B}}(\mathbf{y})$ associated with $\mathscr{B}$ and $\mathbf{y}$ is the matrix with rows and columns indexed by $\mathscr{B}$ such that

$$M_{\mathscr{B}}(\mathbf{y})_{\boldsymbol{\beta}\boldsymbol{\gamma}} := L_{\mathbf{y}}(\mathbf{x}^{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\gamma}}) = y_{\boldsymbol{\beta}+\boldsymbol{\gamma}}, \quad \forall \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathscr{B}.$$

If $\mathscr{B}$ is the standard monomial basis $\mathbb{N}^n_d$, we also denote $M_{\mathscr{B}}(\mathbf{y})$ by $M_d(\mathbf{y})$.

Suppose $g = \sum_{\boldsymbol{\alpha}} g_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} \in \mathbb{R}[\mathbf{x}]$ and let $\mathbf{y} = (y_{\boldsymbol{\alpha}})$ be given. For a positive integer d, the *localizing* matrix $M_d(g\mathbf{y})$ associated with g and $\mathbf{y}$ is the matrix with rows and columns indexed by $\mathbb{N}^n_d$ such that

$$M_d(g\,\mathbf{y})_{\boldsymbol{\beta}\boldsymbol{\gamma}} := L_{\mathbf{y}}(g\,\mathbf{x}^{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\gamma}}) = \sum_{\boldsymbol{\alpha}} g_{\boldsymbol{\alpha}} y_{\boldsymbol{\alpha}+\boldsymbol{\beta}+\boldsymbol{\gamma}}, \quad \forall \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{N}^n_d.$$

3. Exploiting term sparsity in SOS decompositions

For a positive integer r, let $[r] := \{1, \ldots, r\}$. For matrices $A, B \in \mathbb{S}^r$, let $A \circ B \in \mathbb{S}^r$ denote the Hadamard, or entrywise, product of A and B, defined by the equation $[A \circ B]_{ij} = A_{ij}B_{ij}$ and let $\langle A, B\rangle \in \mathbb{R}$ be the trace inner-product, defined by $\langle A, B\rangle = \mathrm{Tr}(A^T B)$. Let $\{0,1\}^{r\times r}$ be the set of $r \times r$ binary matrices. The support of a binary matrix $B \in \mathbb{S}^r \cap \{0,1\}^{r\times r}$ is the set of locations of nonzero entries, i.e.,

$$\mathrm{supp}(B) := \{(i,j) \in [r] \times [r] \mid B_{ij} = 1\}.$$

For a binary matrix $B \in \mathbb{S}^r \cap \{0,1\}^{r\times r}$, we define the set of PSD matrices with sparsity pattern represented by B as

$$\mathbb{S}^r_+(B) := \{Q \in \mathbb{S}^r_+ \mid B \circ Q = Q\}.$$

For a polynomial $f(\mathbf{x}) = \sum_{\boldsymbol{\alpha}\in\mathscr{A}} f_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}$ with $\mathrm{supp}(f) = \mathscr{A}$, fix a monomial basis $\mathscr{B} = \{\boldsymbol{\omega}_i\}_{i=1}^r$. Let $\mathscr{B} + \mathscr{B} := \{\boldsymbol{\beta} + \boldsymbol{\gamma} \mid \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathscr{B}\}$. For any $\boldsymbol{\alpha} \in \mathscr{B} + \mathscr{B}$, associate it with a binary matrix $A_{\boldsymbol{\alpha}} \in \mathbb{S}^r \cap \{0,1\}^{r\times r}$ such that $[A_{\boldsymbol{\alpha}}]_{ij} = 1$ iff $\boldsymbol{\omega}_i + \boldsymbol{\omega}_j = \boldsymbol{\alpha}$ for all i, j. Then $f(\mathbf{x})$ is an SOS polynomial iff there exists $Q \in \mathbb{S}^r_+$ such that the following coefficient matching condition holds:

$$\langle A_{\boldsymbol{\alpha}}, Q\rangle = f_{\boldsymbol{\alpha}} \text{ for all } \boldsymbol{\alpha} \in \mathscr{B} + \mathscr{B}, \tag{4}$$

where we set $f_{\boldsymbol{\alpha}} = 0$ if $\boldsymbol{\alpha} \notin \mathscr{A}$.

For convenience, we define a block-closure operation on binary matrices as follows.

Definition 3.1. *For a binary matrix $B \in \mathbb{S}^r \cap \{0,1\}^{r\times r}$, let $R \subseteq [r] \times [r]$ be the adjacency relation of B, i.e., $(i,j) \in R$ iff $B_{ij} = 1$. Denote the transitive closure of R by $\overline{R}$. Then define $\overline{B} \in \mathbb{S}^r \cap \{0,1\}^{r\times r}$ as*

$$\overline{B}_{ij} := \begin{cases} 1, & (i,j) \in \overline{R}, \\ 0, & \textit{otherwise}. \end{cases}$$

For a binary matrix $B \in \mathbb{S}^r \cap \{0,1\}^{r\times r}$, the evaluation of $\overline{B}$ has a graphical description if $B_{ii} = 1$ for all i. Suppose that G is the adjacency graph of B. Then $\overline{B}$ is the adjacency matrix of the graph obtained by completing the connected components of G to complete subgraphs. Note also that $\overline{B}$ is block-diagonal up to permutation, where each block corresponds to a connected component of G. Figure 1 is a simple example where $\overline{B}$ has two blocks of size 3 and 1 corresponding to the connected components of G: $\{1,3,4\}$ and $\{2\}$, respectively.

$$B = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \qquad \overline{B} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix}$$

G :

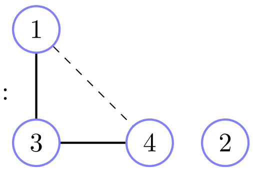

FIGURE 1. Block-diagonalizations and connected components

Remark 3.2. *The block-closure operation $\overline{B}$ used in this paper can be actually replaced by a chordal-extension operation on graphs. Then take maximal cliques rather than connected components. See* [32] *for more details. We use the block-closure in this paper since it is very simple to determine.*

Let $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ with $\operatorname{supp}(f) = \mathscr{A}$ and let $\mathscr{B}$ be a monomial basis with $r = |\mathscr{B}|$. Let $\mathscr{S}^{(0)} = \mathscr{A} \cup (2\mathscr{B})$ where $2\mathscr{B} = \{2\boldsymbol{\beta} \mid \boldsymbol{\beta} \in \mathscr{B}\}$. For $k \geq 1$, we recursively define binary matrices $C_{\mathscr{A}}^{(k)}, B_{\mathscr{A}}^{(k)} \in \mathbb{S}^r \cap \{0,1\}^{r \times r}$ indexed by $\mathscr{B}$ as follows:

$$[C_{\mathscr{A}}^{(k)}]_{\boldsymbol{\beta}\boldsymbol{\gamma}} := \begin{cases} 1, & \text{if } \boldsymbol{\beta} + \boldsymbol{\gamma} \in \mathscr{S}^{(k-1)}, \\ 0, & \text{otherwise}, \end{cases}$$

and $B_{\mathscr{A}}^{(k)} = \overline{C_{\mathscr{A}}^{(k)}}$, with

$$\mathscr{S}^{(k)} := \bigcup_{[B_{\mathscr{A}}^{(k)}]_{\boldsymbol{\beta}\boldsymbol{\gamma}}=1} \{\boldsymbol{\beta} + \boldsymbol{\gamma}\}.$$

By construction, it is easy to see that $\operatorname{supp}(B_{\mathscr{A}}^{(k)}) \subseteq \operatorname{supp}(B_{\mathscr{A}}^{(k+1)})$ for all $k \geq 1$. Hence $\{B_{\mathscr{A}}^{(k)}\}_{k\geq 1}$ stabilizes after a finite number of steps. We denote the stabilized matrix by $B_{\mathscr{A}}^{(*)}$.

Let us denote the set of SOS polynomials supported on $\mathscr{A}$ by

$$\Sigma(\mathscr{A}) := \{f \in \mathscr{P}(\mathscr{A}) \mid \exists Q \in \mathbb{S}_+^r \text{ s.t. } f = (\mathbf{x}^{\mathscr{B}})^T Q \mathbf{x}^{\mathscr{B}}\},$$

and for $k \geq 1$, let $\Sigma_k(\mathscr{A})$ be the subset of $\Sigma(\mathscr{A})$ whose member admits a Gram matrix with sparsity pattern represented by $B_{\mathscr{A}}^{(k)}$, i.e.

$$\Sigma_k(\mathscr{A}) := \{f \in \mathscr{P}(\mathscr{A}) \mid \exists Q \in \mathbb{S}_+^r(B_{\mathscr{A}}^{(k)}) \text{ s.t. } f = (\mathbf{x}^{\mathscr{B}})^T Q \mathbf{x}^{\mathscr{B}}\}. \tag{5}$$

In addition, let

$$\Sigma_*(\mathscr{A}) := \{f \in \mathscr{P}(\mathscr{A}) \mid \exists Q \in \mathbb{S}_+^r(B_{\mathscr{A}}^{(*)}) \text{ s.t. } f = (\mathbf{x}^{\mathscr{B}})^T Q \mathbf{x}^{\mathscr{B}}\}. \tag{6}$$

By construction, we have the following inclusions:

$$\Sigma_1(\mathscr{A}) \subseteq \Sigma_2(\mathscr{A}) \subseteq \cdots \subseteq \Sigma_*(\mathscr{A}) \subseteq \Sigma(\mathscr{A}).$$

Theorem 3.3. *For a finite set $\mathscr{A} \subseteq \mathbb{N}^n$, $\Sigma_*(\mathscr{A}) = \Sigma(\mathscr{A})$.*

Proof. We only need to prove the inclusion $\Sigma(\mathscr{A}) \subseteq \Sigma_*(\mathscr{A})$. Suppose $\mathscr{B}$ is a monomial basis. For any $f \in \Sigma(\mathscr{A})$, let $Q \in \mathbb{S}_+^r$ be a Gram matrix of f. We construct a Gram matrix $\tilde{Q}$ for f by $\tilde{Q} = B_{\mathscr{A}}^{(*)} \circ Q$. It is easy to check that $f = (\mathbf{x}^{\mathscr{B}})^T \tilde{Q} \mathbf{x}^{\mathscr{B}}$. Note that $\tilde{Q}$ is block-diagonal (up to permutation) and each block of $\tilde{Q}$ is a primary submatrix of Q, so $\tilde{Q}$ is PSD. Thus $f \in \Sigma_*(\mathscr{A})$. ☐

Consequently, we obtain a hierarchy of inner approximations of $\Sigma(\mathscr{A})$ which reaches $\Sigma(\mathscr{A})$ in a finite number of steps.

Remark 3.4. *For each $k \geq 1$, $Q \in \mathbb{S}_+^r(B_{\mathscr{A}}^{(k)})$ is block-diagonal (up to permutation), thus checking membership in $\Sigma_k(\mathscr{A})$ boils down to solving an SDP problem involving SDP matrices of small sizes if each block has a small size with respect to the original matrix. This might significantly reduce the overall computational cost.*

The next result states that $\Sigma_1(\mathscr{A}) = \Sigma(\mathscr{A})$ always holds in the quadratic case.

