SIAM J. OPTIM.
Vol. 31, No. 1, pp. 30–58

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 system literature.

Key words. polynomial optimization, moment relaxation, sum of squares, term sparsity, moment-SOS hierarchy, semidefinite programming

AMS subject classifications. Primary, 14P10, 90C22; Secondary, 12D15, 12Y05

DOI. 10.1137/19M1307871

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, which is concerned with the structure of *monomials* involved in the problem, is different from the correlative sparsity pattern already studied in [37] 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 [18]. The *moment-sum of squares* (moment-SOS) hierarchy by Lasserre [15] 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

*Received by the editors December 19, 2019; accepted for publication (in revised form) October 14, 2020; published electronically January 7, 2021.
https://doi.org/10.1137/19M1307871

Funding: This work was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Actions, grant 813211 (POEMA). The work of the first and second authors was supported by Tremplin ERC Stg grant ANR-18-ERC2-0004-01 (T-COPS project). The work of the second author was supported by the FMJH Program PGMO (EPICS project) and EDF, Thales, Orange et Criteo. The work of the second and third authors was supported by ANITI, coordinated by the Federal University of Toulouse within the framework of the French Program "Investing for the Future C PIA3" under grant ANR-19-XXXX-000X. The work of the third author was supported by the European Research Council (ERC) under the European Union Horizon 2020 research and innovation program, grant 666981 (TAMING).

†Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS), F-31031 Toulouse, France (jwang@laas.fr, vmagron@laas.fr, lasserre@laas.fr).

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 dth step of the hierarchy is proportional to $\binom{n+d}{n}$, where n is the number of variables of the initial problem.

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 [33]. One can also exploit (i) the sparsity pattern satisfied by the variables of the initial problem [16, 37] (see also the related SparsePOP solver [38]) as well as (ii) the symmetries [34] of the problem. In particular, sparsity has been successively exploited for specific applications, e.g., for solving optimal power flow problems [12], for round-off error bound analysis [22, 21], or more recently for approximating the volume of sparse semialgebraic sets [36]. 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 then obtains a sparse variant of the moment-SOS hierarchy with quasi-block-diagonal 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. Under mild assumptions, global convergence of this sparse version of the moment-SOS hierarchy is guaranteed if the so-called *running intersection property* (RIP) holds. Recently, this methodology has been extended in [13] to sparse problems with noncommuting variables (for instance, matrices). Other SOS-based representations include the bounded degree sum of squares [17] with its sparse variant [42]. These two latter hierarchies come with the same convergence guarantees as the standard ones (under the same sparsity pattern assumption). 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 [20] and minimal coordinate projections [31] in the unconstrained case. More recently, *cross sparsity patterns*, a new attempt in this direction introduced in [39], apply 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 [39] *to the constrained case and in a more general perspective.*

All the above-mentioned hierarchies require one 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* (SONC) polynomials 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 [4, 8, 11] allow one to compute sums of nonnegative circuits and sums of AGEs by relying on geometric programming and signomial programming, respectively. In [2], 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/(DSOS)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 the resulting lower bounds is not easy in practice. Very recently, a combination of correlative sparsity and SDSOS has been proposed in [26]. 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.

Contributions. We provide a new sparse moment-SOS hierarchy based on term sparsity rather than correlative sparsity. This is in deep contrast with the sparse variant of the moment-SOS hierarchy developed in [16, 37].

In section 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 unconstrained polynomial optimization in section 4 and constrained polynomial optimization in section 5. In both cases, the iterative procedure leads to a converging moment-SOS hierarchy (called TSSOS hierarchy) of primal-dual relaxations involving *block-diagonal* SDP matrices. If the sizes of blocks are small with respect to the original SDP matrices, then the resulting SDP relaxations yield a significant computational saving.

The TSSOS hierarchy (in the constrained case) depends on two parameters, the relaxation order $\hat{d}$ and the sparse order k (corresponding to each iterative step), and hence allows one more level of flexibility by playing with the two parameters $\hat{d}$ and k. The optimal values of the TSSOS hierarchy, at fixed relaxation order $\hat{d}$, yield a nondecreasing sequence converging to the optimal value of the dense moment-SOS hierarchy at the same relaxation order in a few steps (typically two or three). In the unconstrained case, we prove that even at the first iterative step ($k = 1$), the optimal value of the corresponding SDP relaxation is already no more worth than the one obtained with the SDSOS-based decompositions [2].

We prove in section 6 that the block structure of the TSSOS hierarchy at each relaxation order converges to the block structure determined by the sign-symmetries related to the support of the input data. This also enables us to provide a new sparse variant of Putinar's Positivstellensatz [32] for positive polynomials over basic compact semialgebraic sets. In this representation, the supports of all SOS polynomials are reduced according to the sign-symmetries.

In section 7, we compare the efficiency and scalability of the TSSOS hierarchy with existing frameworks on randomly generated examples as well as on problems arising from the networked system literature. The numerical results demonstrate that TSSOS has a significantly better performance in terms of 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 iterative step ($k = 1$) of the TSSOS hierarchy is always the same as the one obtained from the dense moment-SOS hierarchy on all tested examples, a very encouraging sign of efficiency. Last but not least, we emphasize that in all numerical examples (except the Broyden banded function from [37]), the usual correlative sparsity pattern is dense or almost dense and so yields no or little computational savings (or cannot even be implemented).