Theorem 3.5. *For a finite set $\mathscr{A} \subseteq \mathbb{N}^n$, if for all $\boldsymbol{\alpha} = (\alpha_i) \in \mathscr{A}$, $\sum_{i=1}^n \alpha_i \leq 2$, then $\Sigma_1(\mathscr{A}) = \Sigma(\mathscr{A})$.*

Proof. We only need to prove the inclusion $\Sigma(\mathscr{A}) \subseteq \Sigma_1(\mathscr{A})$. Suppose $f \in \Sigma(\mathscr{A})$ is a quadratic polynomial with $\mathrm{supp}(f) = \mathscr{A}$. Let $Q = [q_{ij}]_{i,j=0}^n$ be a PSD Gram matrix of f. To show $f \in \Sigma_1(\mathscr{A})$, we only need to prove that $Q \in \mathbb{S}_+^{n+1}(C_{\mathscr{A}}^{(1)})$, which holds if $[C_{\mathscr{A}}^{(1)}]_{ij} = 0$ implies $q_{ij} = 0$ for all i, j. Clearly, $[C_{\mathscr{A}}^{(1)}]_{00} = 0$ implies $q_{00} = 0$. Let $\{\mathbf{e}_k\}_{k=1}^n$ be the standard basis. If $i = 0, j > 0$, from $[C_{\mathscr{A}}^{(1)}]_{0j} = 0$ we have $\mathbf{e}_j \notin \mathscr{A}$. If $i > 0, j = 0$, from $[C_{\mathscr{A}}^{(1)}]_{i0} = 0$ we have $\mathbf{e}_i \notin \mathscr{A}$. If $i, j > 0$, from $[C_{\mathscr{A}}^{(1)}]_{ij} = 0$ we have $\mathbf{e}_i + \mathbf{e}_j \notin \mathscr{A}$. In any of these three cases, we have $q_{ij} = 0$ as desired. ☐

4. A block SDP hierarchy for unconstrained POPs

In this section, we consider the unconstrained polynomial optimization problem:

$$(\mathrm{P}): \quad \theta^* := \inf_{\mathbf{x}} \{f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\}$$

with $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$, and exploit the sparse SOS decompositions in Sec. 3 to establish a block SDP hierarchy for (P).

Obviously, (P) is equivalent to

$$(\mathrm{P'}): \quad \theta^* = \sup_{\lambda} \{\lambda \mid f(\mathbf{x}) - \lambda \geq 0\}.$$

Replacing the nonnegativity condition by the stronger SOS condition, we obtain an SOS relaxation of (P):

$$(\mathrm{SOS}): \quad \theta_{sos} := \sup_{\lambda} \{\lambda \mid f(\mathbf{x}) - \lambda \in \Sigma(\mathscr{A})\},$$

with $\mathscr{A} = \{\mathbf{0}\} \cup \mathrm{supp}(f)$. If f is sparse and we replace the nonnegativity condition in (P') by the sparse SOS conditions (5), then we obtain a hierarchy of sparse SOS relaxations of (P):

$$(\mathrm{P}^k)^*: \quad \theta_k := \sup_{\lambda} \{\lambda \mid f(\mathbf{x}) - \lambda \in \Sigma_k(\mathscr{A})\}, \quad k = 1, 2, \ldots. \tag{7}$$

For each k, $(\mathrm{P}^k)^*$ corresponds to a block SDP problem. In addition, let

$$(\mathrm{TSSOS}): \quad \theta_{tssos} := \sup_{\lambda} \{\lambda \mid f(\mathbf{x}) - \lambda \in \Sigma_*(\mathscr{A})\}. \tag{8}$$

Then we have the following hierarchy of lower bounds for the optimum of (P):

$$\theta^* \geq \theta_{sos} = \theta_{tssos} \geq \cdots \geq \theta_2 \geq \theta_1,$$

where the equality $\theta_{sos} = \theta_{tssos}$ follows from Theorem 3.3.

Let $\mathscr{B}$ be a monomial basis. For $k \geq 1$, the dual of $(\mathrm{P}^k)^*$ is the following moment problem

$$(\mathrm{P}^k): \quad \begin{cases} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & B_{\mathscr{A}}^{(k)} \circ M_{\mathscr{B}}(\mathbf{y}) \succeq 0, \\ & y_{\mathbf{0}} = 1. \end{cases} \tag{9}$$

Proposition 4.1. *Let $f \in \mathbb{R}[\mathbf{x}]$ with* $\mathrm{supp}(f) = \mathscr{A}$*. For each $k \geq 1$, there is no duality gap between (P^k) and $(P^k)^*$.*

Proof. This easily follows from Proposition 3.1 of [12] for the dense case and the observation that each block of $B_{\mathscr{A}}^{(k)} \circ M_{\mathscr{B}}(\mathbf{y})$ is a primal submatrix of $M_{\mathscr{B}}(\mathbf{y})$. ☐

Example 4.2. *Consider the polynomial $f = 1 + x_1^4 + x_2^4 + x_3^4 + x_1x_2x_3 + x_2$. A monomial basis for f is $\{1, x_2, x_1^2, x_2^2, x_1x_3, x_3^2, x_1, x_2x_3, x_3, x_1x_2\}$. Then*

$$C_{\mathscr{A}}^{(1)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

and this yields

$$B_{\mathscr{A}}^{(1)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Furthermore, we have

$$B_{\mathscr{A}}^{(2)} = C_{\mathscr{A}}^{(2)} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}.$$

Thus $\{B_{\mathscr{A}}^{(k)}\}_{k\geq 1}$ *stabilizes at* $k=2$*. Then solve the SDPs* (P^1), (P^2) *and we obtain* $\theta_1 = \theta_2 = \theta_{tssos} = \theta_{sos} = \theta^* \approx 0.4753$*.*

4.1. **Relationship with DSOS/SDSOS optimization.** The following definitions of DSOS and SDSOS have been introduced in [1]. For more details the interested reader is referred to [1].

A symmetric matrix $Q \in \mathbb{S}^r$ is *diagonally dominant* if $Q_{ii} \geq \sum_{j=1}^r |Q_{ij}|$ for $i = 1, \ldots, r$. We say that a polynomial $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ is a *diagonally dominant sum of squares* (DSOS) polynomial if it admits a Gram matrix representation (2) with a diagonally dominant Gram matrix Q. We denote the set of DSOS polynomials by $DSOS$.

A symmetric matrix $Q \in \mathbb{S}^r$ is *scaled diagonally dominant* if there exists a positive definite $r \times r$ diagonal matrix D such that DAD is diagonally dominant. We say that a polynomial $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ is a *scaled diagonally dominant sum of squares* (SDSOS) polynomial if it admits a Gram matrix representation (2) with a scaled diagonally dominant Gram matrix Q. We denote the set of SDSOS polynomials by $SDSOS$.

For a finite set $\mathscr{A} \subseteq \mathbb{N}^n$, let

$$\mathrm{DSOS}(\mathscr{A}) := \Sigma(\mathscr{A}) \cap DSOS$$

and

$$\mathrm{SDSOS}(\mathscr{A}) := \Sigma(\mathscr{A}) \cap SDSOS.$$

Clearly, we have $\mathrm{DSOS}(\mathscr{A}) \subseteq \mathrm{SDSOS}(\mathscr{A}) \subseteq \Sigma(\mathscr{A})$.

Theorem 4.3. *For a finite set* $\mathscr{A} \subseteq \mathbb{N}^n$, $\mathrm{SDSOS}(\mathscr{A}) \subseteq \Sigma_1(\mathscr{A})$.

Proof. Let $\mathscr{B}$ be a monomial basis with $r = |\mathscr{B}|$. For any $f \in \mathrm{SDSOS}(\mathscr{A})$, there exists a scaled diagonally dominant Gram matrix $Q \in \mathbb{S}_+^r$ indexed by $\mathscr{B}$. We then construct a Gram matrix $\tilde{Q}$ by

$$\tilde{Q}_{\boldsymbol{\beta\gamma}} = \begin{cases} 0, & \text{if } \boldsymbol{\beta} + \boldsymbol{\gamma} \notin \mathscr{A} \cup 2\mathscr{B}, \\ Q_{\boldsymbol{\beta\gamma}}, & \text{otherwise.} \end{cases}$$

It is easy to see that $f = (\mathbf{x}^{\mathscr{B}})^T \tilde{Q} \mathbf{x}^{\mathscr{B}}$. Note that we only replace off-diagonal entries by zeros in Q and replacing off-diagonal entries by zeros does not affect the scaled diagonal dominance of a matrix. Hence $\tilde{Q}$ is also a scaled diagonally dominant matrix. Moreover, we have $B_{\mathscr{A}}^{(1)} \circ \tilde{Q} = \tilde{Q}$ by construction. Thus $f \in \Sigma_1(\mathscr{A})$. $\square$

Replacing the nonnegativity condition in (P') by the DSOS (resp. SDSOS) condition, we obtain the DSOS (resp. SDSOS) relaxation of (P):

$$\text{(DSOS)}: \quad \theta_{dsos} := \sup_{\lambda}\{\lambda \mid f(\mathbf{x}) - \lambda \in \text{DSOS}(\mathscr{A})\}$$

and

$$\text{(SDSOS)}: \quad \theta_{sdsos} := \sup_{\lambda}\{\lambda \mid f(\mathbf{x}) - \lambda \in \text{SDSOS}(\mathscr{A})\}.$$

The above DSOS and SDSOS relaxations for polynomial optimization have been introduced and studied in [1]. By Theorem 4.3, we have the following hierarchy of lower bounds for the optimal value of (P):

$$\theta^* \geq \theta_{sos} = \theta_{tssos} \geq \cdots \geq \theta_2 \geq \theta_1 \geq \theta_{sdsos} \geq \theta_{dsos}.$$

5. A block Moment-SOS hierarchy for constrained POPs

In this section, we consider the constrained polynomial optimization problem:

$$\text{(Q)}: \quad \theta^* := \inf_{\mathbf{x}}\{f(\mathbf{x}) : \mathbf{x} \in \mathbf{K}\}$$

where $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ is a polynomial and $\mathbf{K} \subseteq \mathbb{R}^n$ is the basic semialgebraic set

$$\mathbf{K} = \{\mathbf{x} \in \mathbb{R}^n : g_j(\mathbf{x}) \geq 0, j = 1, \ldots, m\}, \tag{10}$$

for some polynomials $g_j(\mathbf{x}) \in \mathbb{R}[\mathbf{x}], j = 1, \ldots, m$.

Let $d_j = \lceil \deg(g_j)/2 \rceil, j = 1, \ldots, m$ and let $\hat{d} \geq \max\{\lceil \deg(f)/2 \rceil, d_1, \ldots, d_m\}$ be a positive integer. Then the Lasserre's hierarchy [12] of moment semidefinite relaxations of (Q) is defined by:

$$(\mathrm{Q}_{\hat{d}}): \quad \begin{cases} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & M_{\hat{d}}(\mathbf{y}) \succeq 0, \\ & M_{\hat{d}-d_j}(g_j\mathbf{y}) \succeq 0, \quad j = 1, \ldots, m, \\ & y_{\mathbf{0}} = 1, \end{cases} \tag{11}$$

with optimal value denoted by $\theta_{\hat{d}}$. Let $d_0 = 0$ and $\mathbb{N}^n_{2(\hat{d}-d_j)}$ be the standard monomial basis for $j = 0, \ldots, m$. The dual of (11) is an SDP equivalent to the following SOS problem:

$$(\mathrm{Q}_{\hat{d}})^*: \quad \begin{cases} \sup & \lambda \\ \text{s.t.} & f - \lambda = s_0 + \sum_{j=1}^m s_j g_j, \\ & s_j \in \Sigma(\mathbb{N}^n_{2(\hat{d}-d_j)}), \quad j = 0, \ldots, m. \end{cases} \tag{12}$$

Let

$$\mathscr{A} = \text{supp}(f) \cup \bigcup_{j=1}^m \text{supp}(g_j).$$

Set $\mathscr{S}^{(0)}_{0,\hat{d}} = \mathscr{A} \cup (2\mathbb{N})^n$ and $\mathscr{S}^{(0)}_{j,\hat{d}} = \emptyset, j = 1, \ldots, m$. Let us define $r_j := \binom{n+\hat{d}-d_j}{\hat{d}-d_j}$. For $k \geq 1$, we recursively define binary matrices $C^{(k)}_{j,\hat{d}}, B^{(k)}_{j,\hat{d}} \in \mathbb{S}^{r_j} \cap \{0,1\}^{r_j \times r_j}$, indexed by $\mathbb{N}^n_{\hat{d}-d_j}$, $j = 0, \ldots, m$, as follows:

$$[C^{(k)}_{0,\hat{d}}]_{\boldsymbol{\beta}\boldsymbol{\gamma}} := \begin{cases} 1, & \text{if } \boldsymbol{\beta} + \boldsymbol{\gamma} \in \bigcup_{j=0}^m \mathscr{S}^{(k-1)}_{j,\hat{d}}, \\ 0, & \text{otherwise}, \end{cases} \tag{13}$$

$$[C_{j,\hat{d}}^{(k)}]_{\boldsymbol{\beta\gamma}} := \begin{cases} 1, & \text{if } (\mathrm{supp}(g_j)+\boldsymbol{\beta}+\boldsymbol{\gamma}) \cap \bigcup_{j=0}^m \mathscr{S}_{j,\hat{d}}^{(k-1)} \neq \emptyset, \\ 0, & \text{otherwise,} \end{cases} \quad (1 \le j \le m) \tag{14}$$

and $B_{j,\hat{d}}^{(k)} = \overline{C_{j,\hat{d}}^{(k)}}$ $(0 \le j \le m)$, with

$$\mathscr{S}_{0,\hat{d}}^{(k)} := \bigcup_{[B_{0,\hat{d}}^{(k)}]_{\boldsymbol{\beta\gamma}}=1} \{\boldsymbol{\beta}+\boldsymbol{\gamma}\}, \tag{15}$$

and

$$\mathscr{S}_{j,\hat{d}}^{(k)} := \mathrm{supp}(g_j) + \bigcup_{[B_{j,\hat{d}}^{(k)}]_{\boldsymbol{\beta\gamma}}=1} \{\boldsymbol{\beta}+\boldsymbol{\gamma}\}, \quad j=1,\ldots,m. \tag{16}$$

Therefore we can further consider a block hierarchy of relaxations of $(\mathrm{Q}_{\hat{d}})$ for $k \ge 1$:

$$(\mathrm{Q}_{\hat{d}}^k): \quad \begin{cases} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & B_{0,\hat{d}}^{(k)} \circ M_{\hat{d}}(\mathbf{y}) \succeq 0, \\ & B_{j,\hat{d}}^{(k)} \circ M_{\hat{d}-d_j}(g_j\mathbf{y}) \succeq 0, \quad j=1,\ldots,m, \\ & y_{\mathbf{0}} = 1, \end{cases} \tag{17}$$

with optimal value denoted by $\theta_{\hat{d}}^{(k)}$. By construction, we have $\mathrm{supp}(B_{j,\hat{d}}^{(k)}) \subseteq \mathrm{supp}(B_{j,\hat{d}}^{(k+1)})$ for all $k \ge 1$ and $j=0,\ldots,m$. Hence $\{B_{j,\hat{d}}^{(k)}\}_{k\ge1}$ stabilizes for all j after a finite number of steps. We denote the stabilized matrices by $\{B_{j,\hat{d}}^{(*)}\}_j$ and denote the corresponding SDP problem from (17) by $(\mathrm{Q}_{\hat{d}}^*)$ with optimum $\theta_{\hat{d}}^*$.

Theorem 5.1. *The sequence* $\{\theta_{\hat{d}}^{(k)}\}_{k\ge1}$ *of optimal values of* (17) *is monotone nondecreasing and in addition,* $\theta_{\hat{d}}^* = \theta_{\hat{d}}$.

Proof. Since $\mathrm{supp}(B_{j,\hat{d}}^{(k)}) \subseteq \mathrm{supp}(B_{j,\hat{d}}^{(k+1)})$ and $B_{j,\hat{d}}^{(k)}$ is block-diagonal (up to permutation) for all j,k, $(\mathrm{Q}_{\hat{d}}^k)$ is a relaxation of $(\mathrm{Q}_{\hat{d}}^{k+1})$ and $(\mathrm{Q}_{\hat{d}})$. Therefore $\{\theta_{\hat{d}}^{(k)}\}_{k\ge1}$ is nondecreasing and $\theta_{\hat{d}}^* \le \theta_{\hat{d}}$. Suppose that $\mathbf{y}$ is an optimal solution of $(\mathrm{Q}_{\hat{d}}^*)$. Then define $\overline{\mathbf{y}}$ as follows:

$$\overline{\mathbf{y}}_{\boldsymbol{\alpha}} = \begin{cases} \mathbf{y}_{\boldsymbol{\alpha}}, & \text{if } \mathbf{y}_{\boldsymbol{\alpha}} \text{ appears in } (\mathrm{Q}_{\hat{d}}^*), \\ 0, & \text{otherwise.} \end{cases}$$

By construction, we have $M_{\hat{d}}(\overline{\mathbf{y}}) = B_{0,\hat{d}}^{(*)} \circ M_{\hat{d}}(\mathbf{y})$ and $M_{\hat{d}-d_j}(g_j\overline{\mathbf{y}}) = B_{j,\hat{d}}^{(*)} \circ M_{\hat{d}-d_j}(g_j\mathbf{y})$, $j=1,\ldots,m$. Therefore $\overline{\mathbf{y}}$ is a feasible solution of $(\mathrm{Q}_{\hat{d}})$ and hence $\theta_{\hat{d}}^* \ge \theta_{\hat{d}}$. □

Let us note $g_0 := 1$ and $d_0 := 0$. For each $j=1,\ldots,m$, writing $M_{\hat{d}-d_j}(g_j\mathbf{y}) = \sum_{\boldsymbol{\alpha}} D_{\boldsymbol{\alpha}}^j y_{\boldsymbol{\alpha}}$ for appropriate symmetric matrices $\{D_{\boldsymbol{\alpha}}^j\}$, then we can write the dual of $(\mathrm{Q}_{\hat{d}}^k)$ as

$$(\mathrm{Q}_{\hat{d}}^k)^*: \quad \begin{cases} \sup & \lambda \\ \text{s.t.} & \langle Q_0, A_{\boldsymbol{\alpha}}\rangle + \sum_{j=1}^m \langle Q_j, D_{\boldsymbol{\alpha}}^j\rangle + \lambda\delta_{\mathbf{0}\boldsymbol{\alpha}} = f_{\boldsymbol{\alpha}}, \forall \boldsymbol{\alpha} \in \mathbb{N}_{2\hat{d}}^n, \\ & Q_j \in \mathbb{S}_+^{r_j}(B_{j,\hat{d}}^{(k)}), \quad j=0,\ldots,m, \end{cases} \tag{18}$$

where $A_{\boldsymbol{\alpha}}$ is defined in Sec. 3 and $\delta_{\mathbf{0}\boldsymbol{\alpha}}$ is the usual Kronecker symbol.

For a family of polynomials $\mathbf{g} = (g_1, \ldots, g_m) \subseteq \mathbb{R}[\mathbf{x}]$, the associated *quadratic module* $\mathcal{Q}(\mathbf{g}) = \mathcal{Q}(g_1, \ldots, g_m) \subseteq \mathbb{R}[\mathbf{x}]$ is defined by

$$\mathcal{Q}(\mathbf{g}) := \{s_0 + \sum_{j=1}^{m} s_j g_j \mid s_j \text{ is an SOS, } j = 0, \ldots, m\}. \tag{19}$$

The quadratic module $\mathcal{Q}(\mathbf{g})$ associated with $\mathbf{K}$ in (10) is said to be *Archimedean* if there exists $N > 0$ such that the quadratic polynomial $\mathbf{x} \mapsto N - \|\mathbf{x}\|^2$ belongs to $\mathcal{Q}(\mathbf{g})$.

Proposition 5.2. *Let $f \in \mathbb{R}[\mathbf{x}]$ and $\mathbf{K}$ be as in (10). Assume that the quadratic module $\mathcal{Q}(\mathbf{g})$ is Archimedean and $\mathbf{K}$ has a nonempty interior. Then for $\hat{d}$ sufficiently large, there is no duality gap between $(Q^k_{\hat{d}})$ and $(Q^k_{\hat{d}})^*$.*

Proof. By the duality theory of convex programming, this easily follows from Theorem 4.2 of [12] for the dense case and the observation that each block of $B^{(k)}_{j,\hat{d}} \circ M_{\hat{d}-d_j}(g_j\mathbf{y})$ is a primal submatrix of $M_{\hat{d}-d_j}(g_j\mathbf{y})$ for all j, k. $\square$

For any feasible solution of $(\mathrm{Q}^k_{\hat{d}})^*$, multiplying each side of the constraint in (18) by $\mathbf{x}^{\boldsymbol{\alpha}}$ for all $\boldsymbol{\alpha} \in \mathbb{N}^n_{2\hat{d}}$ and summing up yields

$$\langle Q_0, \sum_{\boldsymbol{\alpha} \in \mathbb{N}^n_{2\hat{d}}} A_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} \rangle + \sum_{j=1}^{m} \langle Q_j, \sum_{\boldsymbol{\alpha} \in \mathbb{N}^n_{2\hat{d}}} D^j_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} \rangle = f - \lambda. \tag{20}$$

Note that $\sum_{\boldsymbol{\alpha} \in \mathbb{N}^n_{2\hat{d}}} A_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} = \mathbf{x}^{\mathbb{N}^n_{\hat{d}}} \cdot (\mathbf{x}^{\mathbb{N}^n_{\hat{d}}})^T$ and $\sum_{\boldsymbol{\alpha} \in \mathbb{N}^n_{2\hat{d}}} D^j_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} = g_j \mathbf{x}^{\mathbb{N}^n_{\hat{d}-d_j}} \cdot (\mathbf{x}^{\mathbb{N}^n_{\hat{d}-d_j}})^T$ for $j = 1 \ldots, m$. Hence we can rewrite (20) as

$$(\mathbf{x}^{\mathbb{N}^n_{\hat{d}}})^T Q_0 \mathbf{x}^{\mathbb{N}^n_{\hat{d}}} + \sum_{j=1}^{m} g_j (\mathbf{x}^{\mathbb{N}^n_{\hat{d}-d_j}})^T Q_j \mathbf{x}^{\mathbb{N}^n_{\hat{d}-d_j}} = f - \lambda. \tag{21}$$

For each j, the binary matrix $B^{(k)}_{j,\hat{d}}$ is block-diagonal up to permutation and $B^{(k)}_{j,\hat{d}}$ induces a partition of the monomial basis $\mathbb{N}^n_{\hat{d}-d_j}$: two vectors $\boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{N}^n_{\hat{d}-d_j}$ belong to the same block if and only if the rows and columns indexed by $\boldsymbol{\beta}, \boldsymbol{\gamma}$ belong to the same block in $B^{(k)}_{j,\hat{d}}$. If some diagonal element of $B^{(k)}_{j,\hat{d}}$ is zero, then the corresponding basis element can be discarded. Assume that $v_{j1}(\mathbf{x}), \ldots, v_{jl_j}(\mathbf{x})$ are the resulting blocks in this partition and $Q_{j1}, \ldots, Q_{jl_j}$ are the corresponding submatrices of Q_j. Then (21) reads as

$$\sum_{i=1}^{l_0} v_{ji}(\mathbf{x})^T Q_{ji} v_{ji}(\mathbf{x}) + \sum_{j=1}^{m} g_j \sum_{i=1}^{l_j} v_{ji}(\mathbf{x})^T Q_{ji} v_{ji}(\mathbf{x}) = f - \lambda. \tag{22}$$

For all i, j, the polynomial $s_{ji} := v_{ji}(\mathbf{x})^T Q_{ji} v_{ji}(\mathbf{x})$ is an SOS polynomial since Q_{ji} is PSD. Then we have

$$\sum_{i=1}^{l_0} s_{ji} + \sum_{j=1}^{m} g_j \sum_{i=1}^{l_j} s_{ji} = f - \lambda. \tag{23}$$

Notice that (23) is in fact a *sparse* Putinar's representation for the polynomial $f - \lambda$. This representation is a certificate of positivity on $\mathbf{K}$ for the polynomial $f - \lambda$. Indeed (23) ensures that $f - \lambda$ is nonnegative on $\mathbf{K}$ and each SOS s_{ji} has an associated Gram matrix Q_{ji} indexed in the sparse monomial basis $v_{ji}(\mathbf{x})$.