As mentioned in Remark 3.2 and done in the companion paper [40], one can replace block-closure by chordal-extension to exploit term sparsity, in order to obtain an even more sparse variant of the moment-SOS hierarchy: this is the so-called Chordal-TSSOS moment-SOS hierarchy. In the present paper, we treat general polynomial optimization problems (POPs). However, as often the case, some correlative sparsity is present in the input data (description) of large-scale POPs. Therefore, a natural idea is to combine correlative sparsity with our current TSSOS framework of term sparsity for solving large-scale POPs. Such an extension (called CS-TSSOS) is considered in our recent work [41] and is nontrivial, as it requires extra care when manipulating monomials that involve variables of different cliques that appear in the correlative sparsity pattern. As a result, CS-TSSOS can handle large-scale POPs (e.g., instances of the celebrated Max-Cut and optimal power flow problems) with up to several thousands of variables.

2. Notation and preliminaries.

2.1. Notation and 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 the convex hull of $\mathrm{supp}(f)$, i.e., $\mathrm{New}(f) = \mathrm{conv}(\{\boldsymbol{\alpha} : \boldsymbol{\alpha} \in \mathrm{supp}(f)\})$. We use $|\cdot|$ to denote the cardinality of a set. For $\mathscr{A}_1, \mathscr{A}_2 \subseteq \mathbb{N}^n$, let $\mathscr{A}_1 + \mathscr{A}_2 := \{\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_2 \mid \boldsymbol{\alpha}_1 \in \mathscr{A}_1, \boldsymbol{\alpha}_2 \in \mathscr{A}_2\}$.

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

$$f(\mathbf{x}) = \sum_{i=1}^{t} f_i(\mathbf{x})^2, \tag{2.1}$$

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 provides a certificate for its global nonnegativity. For $d \in \mathbb{N}$, let $\mathbb{N}^n_d := \{\boldsymbol{\alpha} = (\alpha_i) \in \mathbb{N}^n \mid \sum_{i=1}^n \alpha_i \leq d\}$ and assume that $f \in \mathscr{P}(\mathbb{N}^n_{2d})$. If we choose the *standard monomial basis* $\mathbf{x}^{\mathbb{N}^n_d}$, then the SOS condition (2.1) is equivalent to the existence of a PSD matrix Q (which is called a *Gram matrix* [5]) such that

$$f(\mathbf{x}) = (\mathbf{x}^{\mathbb{N}^n_d})^T Q \mathbf{x}^{\mathbb{N}^n_d}, \tag{2.2}$$

which can be formulized as a semidefinite program (SDP).

We say that a polynomial $f \in \mathscr{P}(\mathbb{N}^n_{2d})$ is *sparse* if the number of elements in its support $\mathscr{A} = \mathrm{supp}(f)$ is much smaller than the number of elements in $\mathbb{N}^n_{2d}$ that forms a support of fully dense polynomials in $\mathscr{P}(\mathbb{N}^n_{2d})$. When $f(\mathbf{x})$ is a sparse polynomial in $\mathscr{P}(\mathbb{N}^n_{2d})$, the size of the corresponding SDP (2.2) can be reduced by computing a smaller monomial basis. In fact, the set $\mathbb{N}^n_d$ in (2.2) can be replaced by the integer points in half of the Newton polytope of f, i.e., by

$$\mathscr{B} = \frac{1}{2} \cdot \mathrm{New}(f) \cap \mathbb{N}^n \subseteq \mathbb{N}^n_d. \tag{2.3}$$

See [33] 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 [14, 30]. Throughout this paper, we will use 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 what follows and denote the monomial basis $\mathbf{x}^{\mathscr{B}}$ by the exponents $\mathscr{B}$.

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

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

For 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

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

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 $\mathbf{y} = (y_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathbb{N}^n}$ 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

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

3. Exploiting term sparsity in SOS decompositions. For a positive integer r, let $[r] := \{1, \dots, r\}$. 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 $\mathbb{Z}_2^{r\times r}$ ($\mathbb{Z}_2 := \{0, 1\}$) be the set of $r \times r$ binary matrices. The support of a binary matrix $B \in \mathbb{S}^r \cap \mathbb{Z}_2^{r\times r}$ 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 $B \in \mathbb{S}^r \cap \mathbb{Z}_2^{r\times r}$, we define the set of PSD matrices with the sparsity pattern represented by B as

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

Let $f(\mathbf{x}) = \sum_{\boldsymbol{\alpha}\in\mathscr{A}} f_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}$ with $\mathrm{supp}(f) = \mathscr{A}$, and let $\mathscr{B}$ be a monomial basis with $r = |\mathscr{B}|$. For any $\boldsymbol{\alpha} \in \mathscr{B} + \mathscr{B}$, associate it with a binary matrix $A_{\boldsymbol{\alpha}} \in \mathbb{S}^r \cap \mathbb{Z}_2^{r\times r}$ such that $[A_{\boldsymbol{\alpha}}]_{\boldsymbol{\beta\gamma}} = 1$ if and only if $\boldsymbol{\beta} + \boldsymbol{\gamma} = \boldsymbol{\alpha}$ for all $\boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathscr{B}$. Then $f(\mathbf{x})$ is an SOS polynomial if and only if there exists $Q \in \mathbb{S}^r_+$ such that the following coefficient matching condition holds:

$$\langle A_{\boldsymbol{\alpha}}, Q\rangle = f_{\boldsymbol{\alpha}} \quad \forall \boldsymbol{\alpha} \in \mathscr{B} + \mathscr{B}, \tag{3.1}$$

where we set $f_{\boldsymbol{\alpha}} = 0$ if $\boldsymbol{\alpha} \notin \mathscr{A}$. For later use, we also define $A_{\mathscr{S}} := \sum_{\boldsymbol{\alpha}\in\mathscr{S}} A_{\boldsymbol{\alpha}}$ for any subset $\mathscr{S} \subseteq \mathscr{B} + \mathscr{B}$. For convenience, we define a block-closure operation on binary matrices as follows.

DEFINITION 3.1. *A relation $R \subseteq [r] \times [r]$ is called* transitive *if $(i, j), (j, k) \in R$ implies $(i, k) \in R$. The* transitive closure *of R, denoted by $\overline{R}$, is the smallest relation that contains R and is transitive. For a binary matrix $B \in \mathbb{S}^r \cap \mathbb{Z}_2^{r\times r}$, let $R \subseteq [r] \times [r]$ be the adjacency relation of B, i.e., $(i, j) \in R$, if and only if $B_{ij} = 1$. Then define the* block-closure $\overline{B} \in \mathbb{S}^r \cap \mathbb{Z}_2^{r\times r}$ *as*

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

For a binary matrix $B \in \mathbb{S}^r \cap \mathbb{Z}_2^{r\times r}$, the evaluation of $\overline{B}$ has a graphical description (assume $B_{ii} = 1$ 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. Hence the evaluation of block-closure boils down to the computation of connected components of a graph, which can be done in linear time (in terms of the numbers of vertices and edges of the graph). 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 sizes 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 :

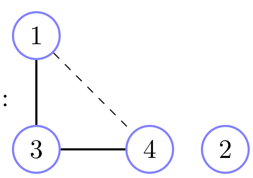

FIG. 1. *Block-closure and connected components.*

Remark 3.2. The block-closure operation $\overline{B}$ used in this paper can actually be replaced by a chordal-extension operation on adjacency graphs, which generally leads to sparser graphs, and then we take maximal cliques rather than connected components by virtue of the decomposition theorem on PSD matrices with chordal sparsity in [1]. See [39, 40] for more details. We use the block-closure in this paper since it is very simple to determine and can guarantee the convergence to the dense moment-SOS relaxation.

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 $B_{\mathscr{A}}^{(k)} \in \mathbb{S}^r \cap \mathbb{Z}_2^{r\times r}$ indexed by $\mathscr{B}$ via two successive steps:

(1) *Support-extension*: define a binary matrix $C_{\mathscr{A}}^{(k)} = A_{\mathscr{S}^{(k-1)}}$, i.e.,

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

(2) *Block-closure*: let $B_{\mathscr{A}}^{(k)} = \overline{C_{\mathscr{A}}^{(k)}}$ and $\mathscr{S}^{(k)} = \bigcup_{[B_{\mathscr{A}}^{(k)}]_{\boldsymbol{\beta\gamma}}=1} \{\boldsymbol{\beta} + \boldsymbol{\gamma}\}$.

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, the sequence of binary matrices $(B_{\mathscr{A}}^{(k)})_{k\geq 1}$ stabilizes after a finite number of steps. We denote the stabilized matrix by $B_{\mathscr{A}}^{(*)}$.

The reason why we set $\mathscr{S}^{(0)} = \mathscr{A} \cup (2\mathscr{B})$ rather than $\mathscr{S}^{(0)} = \mathscr{A}$ for initialization is explained in Remark 4.1.

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 the sparsity pattern represented by $B_{\mathscr{A}}^{(k)}$, i.e.,

$$\Sigma_k(\mathscr{A}) := \{f \in \mathscr{P}(\mathscr{A}) \mid \exists Q \in \mathbb{S}_+^r(B_{\mathscr{A}}^{(k)}) \text{ s.t. } f = (\mathbf{x}^{\mathscr{B}})^T Q \mathbf{x}^{\mathscr{B}}\}. \tag{3.2}$$

In addition, let

$$\Sigma_*(\mathscr{A}) := \{f \in \mathscr{P}(\mathscr{A}) \mid \exists Q \in \mathbb{S}_+^r(B_{\mathscr{A}}^{(*)}) \text{ s.t. } f = (\mathbf{x}^{\mathscr{B}})^T Q \mathbf{x}^{\mathscr{B}}\}. \tag{3.3}$$

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 $\mathscr{A} \subseteq \mathbb{N}^n$, one has $\Sigma_*(\mathscr{A}) = \Sigma(\mathscr{A})$.*