Example 5.3. *Let* $f = x_1^4 + x_2^4 - x_1x_2$ *and* $\mathbf{K} = \{(x_1, x_2) \in \mathbb{R}^2 : g_1 = 1 - 2x_1^2 - x_2^2 \geq 0\}$. *Let* $\mathscr{A} = \{(4, 0), (0, 4), (1, 1), (0, 0), (2, 0), (0, 2)\}$ *and* $\hat{d} = 2$. *Take* $\{1, x_1, x_2, x_1^2, x_1x_2, x_2^2\}$ *as a monomial basis. Then*

$$C_{0,2}^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad C_{1,2}^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

This yields

$$B_{0,2}^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad B_{1,2}^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

Furthermore, we have $B_{j,2}^{(2)} = C_{j,2}^{(1)} = B_{j,2}^{(1)}, j = 1, 2$. *Thus* $\{B_{0,2}^{(k)}, B_{1,2}^{(k)}\}_{k\geq 1}$ *stabilizes at* $k = 1$ *and* (Q_2^1) *can be read as*

$$(Q_2^1): \quad \begin{cases} \inf & y_{40} + y_{04} - y_{11} \\ \text{s.t.} & \begin{bmatrix} y_{00} & & & y_{20} & y_{11} & y_{02} \\ & y_{20} & y_{11} & & & \\ & y_{11} & y_{02} & & & \\ y_{20} & & & y_{40} & y_{31} & y_{22} \\ y_{11} & & & y_{31} & y_{22} & y_{13} \\ y_{02} & & & y_{22} & y_{13} & y_{04} \end{bmatrix} \succeq 0, \\ & \begin{bmatrix} y_{00} - 2y_{20} - y_{02} & & \\ & y_{20} - 2y_{40} - y_{22} & y_{11} - 2y_{31} - y_{13} \\ & y_{11} - 2y_{31} - y_{13} & y_{02} - 2y_{22} - y_{04} \end{bmatrix} \succeq 0, \\ & y_{00} = 1. \end{cases}$$

We have $\theta_2^{(1)} = \theta_2^{(*)} = \theta_2 = \theta^* = -0.125$.

6. A sparse representation theorem for positive polynomials

Suppose that the binary matrix $B_{0,\hat{d}}^{(*)}$ is not an all-one matrix. Then as was already noted in Sec. 5, the block-diagonal (up to permutation) matrix $B_{0,\hat{d}}^{(*)}$ induces a partition of the monomial basis $\mathbb{N}_{\hat{d}}^n$: two vectors $\boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{N}_{\hat{d}}^n$ belong to the same block if and only if the rows and columns indexed by $\boldsymbol{\beta}, \boldsymbol{\gamma}$ belong to the same block in $B_{0,\hat{d}}^{(*)}$. We first provide an interpretation of this partition in terms of *sign-symmetries*, a tool introduced in [17] to characterize block-diagonal SOS decompositions for nonnegative polynomials.

Definition 6.1. *Given a finite set* $\mathscr{A} \subseteq \mathbb{N}^n$, *the* sign-symmetries *of* $\mathscr{A}$ *are defined by all vectors* $\mathbf{r} \in \{0, 1\}^n$ *such that* $\mathbf{r}^T\boldsymbol{\alpha} \equiv 0$ (*mod* 2) *for all* $\boldsymbol{\alpha} \in \mathscr{A}$.

Theorem 6.2. *For a positive integer $\hat{d}$ and a finite set $\mathscr{A} \subseteq \mathbb{N}^n_{2\hat{d}}$, let $\mathbb{N}^n_{\hat{d}}$ be the standard monomial basis and let us define the sign-symmetries of $\mathscr{A}$ with the binary matrix $R = [\mathbf{r}_1, \ldots, \mathbf{r}_s]$ for some positive integer s. Then $\boldsymbol{\beta}, \boldsymbol{\gamma}$ belong to the same block in the partition of $\mathbb{N}^n_{\hat{d}}$ induced by $B^{(*)}_{0,\hat{d}}$ if and only if $R^T(\boldsymbol{\beta}+\boldsymbol{\gamma}) \equiv 0$ (mod 2).*

Proof. Let $G(V, E)$ be the adjacency graph of $B^{(*)}_{0,\hat{d}}$ with vertices $V = \mathbb{N}^n_{\hat{d}}$ and edges $E = \{(\boldsymbol{\beta}, \boldsymbol{\gamma}) \mid [B^{(*)}_{0,\hat{d}}]_{\boldsymbol{\beta}\boldsymbol{\gamma}} = 1\}$. Then the partition of V induced by $B^{(*)}_{0,\hat{d}}$ corresponds to the connected components of $G(V, E)$. Note that every connected component of G is a complete subgraph. Let $\operatorname{supp}(G) := \{\boldsymbol{\beta}+\boldsymbol{\gamma} \mid (\boldsymbol{\beta}, \boldsymbol{\gamma}) \in E\}$. For any $\boldsymbol{\alpha} = (\alpha_i) \in \mathbb{N}^n$, we call $\boldsymbol{\alpha} \,(\text{mod } 2) = (\alpha_i \,(\text{mod } 2)) \in \{0,1\}^n$ the sign type of $\boldsymbol{\alpha}$. Since for any $\boldsymbol{\beta}, \boldsymbol{\gamma} \in V$ with $\boldsymbol{\beta}+\boldsymbol{\gamma} \in (2\mathbb{N})^n$, we have $(\boldsymbol{\beta}, \boldsymbol{\gamma}) \in E$, then each $\boldsymbol{\beta} \in V$ with the same sign type belongs to the same connected component of G.

Claim I. If $\boldsymbol{\alpha} \in \operatorname{supp}(G)$, then for any $\boldsymbol{\alpha}' \in \mathbb{N}^n_{2\hat{d}}$ of the same sign type as $\boldsymbol{\alpha}$, we have $\boldsymbol{\alpha}' \in \operatorname{supp}(G)$.

Proof of Claim I. Suppose $\boldsymbol{\alpha} \in \operatorname{supp}(G)$. Since all $\boldsymbol{\alpha}' \in \mathbb{N}^n_{2\hat{d}}$ of sign type $\mathbf{0}$ are in $\operatorname{supp}(G)$, we can assume $\boldsymbol{\alpha} \,(\text{mod } 2) \neq \mathbf{0}$. For $\mathbf{s} = (s_i), \mathbf{s}' = (s'_i) \in \{0,1\}^n$, let $\tau(\mathbf{s}) := \sum_{i=1}^n s_i$ and we use $\mathbf{s} \perp \mathbf{s}'$ to indicate that $s_i = s'_i = 1$ holds for no i. If $\tau(\boldsymbol{\alpha} \,(\text{mod } 2))$ is odd, let $\mathbf{s}_1, \mathbf{s}_2 \in \{0,1\}^n \cap \mathbb{N}^n_{\hat{d}}$ such that $\boldsymbol{\alpha} \,(\text{mod } 2) = \mathbf{s}_1 + \mathbf{s}_2$ and $\mathbf{s}_1 \perp \mathbf{s}_2$. If $\tau(\boldsymbol{\alpha} \,(\text{mod } 2))$ is even, we further require that $\tau(\mathbf{s}_1), \tau(\mathbf{s}_2)$ have the same parity as $\hat{d}$. It is easy to check that such $\mathbf{s}_1, \mathbf{s}_2$ always exist. Then there must exist $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \in (2\mathbb{N})^n$ such that $\mathbf{s}_1 + \boldsymbol{\beta}_1, \mathbf{s}_2 + \boldsymbol{\beta}_2 \in \mathbb{N}^n_{\hat{d}}$ and $\boldsymbol{\alpha} = (\mathbf{s}_1 + \boldsymbol{\beta}_1) + (\mathbf{s}_2 + \boldsymbol{\beta}_2)$. It follows that $(\mathbf{s}_1 + \boldsymbol{\beta}_1, \mathbf{s}_2 + \boldsymbol{\beta}_2) \in E$ by the construction of $B^{(*)}_{0,\hat{d}}$. Because $\boldsymbol{\alpha}'$ has the same sign type as $\boldsymbol{\alpha}$, there must exist $\boldsymbol{\beta}'_1, \boldsymbol{\beta}'_2 \in (2\mathbb{N})^n$ such that $\mathbf{s}_1 + \boldsymbol{\beta}'_1, \mathbf{s}_2 + \boldsymbol{\beta}'_2 \in \mathbb{N}^n_{\hat{d}}$ and $\boldsymbol{\alpha}' = (\mathbf{s}_1 + \boldsymbol{\beta}'_1) + (\mathbf{s}_2 + \boldsymbol{\beta}'_2)$. Note that $\mathbf{s}_1 + \boldsymbol{\beta}_1$ (resp. $\mathbf{s}_2 + \boldsymbol{\beta}_2$) has the same sign type as $\mathbf{s}_1 + \boldsymbol{\beta}'_1$ (resp. $\mathbf{s}_2 + \boldsymbol{\beta}'_2$) and hence $\mathbf{s}_1 + \boldsymbol{\beta}_1$, $\mathbf{s}_1 + \boldsymbol{\beta}'_1$ (resp. $\mathbf{s}_2 + \boldsymbol{\beta}_2$, $\mathbf{s}_2 + \boldsymbol{\beta}'_2$) belong to the same connected component of G, which together with $(\mathbf{s}_1 + \boldsymbol{\beta}_1, \mathbf{s}_2 + \boldsymbol{\beta}_2) \in E$ implies that $\mathbf{s}_1 + \boldsymbol{\beta}_1$, $\mathbf{s}_1 + \boldsymbol{\beta}'_1$, $\mathbf{s}_2 + \boldsymbol{\beta}_2$, $\mathbf{s}_2 + \boldsymbol{\beta}'_2$ belong to the same connected component of G. So $(\mathbf{s}_1 + \boldsymbol{\beta}'_1, \mathbf{s}_2 + \boldsymbol{\beta}'_2) \in E$ and $\boldsymbol{\alpha}' \in \operatorname{supp}(G)$.