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, and we construct a matrix $\tilde{Q} \in \mathbb{S}_+^r$ by $\tilde{Q} = B_{\mathscr{A}}^{(*)} \circ Q$. We next show that $f = (\mathbf{x}^{\mathscr{B}})^T \tilde{Q} \mathbf{x}^{\mathscr{B}}$. Let $\mathscr{S}^{(*)} = \cup_{[B_{\mathscr{A}}^{(*)}]_{\boldsymbol{\beta}\boldsymbol{\gamma}}=1}\{\boldsymbol{\beta}+\boldsymbol{\gamma}\}$. By construction, $B_{\mathscr{A}}^{(*)}$ is stabilized under the support-extension operation and hence $B_{\mathscr{A}}^{(*)} = A_{\mathscr{S}^{(*)}}$. So we have $(\mathbf{x}^{\mathscr{B}})^T Q \mathbf{x}^{\mathscr{B}} - (\mathbf{x}^{\mathscr{B}})^T \tilde{Q} \mathbf{x}^{\mathscr{B}} = (\mathbf{x}^{\mathscr{B}})^T (Q - \tilde{Q}) \mathbf{x}^{\mathscr{B}} = (\mathbf{x}^{\mathscr{B}})^T (A_{(\mathscr{B}+\mathscr{B})} \circ Q - A_{\mathscr{S}^{(*)}} \circ Q) \mathbf{x}^{\mathscr{B}} = (\mathbf{x}^{\mathscr{B}})^T (A_{(\mathscr{B}+\mathscr{B})\setminus \mathscr{S}^{(*)}} \circ Q) \mathbf{x}^{\mathscr{B}}$. Note that $(\mathbf{x}^{\mathscr{B}})^T (A_{(\mathscr{B}+\mathscr{B})} \circ Q) \mathbf{x}^{\mathscr{B}} = f$ and $\text{supp}(f) = \mathscr{A}$. So for any $\boldsymbol{\alpha} \in (\mathscr{B}+\mathscr{B})\setminus \mathscr{A}$, $(\mathbf{x}^{\mathscr{B}})^T (A_{\boldsymbol{\alpha}} \circ Q) \mathbf{x}^{\mathscr{B}} = 0$. It follows that $(\mathbf{x}^{\mathscr{B}})^T (A_{(\mathscr{B}+\mathscr{B})\setminus \mathscr{S}^{(*)}} \circ Q) \mathbf{x}^{\mathscr{B}} = \sum_{\boldsymbol{\alpha} \in (\mathscr{B}+\mathscr{B})\setminus \mathscr{S}^{(*)}} (\mathbf{x}^{\mathscr{B}})^T (A_{\boldsymbol{\alpha}} \circ Q) \mathbf{x}^{\mathscr{B}} = 0$ since one has $\mathscr{A} \subseteq \mathscr{S}^{(*)}$ by construction. Therefore, $(\mathbf{x}^{\mathscr{B}})^T \tilde{Q} \mathbf{x}^{\mathscr{B}} = (\mathbf{x}^{\mathscr{B}})^T Q \mathbf{x}^{\mathscr{B}} = f$.

Note that $\tilde{Q}$ is block-diagonal (up to permutation) and each block of $\tilde{Q}$ is a principal 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 $\text{supp}(f) = \mathscr{A}$. Let $\mathscr{B} = \{\mathbf{0}\} \cup \{\mathbf{e}_k\}_{k=1}^n$ be the standard monomial basis and $Q = [q_{ij}]_{i,j=0}^n$ a Gram matrix of f. To show that $f \in \Sigma_1(\mathscr{A})$, it suffices 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. If $i = 0, j > 0$, from $[C_{\mathscr{A}}^{(1)}]_{0j} = 0$ one has $\mathbf{e}_j \notin \mathscr{A}$. If $i > 0, j = 0$, from $[C_{\mathscr{A}}^{(1)}]_{i0} = 0$ one has $\mathbf{e}_i \notin \mathscr{A}$. If $i, j > 0$, from $[C_{\mathscr{A}}^{(1)}]_{ij} = 0$ one has $\mathbf{e}_i + \mathbf{e}_j \notin \mathscr{A}$. In any of these three cases, one has $q_{ij} = 0$ 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 section 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 (3.2), then we obtain a hierarchy of sparse SOS relaxations of (P):

$$(4.1) \qquad (\mathrm{P}^k)^*: \quad \theta_k := \sup_{\lambda} \{\lambda \mid f(\mathbf{x}) - \lambda \in \Sigma_k(\mathscr{A})\}, \quad k = 1, 2, \ldots.$$

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

$$(4.2) \qquad (\mathrm{TSSOS}): \quad \theta_{tssos} := \sup_{\lambda} \{\lambda \mid f(\mathbf{x}) - \lambda \in \Sigma_*(\mathscr{A})\}.$$

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 the monomial basis. For each $k \geq 1$, the dual of $(\mathrm{P}^k)^*$ is the following block moment problem:

$$(4.3) \qquad (\mathrm{P}^k): \quad \begin{cases} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & B_{\mathscr{A}}^{(k)} \circ M_{\mathscr{B}}(\mathbf{y}) \succeq 0, \\ & y_{\mathbf{0}} = 1. \end{cases}$$

We call (4.1) and (4.3) the *TSSOS moment-SOS hierarchy* (*TSSOS hierarchy* in short) for the original problem (P) and call k the *sparse order*.

Remark 4.1. We point out that one should set $\mathscr{S}^{(0)} = \mathscr{A} \cup (2\mathscr{B})$ rather than $\mathscr{S}^{(0)} = \mathscr{A}$ for the iterative initialization in section 3 to improve the feasibility of $(\mathrm{P}^1)^*$ (an extra advantage for this choice is that it might also speed up the iteration). To see this, consider the polynomial $f = 1 + x + x^4$ and $\mathscr{A} = \mathrm{supp}(f) = \{0, 1, 4\}$. Take the monomial basis $\mathscr{B} = \{0, 1, 2\}$. If we set $\mathscr{S}^{(0)} = \mathscr{A}$, then $B_{\mathscr{A}}^{(1)} = \left[\begin{smallmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right]$. It is clear that $f-\lambda \notin \Sigma_1(\mathscr{A})$ for any λ. So the corresponding SDP $(\mathrm{P}^1)^*$ is infeasible in this case. On the other hand, if we set $\mathscr{S}^{(0)} = \mathscr{A} \cup (2\mathscr{B})$, then $B_{\mathscr{A}}^{(1)} = \left[\begin{smallmatrix}1&1&1\\1&1&1\\1&1&1\end{smallmatrix}\right]$ and $(\mathrm{P}^1)^*$ is feasible now.

Proposition 4.2. *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 [15] for the dense case and the observation that each block of $B_{\mathscr{A}}^{(k)} \circ M_{\mathscr{B}}(\mathbf{y})$ is a principal submatrix of $M_{\mathscr{B}}(\mathbf{y})$. ∎

Example 4.3. *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 we solve the SDPs (P^1), (P^2) and we obtain $\theta_1 = \theta_2 = \theta_{tssos} = \theta_{sos} = \theta^* \approx 0.4753$.

Relationship with DSOS/SDSOS optimization. The following definitions of DSOS and SDSOS have been introduced in [2]. For more details, the interested reader is referred to [2].

A symmetric matrix $Q \in \mathbb{S}^r$ is *diagonally dominant* if $Q_{ii} \geq \sum_{j\neq i} |Q_{ij}|$ for $i = 1,\dots,r$. 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 (2.2) 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 (2.2) 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, it holds that $\mathrm{DSOS}(\mathscr{A}) \subseteq \mathrm{SDSOS}(\mathscr{A}) \subseteq \Sigma(\mathscr{A})$.

THEOREM 4.4. *For a finite set* $\mathscr{A} \subseteq \mathbb{N}^n$*, one has* $\mathrm{SDSOS}(\mathscr{A}) \subseteq \Sigma_1(\mathscr{A})$.

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 matrix $\tilde{Q} \in \mathbb{S}^r$ by $\tilde{Q} = C^{(1)}_{\mathscr{A}} \circ Q$, i.e.,

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

By construction, $(\mathbf{x}^{\mathscr{B}})^T Q \mathbf{x}^{\mathscr{B}} - (\mathbf{x}^{\mathscr{B}})^T \tilde{Q} \mathbf{x}^{\mathscr{B}} = (\mathbf{x}^{\mathscr{B}})^T (A_{(\mathscr{B}+\mathscr{B})\setminus(\mathscr{A}\cup 2\mathscr{B})} \circ Q)\mathbf{x}^{\mathscr{B}} = 0$ since $(\mathbf{x}^{\mathscr{B}})^T (A_{(\mathscr{B}+\mathscr{B})} \circ Q)\mathbf{x}^{\mathscr{B}} = f$ and $\mathrm{supp}(f) = \mathscr{A}$. Thus, $(\mathbf{x}^{\mathscr{B}})^T \tilde{Q} \mathbf{x}^{\mathscr{B}} = (\mathbf{x}^{\mathscr{B}})^T Q \mathbf{x}^{\mathscr{B}} = f$. 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 $B^{(1)}_{\mathscr{A}} \circ \tilde{Q} = C^{(1)}_{\mathscr{A}} \circ \tilde{Q} = \tilde{Q}$ by construction. Thus, $f \in \Sigma_1(\mathscr{A})$. ∎

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

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

and

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

The above DSOS and SDSOS relaxations for polynomial optimization have been introduced and studied in [2]. By Theorem 4.4, 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

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

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

$$\mathbf{K} = \{\mathbf{x} \in \mathbb{R}^n : g_j(\mathbf{x}) \geq 0,\ j = 1, \ldots, m\} \tag{5.1}$$

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 $d = \max\{\lceil \deg(f)/2 \rceil, d_1, \ldots, d_m\}$, where $g_0 := 1$. With $\hat{d} \geq d$ being a positive integer, the Lasserre hierarchy [15] of moment semidefinite relaxations of (Q) is defined by

$$(\mathrm{Q}_{\hat{d}}): \quad \begin{cases} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & M_{\hat{d}}(\mathbf{y}) \succeq 0, \\ & M_{\hat{d}-d_j}(g_j \mathbf{y}) \succeq 0, \quad j = 1, \ldots, m, \\ & y_{\mathbf{0}} = 1, \end{cases} \tag{5.2}$$

with the optimal value denoted by $\theta_{\hat{d}}$, and we call $\hat{d}$ the *relaxation order*. Let $\mathbb{N}^n_{2(\hat{d}-d_j)}$ be the standard monomial basis for $j = 0, \ldots, m$. The dual of (5.2) is an SDP equivalent to the following SOS problem:

$$(\mathrm{Q}_{\hat{d}})^*: \quad \begin{cases} \sup & \lambda \\ \text{s.t.} & f - \lambda = s_0 + \sum_{j=1}^m s_j g_j\,, \\ & s_j \in \Sigma(\mathbb{N}^n_{2(\hat{d}-d_j)}), \quad j = 0, \ldots, m. \end{cases} \tag{5.3}$$

Let

$$\mathscr{A} = \mathrm{supp}(f) \cup \bigcup_{j=1}^m \mathrm{supp}(g_j). \tag{5.4}$$

Set $\mathscr{S}^{(0)}_{0,\hat{d}} = \mathscr{A} \cup (2\mathbb{N}^n_{\hat{d}})$ 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 $B^{(k)}_{j,\hat{d}} \in \mathbb{S}^{r_j} \cap \mathbb{Z}_2^{r_j \times r_j}$, indexed by $\mathbb{N}^n_{\hat{d}-d_j}$, $j = 0, \ldots, m$, via two successive steps:

(1) *Support-extension*: define a binary matrix $C^{(k)}_{j,\hat{d}} \in \mathbb{S}^{r_j} \cap \mathbb{Z}_2^{r_j \times r_j}$ with rows and columns indexed by $\mathbb{N}^n_{\hat{d}-d_j}$ by

$$[C^{(k)}_{j,\hat{d}}]_{\boldsymbol{\beta\gamma}} := \begin{cases} 1 & \text{if } (\mathrm{supp}(g_j) + \boldsymbol{\beta} + \boldsymbol{\gamma}) \cap \bigcup_{j=0}^m \mathscr{S}^{(k-1)}_{j,\hat{d}} \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

(2) *Block-closure*: let $B^{(k)}_{j,\hat{d}} = \overline{C^{(k)}_{j,\hat{d}}}$ and

$$\mathscr{S}^{(k)}_{j,\hat{d}} := \mathrm{supp}(g_j) + \bigcup_{[B^{(k)}_{j,\hat{d}}]_{\boldsymbol{\beta\gamma}}=1} \{\boldsymbol{\beta} + \boldsymbol{\gamma}\}. \tag{5.6}$$

Therefore, with $k \geq 1$, we can further consider a block moment relaxation of $(\mathrm{Q}_{\hat{d}})$ (5.2):

$$(5.7)\qquad (\mathrm{Q}^k_{\hat{d}}): \quad \begin{cases} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & B^{(k)}_{0,\hat{d}} \circ M_{\hat{d}}(\mathbf{y}) \succeq 0, \\ & B^{(k)}_{j,\hat{d}} \circ M_{\hat{d}-d_j}(g_j\mathbf{y}) \succeq 0, \quad j = 1,\ldots,m, \\ & y_{\mathbf{0}} = 1, \end{cases}$$

with the optimal value denoted by $\theta^{(k)}_{\hat{d}}$. By construction, we then have the inclusion $\mathrm{supp}(B^{(k)}_{j,\hat{d}}) \subseteq \mathrm{supp}(B^{(k+1)}_{j,\hat{d}})$ for all $k \geq 1$ and $j = 0,\ldots,m$. Hence, the sequence of binary matrices $(B^{(k)}_{j,\hat{d}})_{k\geq 1}$ stabilizes for all j after a finite number of steps. We denote the stabilized matrices by $B^{(*)}_{j,\hat{d}}$, $j = 0,\ldots,m$, and denote the corresponding SDP problem (5.7) by $(\mathrm{Q}^{\mathrm{ts}}_{\hat{d}})$ with optimal value $\theta^*_{\hat{d}}$.

Remark 5.1. As for the unconstrained case (see Remark 4.1), we initialize by letting $\mathscr{S}^{(0)}_{0,\hat{d}} = \mathscr{A} \cup (2\mathbb{N}^n_{\hat{d}})$ rather than $\mathscr{S}^{(0)}_{0,\hat{d}} = \mathscr{A}$. The setting of $\mathscr{A} = \mathrm{supp}(f) \cup \bigcup_{j=1}^m \mathrm{supp}(g_j)$ can be understood by noting that for $j = 1,\ldots,m$, we have $\mathbf{0} \in \mathbb{N}^n_{\hat{d}-d_j}$ (i.e., "1" is part of the monomial basis of the SOS multiplier s_j associated to g_j) and $\mathrm{supp}(g_j) + \mathbf{0} + \mathbf{0} = \mathrm{supp}(g_j)$. As we will see later, this setting is crucial for Corollary 6.8.

THEOREM 5.2. *For fixed $\hat{d} \geq d$, the sequence $(\theta^{(k)}_{\hat{d}})_{k\geq 1}$ of optimal values of* (5.7) *is monotone nondecreasing and in addition, $\theta^*_{\hat{d}} = \theta_{\hat{d}}$.*

Proof. Since $\mathrm{supp}(B^{(k)}_{j,\hat{d}}) \subseteq \mathrm{supp}(B^{(k+1)}_{j,\hat{d}})$ and $B^{(k)}_{j,\hat{d}}$ is block-diagonal (up to permutation) for all j,k, $(\mathrm{Q}^k_{\hat{d}})$ is a relaxation of $(\mathrm{Q}^{k+1}_{\hat{d}})$ and $(\mathrm{Q}_{\hat{d}})$. Therefore, $(\theta^{(k)}_{\hat{d}})_{k\geq 1}$ is nondecreasing and $\theta^*_{\hat{d}} \leq \theta_{\hat{d}}$.

Let $\mathscr{S}^{(*)}_{\hat{d}} = \cup_{j=0}^m (\mathrm{supp}(g_j) + \cup_{[B^{(*)}_{j,\hat{d}}]_{\boldsymbol{\beta\gamma}}=1} \{\boldsymbol{\beta}+\boldsymbol{\gamma}\})$. Suppose that $\mathbf{y} = (y_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathscr{S}^{(*)}_{\hat{d}}}$ is any feasible solution of $(\mathrm{Q}^{\mathrm{ts}}_{\hat{d}})$. Then define a sequence $\overline{\mathbf{y}} = (\overline{y}_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathbb{N}^n_{2\hat{d}}}$ by

$$\overline{y}_{\boldsymbol{\alpha}} = \begin{cases} y_{\boldsymbol{\alpha}} & \text{if } \boldsymbol{\alpha} \in \mathscr{S}^{(*)}_{\hat{d}}, \\ 0 & \text{otherwise.} \end{cases}$$