For a subset $S \subseteq \{0,1\}^n$, we say that S is closed under addition modulo 2 if $\mathbf{s}_1, \mathbf{s}_2 \in S$ implies $(\mathbf{s}_1 + \mathbf{s}_2) \,(\text{mod } 2) \in S$. The minimal set containing S with elements which are closed under addition modulo 2 is denoted by $\overline{S}$. Note that $\overline{S} \subseteq \mathbb{Z}_2^n$ is just the subspace spanned by S in $\mathbb{Z}_2^n$. This is because the subspace spanned by S is $\{\sum_i \mathbf{s}_i \,(\text{mod } 2) \mid \mathbf{s}_i \in S\}$ which is closed under addition modulo 2. Now let

$$S := \{\boldsymbol{\alpha} \,(\text{mod } 2) \mid \boldsymbol{\alpha} \in \mathscr{A}\}.$$

Claim II. The edge set of G is

$$E = \{(\boldsymbol{\beta}, \boldsymbol{\gamma}) \in V^2 \mid (\boldsymbol{\beta}+\boldsymbol{\gamma}) \,(\text{mod } 2) \in \overline{S}\},$$

which is equivalent to

$$\operatorname{supp}(G) = \{\boldsymbol{\alpha} \in \mathbb{N}^n_{2\hat{d}} \mid \boldsymbol{\alpha} \,(\text{mod } 2) \in \overline{S}\}.$$

Proof of Claim II. First we prove that $\operatorname{supp}(G) \subseteq \{\boldsymbol{\alpha} \in \mathbb{N}^n_{2\hat{d}} \mid \boldsymbol{\alpha} \,(\text{mod } 2) \in \overline{S}\}$. For $j = 0, \ldots, m$, let $\mathscr{S}^{(k)}_{j,\hat{d}}, C^{(k)}_{j,\hat{d}}, B^{(k)}_{j,\hat{d}}$ be defined as in Sec. 5 and let H_j^k, G_j^k be the adjacency graphs of $C^{(k)}_{j,\hat{d}}, B^{(k)}_{j,\hat{d}}$, respectively. By construction, one has $\operatorname{supp}(G) =$

$\bigcup_{k\ge 0}\bigcup_{j=0}^m \mathscr{S}_{j,\hat{d}}^{(k)}$. It suffices to prove

$$\bigcup_{j=0}^{m} \mathscr{S}_{j,\hat{d}}^{(k)} \subseteq \{\boldsymbol{\alpha}\in\mathbb{N}^n_{2\hat{d}} \mid \boldsymbol{\alpha}\,(\mathrm{mod}\ 2)\in\overline{S}\}. \tag{24}$$

Let us do induction on $k\ge 0$. It is obvious that (24) is valid for $k=0$. Now assume that (24) holds for a given $k\ge 0$. By (13), $\mathrm{supp}(H_0^{k+1})=\bigcup_{j=0}^m \mathscr{S}_{j,\hat{d}}^{(k)}$ and G_0^{k+1} is obtained from H_0^{k+1} by completing each connected component into a complete subgraph. Moreover, we have that for any $(\boldsymbol{\beta},\boldsymbol{\gamma}),(\boldsymbol{\beta},\boldsymbol{\gamma}')\in E(G_0^{k+1})$ (the edge set of G_0^{k+1}), if $(\boldsymbol{\beta}+\boldsymbol{\gamma})\,(\mathrm{mod}\ 2),(\boldsymbol{\beta}+\boldsymbol{\gamma}')\,(\mathrm{mod}\ 2)\in\overline{S}$, then $(\boldsymbol{\gamma}+\boldsymbol{\gamma}')\,(\mathrm{mod}\ 2)\in\overline{S}$. From these facts and the induction hypothesis, we deduce that $\mathscr{S}_{0,\hat{d}}^{(k+1)}=\mathrm{supp}(G_0^{k+1})\subseteq\{\boldsymbol{\alpha}\in\mathbb{N}^n_{2\hat{d}}\mid\boldsymbol{\alpha}\,(\mathrm{mod}\ 2)\in\overline{S}\}$.

For $1\le j\le m$ and for any $\boldsymbol{\alpha}'\in\mathrm{supp}(H_j^{k+1})$, by (14) we have $(\mathrm{supp}(g_j)+\boldsymbol{\alpha}')\cap\bigcup_{j=0}^m\mathscr{S}_{j,\hat{d}}^{(k)}\neq\emptyset$, which implies that $(\mathrm{supp}(g_j)+\boldsymbol{\alpha}')\cap\{\boldsymbol{\alpha}\in\mathbb{N}^n_{2\hat{d}}\mid\boldsymbol{\alpha}\,(\mathrm{mod}\ 2)\in\overline{S}\}\neq\emptyset$ by the induction hypothesis. It follows that $\boldsymbol{\alpha}'\,(\mathrm{mod}\ 2)\in\overline{S}$. So $\mathrm{supp}(H_j^{k+1})\subseteq\{\boldsymbol{\alpha}\in\mathbb{N}^n_{2\hat{d}}\mid\boldsymbol{\alpha}\,(\mathrm{mod}\ 2)\in\overline{S}\}$. Note that G_j^{k+1} is obtained from H_j^{k+1} by completing each connected component into a complete subgraph. Thus $\mathrm{supp}(G_j^{k+1})\subseteq\{\boldsymbol{\alpha}\in\mathbb{N}^n_{2\hat{d}}\mid\boldsymbol{\alpha}\,(\mathrm{mod}\ 2)\in\overline{S}\}$. By (16), $\mathscr{S}_{j,d}^{(k+1)}=\mathrm{supp}(g_j)+\mathrm{supp}(G_j^{k+1})$. Hence $\mathscr{S}_{j,d}^{(k+1)}\subseteq\{\boldsymbol{\alpha}\in\mathbb{N}^n_{2\hat{d}}\mid\boldsymbol{\alpha}\,(\mathrm{mod}\ 2)\in\overline{S}\}$. This completes the induction.

Next we need to prove that $\{\boldsymbol{\alpha}\in\mathbb{N}^n_{2\hat{d}}\mid\boldsymbol{\alpha}\,(\mathrm{mod}\ 2)\in\overline{S}\}\subseteq\mathrm{supp}(G)$, or equivalently

$$\overline{S}\cap\mathbb{N}^n_{2\hat{d}}\subseteq\mathrm{supp}(G)\,(\mathrm{mod}\ 2):=\{\boldsymbol{\alpha}\,(\mathrm{mod}\ 2)\mid\boldsymbol{\alpha}\in\mathrm{supp}(G)\}. \tag{25}$$

For any $\mathbf{s}\in\overline{S}\cap\mathbb{N}^n_{2\hat{d}}$, we can write $\mathbf{s}=\sum_{i=1}^l\mathbf{s}_i\,(\mathrm{mod}\ 2)$ for some $\{\mathbf{s}_i\}_i\subseteq S$. Let us prove (25) by induction on l. The case of $l=1$ follows from $\mathbf{s}_1\in S\subseteq\mathrm{supp}(G)\,(\mathrm{mod}\ 2)$. Now assume that $\sum_{i=1}^l\mathbf{s}_i\,(\mathrm{mod}\ 2)\in\mathrm{supp}(G)\,(\mathrm{mod}\ 2)$. Suppose $\sum_{i=1}^l\mathbf{s}_i\,(\mathrm{mod}\ 2)=(p_s)_{s=1}^n$ and $\mathbf{s}_{l+1}=(q_s)_{s=1}^n$. Let $a=|\{s\mid p_s=q_s=1\}|$, $b=|\{s\mid p_s=1,q_s=0\}|$ and $c=|\{s\mid p_s=0,q_s=1\}|$. If $a,b,c\le\hat{d}$, let I be the set $\{s\mid p_s=q_s=1\}$; if $a>\hat{d}$, let I be any $\hat{d}$-subset of $\{s\mid p_s=q_s=1\}$; if $b>\hat{d}$, let I be any $\hat{d}$-subset of $\{s\mid p_s=1,q_s=0\}$; if $c>\hat{d}$, let I be any $\hat{d}$-subset of $\{s\mid p_s=0,q_s=1\}$. Then define $\mathbf{u}=(u_s)\in\{0,1\}^n\cap\mathbb{N}^n_{\hat{d}}$ by

$$u_s=\begin{cases}1, & s\in I,\\ 0, & \text{otherwise},\end{cases}$$

and let $\boldsymbol{v}=(\sum_{i=1}^l\mathbf{s}_i+\mathbf{u})\,(\mathrm{mod}\ 2)$, $\boldsymbol{\omega}=(\mathbf{s}_{l+1}+\mathbf{u})\,(\mathrm{mod}\ 2)$. One can check that $\boldsymbol{v},\boldsymbol{\omega}\in\mathbb{N}^n_{\hat{d}}$. Therefore we have $\sum_{i=1}^l\mathbf{s}_i\equiv\mathbf{u}+\boldsymbol{v}\,(\mathrm{mod}\ 2)$ and $\mathbf{s}_{l+1}\equiv\mathbf{u}+\boldsymbol{\omega}\,(\mathrm{mod}\ 2)$. By the induction hypothesis, $(\mathbf{u}+\boldsymbol{v})\,(\mathrm{mod}\ 2)\in\mathrm{supp}(G)\,(\mathrm{mod}\ 2)$ which implies $\mathbf{u}+\boldsymbol{v}\in\mathrm{supp}(G)$ by Claim I and hence $(\mathbf{u},\boldsymbol{v})\in E$. We also have $(\mathbf{u}+\boldsymbol{\omega})\,(\mathrm{mod}\ 2)\in S\subseteq\mathrm{supp}(G)\,(\mathrm{mod}\ 2)$ which implies $\mathbf{u}+\boldsymbol{\omega}\in\mathrm{supp}(G)$ by Claim I and $(\mathbf{u},\boldsymbol{\omega})\in E$. It follows that $(\boldsymbol{v},\boldsymbol{\omega})\in E$ and $\boldsymbol{v}+\boldsymbol{\omega}\in\mathrm{supp}(G)$. Thus $\sum_{i=1}^{l+1}\mathbf{s}_i=\sum_{i=1}^l\mathbf{s}_i+\mathbf{s}_{l+1}\equiv(\boldsymbol{v}+\boldsymbol{\omega})\,(\mathrm{mod}\ 2)\in\mathrm{supp}(G)\,(\mathrm{mod}\ 2)$ which completes the induction and also completes the proof of Claim II.

By definition, the set of sign-symmetries R of $\mathscr{A}$ is just the orthogonal complement space of S (or $\overline{S}$) in $\mathbb{Z}_2^n$. Thus $\boldsymbol{\beta}, \boldsymbol{\gamma}$ belong to the same connected component of $G(V,E)$ if and only if $R^T(\boldsymbol{\beta}+\boldsymbol{\gamma}) \equiv 0 \pmod 2$ by Claim II. $\square$

Remark 6.3. *Theorem 6.2 is applied for the standard monomial basis $\mathbb{N}^n_{\hat{d}}$. If we choose a smaller monomial basis, then we only have the "only if" part of the conclusion in Theorem 6.2.*

As a corollary of Theorem 6.2, we obtain the following sparse representation theorem for positive polynomials over basic compact semialgebraic sets.

Theorem 6.4. *Let $f \in \mathbb{R}[\mathbf{x}]$ and $\mathbf{K}$ be as in (10). Assume that the quadratic module $\mathcal{Q}(\mathbf{g})$ is Archimedean and that f is positive on $\mathbf{K}$. Let $\mathscr{A} = \operatorname{supp}(f) \cup \bigcup_{j=1}^m \operatorname{supp}(g_j)$ and let us define the sign-symmetries of $\mathscr{A}$ with the binary matrix R. Then f can be represented as*

$$f = s_0 + \sum_{j=1}^m s_j g_j,$$

for some SOS polynomials $s_0, s_1, \ldots, s_m$ satisfying $R^T\boldsymbol{\alpha} \equiv 0 \pmod 2$ for any $\boldsymbol{\alpha} \in \operatorname{supp}(s_j), j = 0, \ldots, m$.

Proof. By Putinar's Positivstellensatz [28], there exist SOS polynomials $t_0, t_1, \ldots, t_m$ such that

$$f = t_0 + \sum_{j=1}^m t_j g_j. \tag{26}$$

Let $d_j = \lceil \deg(t_j g_j)/2 \rceil, j = 1, \ldots, m$ and let $d = \max\{\lceil \deg(t_0)/2 \rceil, d_1, \ldots, d_m\}$. Let Q_0 be a Gram matrix associated to t_0 and indexed by the monomial basis $\mathbb{N}^n_d$, and Q_j be a Gram matrix associated to t_j and indexed by the monomial basis $\mathbb{N}^n_{d-d_j}, j = 1, \ldots, m$. Then set $s_0 = (\mathbf{x}^{\mathbb{N}^n_d})^T (B^{(*)}_{0,d} \circ Q_0) \mathbf{x}^{\mathbb{N}^n_d}$ and $s_j = (\mathbf{x}^{\mathbb{N}^n_{d-d_j}})^T (B^{(*)}_{j,d} \circ Q_j) \mathbf{x}^{\mathbb{N}^n_{d-d_j}}$ for $j = 1, \ldots, m$. For all $j = 0, \ldots, m$, $B^{(*)}_{j,d} \circ Q_j$ is block-diagonal up to permutation and Q_j is positive semidefinite, thus s_j is an SOS polynomial.