Because $B^{(*)}_{j,\hat{d}}$ is stabilized under the support-extension operation, by (5.5), one has $(\mathrm{supp}(g_j) + \boldsymbol{\beta} + \boldsymbol{\gamma}) \cap \mathscr{S}^{(*)}_{\hat{d}} = \emptyset$ for all $(\boldsymbol{\beta},\boldsymbol{\gamma}) \notin \mathrm{supp}(B^{(*)}_{j,\hat{d}})$ for $j = 0,\ldots,m$. Thus, we have $M_{\hat{d}-d_j}(g_j\overline{\mathbf{y}}) = B^{(*)}_{j,\hat{d}} \circ M_{\hat{d}-d_j}(g_j\mathbf{y})$ for $j = 0,\ldots,m$. Therefore, $\overline{\mathbf{y}}$ is also a feasible solution of $(\mathrm{Q}_{\hat{d}})$ and hence $L_{\mathbf{y}}(f) = L_{\overline{\mathbf{y}}}(f) \geq \theta_{\hat{d}}$. Hence, $\theta^*_{\hat{d}} \geq \theta_{\hat{d}}$ since $\mathbf{y}$ is an arbitrary feasible solution of $(\mathrm{Q}^{\mathrm{ts}}_{\hat{d}})$. It follows that $\theta^*_{\hat{d}} = \theta_{\hat{d}}$. ∎

THEOREM 5.3. *For fixed $k \geq 1$, the sequence $(\theta^{(k)}_{\hat{d}})_{\hat{d}\geq d}$ of optimal values of* (5.7) *is monotone nondecreasing.*

Proof. We only need to show that $\mathrm{supp}(B^{(k)}_{j,\hat{d}}) \subseteq \mathrm{supp}(B^{(k)}_{j,\hat{d}+1})$ for all j,k since this together with the fact that $B^{(k)}_{j,\hat{d}}, B^{(k)}_{j,\hat{d}+1}$ are block-diagonal (up to permutation) implies that $(\mathrm{Q}^k_{\hat{d}})$ is a relaxation of $(\mathrm{Q}^k_{\hat{d}+1})$ and hence $\theta^{(k)}_{\hat{d}} \leq \theta^{(k)}_{\hat{d}+1}$. Let us prove this conclusion by induction on k. For $k=1$, by (5.5), we have $\mathrm{supp}(C^{(1)}_{j,\hat{d}})\subseteq\mathrm{supp}(C^{(1)}_{j,\hat{d}+1})$ for $j=0,\ldots,m$, which implies that $\mathrm{supp}(B^{(1)}_{j,\hat{d}})\subseteq\mathrm{supp}(B^{(1)}_{j,\hat{d}+1})$ for $j=0,\ldots,m$. Now assume that $\mathrm{supp}(B^{(k)}_{j,\hat{d}})\subseteq\mathrm{supp}(B^{(k)}_{j,\hat{d}+1})$, $j=0,\ldots,m$, hold for a given $k\geq 1$. By (5.6) and by the induction hypothesis, we have $\mathscr{S}^{(k)}_{j,\hat{d}}\subseteq\mathscr{S}^{(k)}_{j,\hat{d}+1}$ for all j. Again by (5.5), we have $\mathrm{supp}(C^{(k+1)}_{j,\hat{d}})\subseteq\mathrm{supp}(C^{(k+1)}_{j,\hat{d}+1})$, which implies $\mathrm{supp}(B^{(k+1)}_{j,\hat{d}})\subseteq\mathrm{supp}(B^{(k+1)}_{j,\hat{d}+1})$ for $j=0,\ldots,m$. Thus, we complete the induction. ∎

Consequently, combining Theorem 5.2 with Theorem 5.3, we obtain the following two-level hierarchy of lower bounds for the optimal value of (Q):

$$
(5.8)\qquad
\begin{array}{ccccccc}
\theta^{(1)}_{d} & \leq & \theta^{(2)}_{d} & \leq & \cdots & \leq & \theta^{*}_{d}=\theta_{d}\\
\wedge| & & \wedge| & & & & \wedge|\\
\theta^{(1)}_{d+1} & \leq & \theta^{(2)}_{d+1} & \leq & \cdots & \leq & \theta^{*}_{d+1}=\theta_{d+1}\\
\wedge| & & \wedge| & & & & \wedge|\\
\vdots & & \vdots & & \vdots & & \vdots\\
\wedge| & & \wedge| & & & & \wedge|\\
\theta^{(1)}_{\hat{d}} & \leq & \theta^{(2)}_{\hat{d}} & \leq & \cdots & \leq & \theta^{*}_{\hat{d}}=\theta_{\hat{d}}\\
\wedge| & & \wedge| & & & & \wedge|\\
\vdots & & \vdots & & \vdots & & \vdots
\end{array}
$$