Following the notation from Theorem 6.2, let G be the adjacency graph of $B^{(*)}_{0,d}$. By construction, $\operatorname{supp}(s_0) \subseteq \operatorname{supp}(G)$. For $j = 1, \ldots, m$, let $B^{(k)}_{j,d}, B^{(*)}_{j,d}$ be defined as in Sec. 5 and let G^k_j, G_j be the adjacency graphs of $B^{(k)}_{j,d}, B^{(*)}_{j,d}$, respectively. By construction $\operatorname{supp}(G_j) = \bigcup_{k \geq 1} \operatorname{supp}(G^k_j)$. By the proof Claim II of Theorem 6.2, $\operatorname{supp}(G^k_j) \subseteq \operatorname{supp}(G)$ for all $k \geq 1$. It follows that $\operatorname{supp}(G_j) \subseteq \operatorname{supp}(G)$ for $j = 1, \ldots, m$.

Therefore, we have $\operatorname{supp}(s_j) \subseteq \operatorname{supp}(G_j) \subseteq \operatorname{supp}(G)$ for $1 \leq j \leq m$. Hence for any j and any $\boldsymbol{\alpha} \in \operatorname{supp}(s_j)$, one has $\boldsymbol{\alpha} \pmod 2 \in \overline{S}$, which implies $R^T\boldsymbol{\alpha} \equiv 0 \pmod 2$. Moreover, for any $\boldsymbol{\alpha}' \in \operatorname{supp}(g_j)$, we have $(\boldsymbol{\alpha}+\boldsymbol{\alpha}') \pmod 2 \in \overline{S}$ and for any $\boldsymbol{\alpha}'' \in \mathbb{N}^n_{2d} \backslash \operatorname{supp}(G)$, we have $(\boldsymbol{\alpha}''+\boldsymbol{\alpha}') \pmod 2 \notin \overline{S}$ since $\overline{S}$ is closed under addition modulo 2. From this fact we deduce that substituting t_i by s_i in (26) is just removing the terms whose exponents modulo 2 are not in $\overline{S}$ from the right hand side of (26). Doing so, one does not change the match of coefficients on both sides of the equality. Thus we have

$$f = s_0 + \sum_{j=1}^m s_j g_j,$$

with the desired property. □

Remark 6.5. *It is worth emphasizing that the sign-symmetries binary matrix R permits to detect whether there is a block structure to exploit only by inspecting the support of the input data polynomials f and g_j. That is, if R is non trivial, then no matter the relaxation order $\hat{d}$, there is a guarantee that a block structure exists and will be exploited in the SDP relaxation $(Q_{\hat{d}}^{(\star)})$, where the size of the blocks depends on $\hat{d}$. On the other hand, even if R is trivial, there might still exist a block structure, to be discovered by building the successive SDP relaxations $(Q_{\hat{d}}^{k})$.*

7. Numerical experiments

In this section, we present numerical results of the proposed primal-dual hierarchies (7)-(9) and (17)-(18) of block SDP relaxations for both unconstrained and constrained polynomial optimization problems, respectively. Our algorithm, named TSSOS, is implemented in Julia for constructing instances of the dual SDP problems (7) and (18), then relies on MOSEK to solve them. In the following subsections, we compare the performance of TSSOS with GloptiPoly [7] to solve the primal moment problem (11) of the dense hierarchy, and Yalmip [16] to solve its dual SOS problem (12). As for TSSOS, GloptiPoly and Yalmip rely on MOSEK to solve SDP problems.

Our TSSOS tool can be downloaded at github:TSSOS. All numerical examples were computed on an Intel Core i5-8265U@1.60GHz CPU with 8GB RAM memory and the WINDOWS 10 system. The notations that we use are listed in Table 1.

Table 1. The notations

n	the number of variables
$2d$	the degree
s	the number of terms
$\hat{d}$	the order of Lasserre's hierarchy
k	the order of the block hierarchy
bs	the size of monomial bases
mb	whose i-th entry is the maximal size of blocks obtained at the i-th block hierarchy
opt	whose i-th entry is the optimum obtained at the i-th block hierarchy
time	whose i-th entry is the time of computing the i-th block hierarchy
0	a number whose absolute value less than $1e$-05
#block	the size of blocks
$i\times j$	j blocks of size i
-	out of memory

7.1. Unconstrained polynomial optimization problems.

For the unconstrained case, let us first look at an illustrative example.

Example 7.1. *Let*

$$\begin{aligned}f =&4(\sum_{i=1}^4 p_i^2)^4\sum_{i=1}^4 p_i^2a_i^{10}-(\sum_{i=1}^4 p_i^2)^3\sum_{i=1}^4 p_i^2a_i^8\sum_{i=1}^4 p_i^2a_i^2-(\sum_{i=1}^4 p_i^2a_i^2)^5\\&+2(\sum_{i=1}^4 p_i^2)^2\sum_{i=1}^4 p_i^2a_i^6(\sum_{i=1}^4 p_i^2a_i^2)^2-3(\sum_{i=1}^4 p_i^2)^2(\sum_{i=1}^4 p_i^2a_i^4)^2\sum_{i=1}^4 p_i^2a_i^2\\&+3\sum_{i=1}^4 p_i^2\sum_{i=1}^4 p_i^2a_i^4(\sum_{i=1}^4 p_i^2a_i^2)^3-4(\sum_{i=1}^4 p_i^2)^3\sum_{i=1}^4 p_i^2a_i^6\sum_{i=1}^4 p_i^2a_i^4.\end{aligned}$$

The polynomial f has 8 variables and is of degree 20. We compute a basis by the Newton polytope method (see (3)) which has 1284 monomials. The first step of the block hierarchy (9) gives us a block-diagonalization as follows:

size	1	2	3	4	10	11	14	19	20	31	42
number	1	6	36	18	5	6	4	1	18	12	4

where the first line is the size of blocks and the second line is the number of blocks of the corresponding size. We obtain the optimum $-2.1617e$-06 at the first step of the block hierarchy (9). The whole computation takes only 12s! It turns out that the hierarchy converges at the first iteration for this polynomial.

• Randomly generated examples

Now we present the numerical results for randomly generated polynomials of two types. The first type is of the SOS form. More concretely, we consider the polynomial

$$f=\sum_{i=1}^t f_i^2\in\mathbf{randpoly1}(n,2d,t,p)\,,$$

constructed as follows: first randomly choose a subset of monomials M from $\mathbf{x}^{\mathbb{N}_d^n}$ with probability p, and then randomly assign the elements of M to $f_1,\dots,f_t$ with random coefficients between -1 and 1. We generate 18 random polynomials $F_1,\dots,F_{18}$ from 6 different classes, where

$$\begin{gathered}F_1,F_2,F_3\in\mathbf{randpoly1}(8,8,30,0.1),\\F_4,F_5,F_6\in\mathbf{randpoly1}(8,10,25,0.04),\\F_7,F_8,F_9\in\mathbf{randpoly1}(9,10,30,0.03),\\F_{10},F_{11},F_{12}\in\mathbf{randpoly1}(10,12,20,0.01),\\F_{13},F_{14},F_{15}\in\mathbf{randpoly1}(10,16,30,0.003),\\F_{16},F_{17},F_{18}\in\mathbf{randpoly1}(12,12,50,0.01).\end{gathered}$$

For these polynomials, we compute a monomial basis using the Newton polytope method (3). Table 2 displays the numerical results on these polynomials. Note that the time spent to compute a monomial basis is included in the total running time to solve SDP (9). In Table 3, we compare the performance of TSSOS, GloptiPoly and Yalmip on these polynomials. In Yalmip, we turn the option “sos.newton” on to compute a monomial basis by the Newton polytope method.

For these examples, TSSOS always provides a nice block-diagonalization and we obtain the same optimum as the dense moment-SOS method in much less time. Due to the memory limit, GloptiPoly (resp. Yalmip) cannot handle polynomials with more than 8 (resp. 10) variables while TSSOS can solve problems involving up to 12 variables.

Table 2. The results for randomly generated polynomials of type I

	n	$2d$	s	bs	mb	opt	time (s)
F_1	8	8	64	106	$[31, 105, 106]$	$[0, 0, 0]$	$[1.7, 3.8, 3.9]$
F_2	8	8	102	122	$[71, 122]$	$[0, 0]$	$[4.6, 11]$
F_3	8	8	104	150	$[102, 150]$	$[0, 0]$	$[8.8, 15]$
F_4	8	10	103	202	$[64, 202]$	$[0, 0]$	$[4.8, 83]$
F_5	8	10	85	201	$[66, 201]$	$[0, 0]$	$[4.2, 68]$
F_6	8	10	111	128	$[76, 128]$	$[0, 0]$	$[5.2, 20]$
F_7	9	10	101	145	$[35, 142, 145]$	$[0, 0, 0]$	$[3.2, 38, 42]$
F_8	9	10	166	178	$[67, 178]$	$[0, 0]$	$[6.5, 96]$
F_9	9	10	161	171	$[62, 170, 171]$	$[0, 0, 0]$	$[5.9, 89, 101]$
F_{10}	10	12	271	223	$[75, 220, 223]$	$[0, 0, 0]$	$[12, 403, 435]$
F_{11}	10	12	253	176	$[60, 167, 176]$	$[0, 0, 0]$	$[9.2, 98, 122]$
F_{12}	10	12	261	204	$[73, 204]$	$[0, 0]$	$[12, 324]$
F_{13}	10	16	370	1098	$[99, 1098]$	[0, -]	[36, -]
F_{14}	10	16	412	800	$[195, 800]$	[0, -]	[305, -]
F_{15}	10	16	436	618	$[186, 617, 618]$	[0, -, -]	[207, -, -]
F_{16}	12	12	488	330	$[129, 324, 330]$	[0, -, -]	[61, -, -]
F_{17}	12	12	351	264	$[26, 42, 151, 263, 264]$	[0, 0, 0, -, -]	[17, 0.45, 76, -, -]
F_{18}	12	12	464	316	$[45, 274, 316]$	[0, -, -]	[22, -, -]

Table 3. Comparison with GloptiPoly and Yalmip for randomly generated polynomials of type I

	time (s)				time (s)		
	TSSOS	GloptiPoly	Yalmip		TSSOS	GloptiPoly	Yalmip
F_1	1.7	306	10	F_{10}	12	-	474
F_2	4.6	348	13	F_{11}	9.2	-	147
F_3	8.8	326	19	F_{12}	12	-	350
F_4	4.8	-	92	F_{13}	36	-	-
F_5	4.2	-	72	F_{14}	305	-	-
F_6	5.2	-	22	F_{15}	207	-	-
F_7	3.2	-	44	F_{16}	61	-	-
F_8	6.5	-	143	F_{17}	17	-	-
F_9	5.9	-	109	F_{18}	22	-	-

The second type of randomly generated problems are polynomials whose Newton polytopes are scaled standard simplices. More concretely, we consider polynomials defined by

$$f = c_0 + \sum_{i=1}^{n} c_i x_i^{2d} + \sum_{j=1}^{s-n-1} c_j' \mathbf{x}^{\boldsymbol{\alpha}_j} \in \mathbf{randpoly2}(n, 2d, s)\,,$$

constructed as follows: we randomly choose coefficients c_i between 0 and 1, as well as $s-n-1$ vectors $\boldsymbol{\alpha}_j$ in $\mathbb{N}^n_{2d-1}\backslash\{\mathbf{0}\}$ with random coefficients c'_j between -1 and 1. We generate 18 random polynomials $G_1,\ldots,G_{18}$ from 6 different classes, where

$$G_1,G_2,G_3 \in \textbf{randpoly2}(8,8,15),$$
$$G_4,G_5,G_6 \in \textbf{randpoly2}(9,8,20),$$
$$G_7,G_8,G_9 \in \textbf{randpoly2}(9,10,15),$$
$$G_{10},G_{11},G_{12} \in \textbf{randpoly2}(10,8,20),$$
$$G_{13},G_{14},G_{15} \in \textbf{randpoly2}(11,8,20),$$
$$G_{16},G_{17},G_{18} \in \textbf{randpoly2}(12,8,25).$$

Table 4 displays the numerical results on these polynomials. Table 5 indicates similar performance and accuracy results as for randomly generated polynomials of type I. In Yalmip, we turn the option “sos.congruence” on to take sign-symmetries into account, which allows one to handle slightly more polynomials than GloptiPoly.

Table 4. The results for randomly generated polynomials of type II

	n	$2d$	s	bs	mb	opt	time (s)
G_1	8	8	15	495	$[126,219]$	$[-0.5758,-0.5758]$	$[8.5,26]$
G_2	8	8	15	495	$[86,169]$	$[-34.6897,-34.6897]$	$[2.6,21]$
G_3	8	8	15	495	$[59,75]$	$[0.7073,0.7073]$	$[1.0,3.3]$
G_4	9	8	20	715	$[170,715]$	$[-801.6920,\text{-}]$	$[40,\text{-}]$
G_5	9	8	20	715	$[160,365]$	$[-0.8064,-0.8064]$	$[24,322]$
G_6	9	8	20	715	$[186,331]$	$[-1.6981,-1.6981]$	$[31,126]$
G_7	9	10	15	2002	$[122,224]$	$[-1.2945,-1.2945]$	$[24,303]$
G_8	9	10	15	2002	$[143,170]$	$[-0.6622,-0.6622]$	$[28,195]$
G_9	9	10	15	2002	$[154,208]$	$[0.5180,0.5180]$	$[21,180]$
G_{10}	10	8	20	1001	$[133,525]$	$[-0.4895,\text{-}]$	$[13,\text{-}]$
G_{11}	10	8	20	1001	$[223,403]$	$[0.1867,0.1867]$	$[86,481]$
G_{12}	10	8	20	1001	$[208,511]$	$[0.4943,\text{-}]$	$[66,\text{-}]$
G_{13}	11	8	20	1365	$[110,296]$	$[-3.9625,-3.9625]$	$[13,580]$
G_{14}	11	8	20	1365	$[128,436]$	$[-2.1835,\text{-}]$	$[37,\text{-}]$
G_{15}	11	8	20	1365	$[174,272]$	$[0.0588,0.0588]$	$[36,310]$
G_{16}	12	8	25	1820	$[263,924]$	$[-688.0269,\text{-}]$	$[693,\text{-}]$
G_{17}	12	8	25	1820	$[256,924]$	$[-40.2178,\text{-}]$	$[333,\text{-}]$
G_{18}	12	8	25	1820	$[275,924]$	$[-14.2693,\text{-}]$	$[393,\text{-}]$

• Examples from networked systems

Next we consider Lyapunov functions emerging from some networked systems. In [6], the authors propose a structured SOS decomposition for those systems, which allows them to handle structured Lyapunov function candidates up to 50 variables.

The following polynomial is from Example 2 in [6]:

$$f=\sum_{i=1}^{N}a_i(x_i^2+x_i^4)-\sum_{i=1}^{N}\sum_{k=1}^{N}b_{ik}x_i^2x_k^2,$$

TABLE 5. Comparison with GloptiPoly and Yalmip for randomly generated polynomials of type II

	TSSOS		GloptiPoly		Yalmip	
	opt	time (s)	opt	time (s)	opt	time (s)
G_1	-0.5758	8.5	-0.5758	346	-0.5758	31
G_2	-34.6897	2.6	-34.690	447	-34.6897	24
G_3	0.7073	1.0	0.7073	257	0.7073	6.0
G_4	-801.692	40	-	-	-	-
G_5	-0.8064	24	-	-	-0.8064	363
G_6	-1.6981	31	-	-	-1.6981	141
G_7	-1.2945	24	-	-	-1.2945	322
G_8	-0.6622	28	-	-	-0.6622	233
G_9	0.5180	21	-	-	0.5180	249
G_{10}	-0.4895	13	-	-	-	-
G_{11}	0.1867	86	-	-	0.1867	536
G_{12}	0.4943	66	-	-	-	-
G_{13}	-3.9625	13	-	-	-3.9625	655
G_{14}	-2.1835	37	-	-	-	-
G_{15}	0.0588	36	-	-	0.0588	340
G_{16}	-688.0269	693	-	-	-	-
G_{17}	-40.2178	333	-	-	-	-
G_{18}	-14.2693	393	-	-	-	-

where a_i are randomly chosen from $[1, 2]$ and b_{ik} are randomly chosen from $[\frac{0.5}{N}, \frac{1.5}{N}]$. Here, N is the number of nodes in the network. The task is to determine whether f is globally nonnegative. Here we solve again SDP (9) with TSSOS for $N = 10, 20, 30, 40, 50, 60, 70, 80$. The results are listed in Table 6.

TABLE 6. The results for network problem I

N	10	20	30	40	50	60	70	80
mb	11	31	31	41	51	61	71	81
time (s)	0.006	0.03	0.10	0.34	0.92	1.9	4.7	12

For this example, the size of systems that can be handled in [6] is up to $N = 50$ nodes while our approach can easily handle systems with up to $N = 80$ nodes.

The following polynomial is from Example 3 in [6]:

$$V = \sum_{i=1}^{N} a_i(\frac{1}{2}x_i^2 - \frac{1}{4}x_i^4) + \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N} b_{ik}\frac{1}{4}(x_i - x_k)^4, \tag{27}$$

where a_i are randomly chosen from $[0.5, 1.5]$ and b_{ik} are randomly chosen from $[\frac{0.5}{N}, \frac{1.5}{N}]$. The task is to analyze the domain on which the Hamiltonian function V for a network of Duffing oscillators is positive definite. We use the following condition to establish an inner approximation of the domain on which V is positive definite:

$$f = V - \sum_{i=1}^{N} \lambda_i x_i^2 (g - x_i^2) \geq 0\,, \tag{28}$$

where $\lambda_i > 0$ are scalar decision variables and g is a fixed positive scalar. Clearly, the condition (28) ensures that V is positive definite when $x_i^2 < g$. Here we solve SDP (9) with TSSOS for $N = 10, 20, 30, 40, 50$. For this example, graphs arising in the block hierarchy are naturally chordal, so we simply exploit chordal decompositions. This example was also examined in [20] to demonstrate the advantage of SDSOS programming compared to dense SOS programming. Here we compare the running time of TSSOS with that of SPOT [22] which uses SDSOS programming (using MOSEK as an SDP solver). The results are listed in Table 7.

TABLE 7. The results for network problem II

N		10	20	30	40	50
mb		11	21	31	41	51
time (s)	TSSOS	0.01	0.06	0.17	0.50	0.89
	SDSOS	0.47	1.14	5.47	20	70

For this example, we find that our approach spends much less time compared to SDSOS programming. On the other hand, we can compute a positive definite form V after selecting a value for g up to 2 (which is the same as the maximal value obtained by dense SOS) while the method in [6] can select g up to 1.8 and the one based on SDSOS programming only works out for a maximal value of g up to around 1.5.

- **Broyden banded functions** The Broyden banded function ([33]) is defined by

$$f_{\mathrm{Bb}}(\mathbf{x}) = \sum_{i=1}^{n} (x_i(2 + 5x_i^2) + 1 - \sum_{j\in J_i} (1 + x_j)x_j)^2,$$

where $J_i = \{j \mid j \neq i, \max(1, i-5) \leq j \leq \min(n, i+1)\}$. We prove that f_{Bb} is nonnegative by solving SDP (9) with TSSOS for $n = 6, 7, 8, 9, 10$. We make a comparison between TSSOS and SparsePOP [34] which exploits correlative sparsity (SparsePOP uses SeDuMi as an SDP solver). For this example, since TSSOS and SparsePOP use different SDP solvers, the running time is not comparable directly. We thereby also provide the number of SDP variables involved in TSSOS and SparsePOP respectively. The results are displayed in Table 8.

7.2. **Constrained polynomial optimization problems.** For the constrained case, we also begin with an illustrative example.

Example 7.2. *Consider the following problem:*

$$\begin{cases} \min & f = 27 - ((x_1 - x_2)^2 + (y_1 - y_2)^2)((x_1 - x_3)^2 + (y_1 - y_3)^2) \\ & \qquad ((x_2 - x_3)^2 + (y_2 - y_3)^2) \\ s.t. & g_1 = ((x_1^2 + y_1^2) + (x_2^2 + y_2^2) + (x_3^2 + y_3^2)) - 3 \\ & g_2 = 3 - ((x_1^2 + y_1^2) + (x_2^2 + y_2^2) + (x_3^2 + y_3^2)) \end{cases}$$

Table 8. The results for Broyden banded functions

n		6	7	8	9	10
#block	TSSOS	64×1, 1×20	85×1, 1×35	108×1, 1×57	133×1, 1×87	160×1, 1×126
	SparsePOP	84×1	120×1	120×2	120×3	120×4
#SDP variables	TSSOS	4116	7260	11721	17776	25726
	SparsePOP	7056	14400	28800	43200	57600
time (s)	TSSOS	0.27	0.76	1.9	5.3	13
	SparsePOP	2.0	9.0	20	30	42

We consider the block hierarchies of SDP problems (18) *for* $\hat{d} = 3$ *and* $\hat{d} = 4$*. For* $\hat{d} = 3$*, and* $k = 1$*, we obtain the following block-diagonalization:*

$M_3(\mathbf{y})$	$31 \times 2, 7 \times 1, 1 \times 15$
$M_2(g_1\mathbf{y})$	$13 \times 1, 9 \times 1, 1 \times 6$
$M_2(g_2\mathbf{y})$	$13 \times 1, 9 \times 1, 1 \times 6$

and we obtain an optimal value of $-5.0324e$-08*. For* $\hat{d} = 3$*, and* $k = 2$*, we have:*

$M_3(\mathbf{y})$	$31 \times 2, 13 \times 1, 9 \times 1$
$M_2(g_1\mathbf{y})$	$13 \times 1, 9 \times 1, 3 \times 2$
$M_2(g_2\mathbf{y})$	$13 \times 1, 9 \times 1, 3 \times 2$

and an optimal value of $-1.6016e$-07*. For* $\hat{d} = 3$*, the hierarchy converges for* $k = 2$*.*

For $\hat{d} = 4$*, the hierarchy immediately converges for* $k = 1$*, yielding the following block-diagonalization:*

$M_4(\mathbf{y})$	$79 \times 1, 69 \times 1, 31 \times 2$
$M_3(g_1\mathbf{y})$	$31 \times 2, 13 \times 1, 9 \times 1$
$M_3(g_2\mathbf{y})$	$31 \times 2, 13 \times 1, 9 \times 1$

and an optimal value of $-2.5791e$-10*.*

Now we present the numerical results for the constrained polynomial optimization problems. We generate six randomly generated polynomials $H_1, \ldots, H_6$ of type II as objective functions f and minimize f over a basic semialgebraic set $\mathbf{K} \subseteq \mathbb{R}^n$ for two cases: the unit ball

$$\mathbf{K} = \{(x_1, \ldots, x_n) \in \mathbb{R}^n \mid g_1 = 1 - (x_1^2 + \cdots + x_n^2) \geq 0\},$$

and the unit hypercube

$$\mathbf{K} = \{(x_1, \ldots, x_n) \in \mathbb{R}^n \mid g_1 = 1 - x_1^2 \geq 0, \ldots, g_n = 1 - x_n^2 \geq 0\}.$$

We compare the performance of TSSOS and GloptiPoly in these two cases. We output the related numerical results in Table 9 and Table 10. As in the unconstrained case, TSSOS performs better than the dense moment-SOS without compromising accuracy.

8. Conclusions

We have provided a new variant of the moment-SOS hierarchy to handle polynomial optimization problems with term sparsity. This hierarchy shares the same

Table 9. The results for minimizing randomly generated polynomials of type II over unit balls

	$(n, 2d, s)$	$\hat{d}$	k	mb	TSSOS		GloptiPoly	
					opt	time (s)	opt	time (s)
H_1	(6,8,10)	4	1	$(59, 25)$	0.1362	0.67	0.1362	8.0
			2	$(59, 25)$	0.1362	0.39		
		5	1	$(113, 59)$	0.1362	3.0	0.1362	80
			2	$(113, 59)$	0.1362	3.1		
H_2	(7,8,12)	4	1	$(85, 36)$	0.1373	1.6	0.1373	34
			2	$(99, 40)$	0.1373	1.7		
		5	1	$(176, 85)$	0.1373	11	-	-
			2	$(212, 99)$	0.1373	21		
H_3	(8,8,15)	4	1	$(69, 23)$	0.1212	2.8	0.1212	225
			2	$(135, 45)$	0.1212	13		
		5	1	$(144, 69)$	0.1212	35	-	-
			2	$(333, 135)$	0.1212	425		
H_4	(9,6,15)	3	1	$(48, 17)$	0.8704	1.0	0.8704	16
			2	$(50, 17)$	0.8704	0.35		
		4	1	$(131, 48)$	0.8704	6.8	-	-
			2	$(140, 50)$	0.8704	9.7		
H_5	(10,6,20)	3	1	$(67, 22)$	0.5966	2.1	0.5966	48
			2	$(92, 27)$	0.5966	1.6		
		4	1	$(193, 67)$	0.5966	48	-	-
			2	$(274, 92)$	0.5966	77		
H_6	(11,6,20)	3	1	$(67, 19)$	0.1171	2.1	0.1171	115
			2	$(104, 28)$	0.1171	4.0		
		4	1	$(170, 67)$	0.1171	40	-	-
			2	$(356, 104)$	0.1171	389		

In this table, the first entry of mb is the maximal size of blocks corresponding to the moment matrix $M_{\hat{d}}(\mathbf{y})$ and the second entry of mb is the maximal size of blocks corresponding to the localizing matrix $M_{\hat{d}-d_1}(g_1\mathbf{y})$.

theoretical convergence guarantees with the standard one and our numerical benchmarks demonstrate the performance speedup which can be achieved in both unconstrained and constrained cases.

As already mentioned in Remark (3.2), one direction for further research would be to replace block-closure (resp. block-decomposition) by chordal-extension (resp. chordal-decomposition) to exploit term sparsity, in order to obtain a more sparse variant of the moment-SOS hierarchy. Another interesting direction of research would be to exploit both correlative sparsity and term sparsity at the same time.

Last but not least, it would be worth investigating if one can benefit from the same term sparsity exploitation for other variants of the moment-SOS hierarchy, including the ones dedicated to optimal control, approximations of sets of interest (maximal invariant, reachable set) in dynamical systems, or the ones dedicated to eigenvalue and trace optimization of polynomials in non-commuting variables.

TABLE 10. The results for minimizing randomly generated polynomials of type II over unit hypercubes

	$(n, 2d, s)$	$\hat{d}$	k	mb	TSSOS		GloptiPoly	
					opt	time (s)	opt	time (s)
H_1	(6,8,10)	4	1	$(59, 25)$	-0.4400	1.1	-0.4400	19
			2	$(59, 25)$	-0.4400	0.88		
		5	1	$(113, 59)$	-0.4400	8.0	-0.4400	237
			2	$(113, 59)$	-0.4400	9.1		
H_2	(7,8,12)	4	1	$(85, 34)$	-0.1289	3.0	-0.1289	101
			2	$(99, 40)$	-0.1289	4.1		
		5	1	$(176, 85)$	-0.1289	40	-	-
			2	$(212, 99)$	-0.1289	87		
H_3	(8,8,15)	4	1	$(69, 23)$	-0.1465	3.9	-0.1465	433
			2	$(135, 45)$	-0.1465	30		
		5	1	$(144, 69)$	-0.1465	77	-	-
			2	$(333, 135)$	-0.1465	900		
H_4	(9,6,15)	3	1	$(48, 10)$	0.1199	1.3	0.1199	27
			2	$(50, 17)$	0.1199	0.64		
		4	1	$(131, 48)$	0.1199	12	-	-
			2	$(140, 50)$	0.1199	26		
H_5	(10,6,20)	3	1	$(67, 13)$	-0.2813	2.1	-0.2813	69
			2	$(92, 27)$	-0.2813	2.7		
			3	$(92, 27)$	-0.2813	2.7		
		4	1	$(193, 67)$	-0.2813	75	-	-
			2	$(274, 92)$	-0.2813	181		
H_6	(11,6,20)	3	1	$(67, 15)$	-0.2316	2.6	-0.2316	211
			2	$(104, 28)$	-0.2316	7.5		
			3	$((104, 28)$	-0.2316	7.6		
		4	1	$(170, 67)$	-0.2316	103	-	-
			2	$(356, 104)$	-0.2316	1108		

In this table, the first entry of mb is the maximal size of blocks corresponding to the moment matrix $M_{\hat{d}}(\mathbf{y})$ and the second entry of mb is the maximal size of blocks corresponding to the localizing matrices $M_{\hat{d}-d_j}(g_j\mathbf{y}), j = 1, \ldots, m$.

Acknowledgments. The first and second author were supported from the Tremplin ERC Stg Grant ANR-18-ERC2-0004-01 (T-COPS project). The second author was supported by the FMJH Program PGMO (EPICS project) and EDF, Thales, Orange et Criteo. The second and third author received funding from ANITI, coordinated by the Federal University of Toulouse within the framework of French Program "Investing for the Future PIA3" program under the Grant agreement n°ANR-19-XXXX-000X.

REFERENCES

[1] A. A. Ahmadi and A. Majumdar, *DSOS and SDSOS optimization: LP and SOCP-based alternatives to sum of squares optimization*, 48th annual conference on information sciences and systems (CISS), 2014:1-5. 3, 9, 10

[2] V. Chandrasekaran and P. Shah, *Relative Entropy Relaxations for Signomial Optimization. SIAM J. Optim.* 26(2):1147–1173, 2016. 3
[3] M. D. Choi, T. Y. Lam, and B. Reznick, *Sums of squares of real polynomials*, Proceedings of Symposia in Pure mathematics, AMS, 58(1995): 103-126. 4
[4] M. Fukuda, M. Kojima, K. Murota, and K. Nakata. Exploiting sparsity in semidefinite programming via matrix completion. I. General framework. *SIAM J. Optim.*, 11(3):647–674, 2000/01. 2
[5] M. Ghasemi and M. Marshall, *Lower bounds for polynomials using geometric programming*, *SIAM J. Optim.*, 22(2):460–473, 2012. 3
[6] E. J. Hancock and A. Papachristodoulou, *Structured Sum of Squares for Networked Systems Analysis*, 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC). 20, 21, 22
[7] D. Henrion and J. B. Lasserre, *GloptiPoly: Global Optimization over Polynomials with Matlab and SeDuMi*, IEEE Conf. Decis. Control, Las Vegas, Nevada, 2002:747-752. 17
[8] S. Iliman and T. de Wolff, *Amoebas, nonnegative polynomials and sums of squares supported on circuits*, Res. Math. Sci. 3:3-9, 2016. 3
[9] Cédric Josz. *Application of polynomial optimization to electricity transmission networks.* Theses, Université Pierre et Marie Curie - Paris VI, July 2016. 2
[10] I. Klep, V. Magron, J. Povh, *Sparse Noncommutative Polynomial Optimization. preprint arXiv:1909.00569*, 2019. 2
[11] M. Kojima, S. Kim, H. Waki, *Sparsity in sums of squares of polynomials*, Math. Program., 103(2005):45-62. 4
[12] J. B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM Journal on Optimization, 11(3)(2001):796-817. 1, 8, 10, 12
[13] J.-B. Lasserre. Convergent SDP-relaxations in polynomial optimization with sparsity. *SIAM J. Optim.*, 17(3):822–843, 2006. 2, 3
[14] J.-B. Lasserre, K.-C. Toh, and S. Yang. A bounded degree SOS hierarchy for polynomial optimization. *EURO J. Comput. Optim.*, 5(1-2):87–117, 2017. 2
[15] M. Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging applications of algebraic geometry*, volume 149 of *IMA Vol. Math. Appl.*, pages 157–270. Springer, New York, 2009. 1
[16] J. Löfberg, YALMIP: a toolbox for modeling and optimization in MATLAB, In 2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508), 284-289. 17
[17] J. Löfberg, *Pre- and Post-Processing Sum-of-Squares Programs in Practice*, IEEE Transactions on Automatic Control, 54(5)(2009):1007-1011. 2, 13
[18] V. Magron, G. Constantinides, and A. Donaldson. Certified roundoff error bounds using semidefinite programming. *ACM Trans. Math. Software*, 43(4):Art. 34, 31, 2017. 2
[19] V. Magron. Interval Enclosures of Upper Bounds of Roundoff Errors Using Semidefinite Programming. *ACM Trans. Math. Softw.*, 44(4):41:1–41:18, June 2018. 2
[20] A. Majumdar, A. A. Ahmadi and R. Tedrake, *Control and verification of high-dimensional systems with DSOS and SDSOS programming*, In 53rd IEEE Conference on Decision and Control, 2014:394-401. 22
[21] A. Marandi, E. D. Klerk, and J. Dahl, *Solving sparse polynomial optimization problems with chordal structure using the sparse bounded-degree sum-of-squares hierarchy*, Discrete Applied Mathematics, 2017.
[22] A. Megretski, Systems polynomial optimization tools (SPOT), 2010. Available at https://github.com/spot-toolbox/spotless. 22
[23] J. Miller, Y. Zheng, M. Sznaier, and A. Papachristodoulou, *Decomposed Structured Subsets for Semidefinite and Sum-of-Squares Optimization*, *preprint arXiv:1911.12859*, 2019. 3
[24] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota. Exploiting sparsity in semidefinite programming via matrix completion. II. Implementation and numerical results. *Math. Program.*, 95(2, Ser. B):303–327, 2003. 2
[25] P. A. Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*, Ph.D. Thesis, California Institute of Technology, 2000.
[26] F. Permenter, P. A. Parrilo, *Basis selection for SOS programs via facial reduction and polyhedral approximations*, Decision and Control, IEEE, 2014:6615-6620. 4

[27] F. Permenter, P. A. Parrilo, *Finding sparse, equivalent SDPs using minimal coordinate projections*, In 54th IEEE Conference on Decision and Control, CDC 2015, Osaka, Japan, December 15-18, 2015:7274-7279. 2

[28] M. Putinar, *Positive polynomials on compact semialgebraic sets*, Indiana Univ. Math. J., 42(1993):969-984. 3, 16

[29] B. Reznick, *Extremal PSD forms with few terms*, Duke Math. J., 45(1978):363-374. 2, 4

[30] C. Riener, T. Theobald, L. J. Andrén, and J.-B. Lasserre. Exploiting symmetries in SDP-relaxations for polynomial optimization. *Math. Oper. Res.*, 38(1):122–141, 2013. 2

[31] M. Tacchi, T. Weisser, J.-B. Lasserre, and D. Henrion. Exploiting sparsity for semi-algebraic set volume computation. *preprint arXiv:1902.02976*, 2019. 2

[32] J. Wang, H. Li and B. Xia, *A New Sparse SOS Decomposition Algorithm Based on Term Sparsity*, in Proceedings of the 2019 on International Symposium on Symbolic and Algebraic Computation, ACM, 2019:347-354. 2, 6

[33] H. Waki, S. Kim, M. Kojima, and M. Muramatsu, *Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity*, SIAM Journal on Optimization, 17(1)(2016):218-242. 1, 2, 3, 22

[34] H. Waki, S. Kim, M. Kojima, M. Muramatsu, and H. Sugimoto. Algorithm 883: sparsePOP—a sparse semidefinite programming relaxation of polynomial optimization problems. *ACM Trans. Math. Software*, 35(2):Art. 15, 13, 2009. 2, 22

[35] T. Weisser, J. B. Lasserre, and K. C. Toh, *Sparse-BSOS: a bounded degree SOS hierarchy for large scale polynomial optimization with sparsity*, Mathematical Programming Computation, 10(1)(2018):1-32. 2