For each $j=1,\ldots,m$, writing $M_{\hat{d}-d_j}(g_j\mathbf{y})=\sum_{\boldsymbol{\alpha}}D^j_{\boldsymbol{\alpha}}y_{\boldsymbol{\alpha}}$ for appropriate symmetric matrices $\{D^j_{\boldsymbol{\alpha}}\}$, then the dual of $(\mathrm{Q}^k_{\hat{d}})$ reads as

$$
(5.9)\qquad (\mathrm{Q}^k_{\hat{d}})^*:\quad\begin{cases}\sup & \lambda\\ \text{s.t.} & \langle Q_0,A_{\boldsymbol{\alpha}}\rangle+\sum_{j=1}^m\langle Q_j,D^j_{\boldsymbol{\alpha}}\rangle+\lambda\delta_{\mathbf{0}\boldsymbol{\alpha}}=f_{\boldsymbol{\alpha}}\quad\forall\boldsymbol{\alpha}\in\mathscr{S}^{(k)}_{\hat{d}},\\ & Q_j\in\mathbb{S}^{r_j}_+(B^{(k)}_{j,\hat{d}}),\quad j=0,\ldots,m,\end{cases}
$$

where $\mathscr{S}^{(k)}_{\hat{d}}=\cup_{j=0}^m(\mathrm{supp}(g_j)+\cup_{[B^{(k)}_{j,\hat{d}}]_{\boldsymbol{\beta\gamma}}=1}\{\boldsymbol{\beta}+\boldsymbol{\gamma}\})$, $A_{\boldsymbol{\alpha}}$ is defined in section 3, and $\delta_{\mathbf{0}\boldsymbol{\alpha}}$ is the usual Kronecker symbol.

We call (5.7) and (5.9) the *TSSOS moment-SOS hierarchy* (*TSSOS hierarchy* in short) for the original problem (Q) and call k the *sparse order*.

PROPOSITION 5.4. *Let $f\in\mathbb{R}[\mathbf{x}]$, and let $\mathbf{K}$ be as in* (5.1)*. Assume that K has a nonempty interior. Then there is no duality gap between $(Q^k_{\hat{d}})$ and $(Q^k_{\hat{d}})^*$ for any $\hat{d}\geq d$ and $k\geq 1$.*

Proof. By the duality theory of convex programming, this easily follows from Theorem 4.2 of [15] 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 principal submatrix of $M_{\hat{d}-d_j}(g_j\mathbf{y})$ for all j,k. ∎

For any feasible solution of $(\mathrm{Q}^k_{\hat{d}})^*$, multiplying each side of the constraint in (5.9) by $\mathbf{x}^{\boldsymbol{\alpha}}$ for all $\boldsymbol{\alpha}\in\mathbb{N}^n_{2\hat{d}}$ and summing up yields

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

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 (5.10) as

$$(\mathbf{x}^{\mathbb{N}^n_{\hat{d}}})^T Q_0 \mathbf{x}^{\mathbb{N}^n_{\hat{d}}} + \sum_{j=1}^m g_j (\mathbf{x}^{\mathbb{N}^n_{\hat{d}-d_j}})^T Q_j \mathbf{x}^{\mathbb{N}^n_{\hat{d}-d_j}} = f - \lambda. \tag{5.11}$$

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 principal submatrices of Q_j. Then (5.11) reads as

$$\sum_{i=1}^{l_0} v_{ji}(\mathbf{x})^T Q_{ji} v_{ji}(\mathbf{x}) + \sum_{j=1}^m g_j \sum_{i=1}^{l_j} v_{ji}(\mathbf{x})^T Q_{ji} v_{ji}(\mathbf{x}) = f - \lambda. \tag{5.12}$$

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

$$\sum_{i=1}^{l_0} s_{ji} + \sum_{j=1}^m g_j \sum_{i=1}^{l_j} s_{ji} = f - \lambda. \tag{5.13}$$

Notice that (5.13) 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, (5.13) 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 5.5. *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*

$$C^{(1)}_{0,2} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \quad \textit{and} \quad C^{(1)}_{1,2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

This yields

$$B^{(1)}_{0,2} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \quad \textit{and} \quad B^{(1)}_{1,2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}.$$

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

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

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

6. Sign-symmetries and 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 section 5, the block-diagonal (up to permutation) matrix $B_{0,\hat{d}}^{(*)}$ induces a partition of the monomial basis $\mathbb{N}^n_{\hat{d}}$: two vectors $\boldsymbol{\beta},\boldsymbol{\gamma}\in\mathbb{N}^n_{\hat{d}}$ 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 next provide an interpretation of this partition in terms of *sign-symmetries*, a tool introduced in [20] 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* $\mathbf{r}\in\mathbb{Z}_2^n$ *such that* $\mathbf{r}^T\boldsymbol{\alpha}\equiv 0\ (\text{mod } 2)$ *for all* $\boldsymbol{\alpha}\in\mathscr{A}$.

For any $\boldsymbol{\alpha}\in\mathbb{N}^n$, we define $(\boldsymbol{\alpha})_2:=(\alpha_1(\text{mod } 2),\dots,\alpha_n(\text{mod } 2))\in\mathbb{Z}_2^n$. We also use the same notation for any subset $\mathscr{A}\subseteq\mathbb{N}^n$, i.e., $(\mathscr{A})_2:=\{(\boldsymbol{\alpha})_2\mid\boldsymbol{\alpha}\in\mathscr{A}\}\subseteq\mathbb{Z}_2^n$.

For a subset $S\subseteq\mathbb{Z}_2^n$, the *subspace* spanned by S in $\mathbb{Z}_2^n$, denoted by $\overline{S}$, is the set $\{(\sum_i\mathbf{s}_i)_2\mid\mathbf{s}_i\in S\}$ and the *orthogonal complement space* of S in $\mathbb{Z}_2^n$, denoted by $S^\perp$, is the set $\{\boldsymbol{\alpha}\in\mathbb{Z}_2^n\mid\boldsymbol{\alpha}^T\mathbf{s}\equiv 0\,(\text{mod } 2)\,\forall\mathbf{s}\in S\}$.

Remark 6.2. By definition, the set of sign-symmetries of $\mathscr{A}$ is just the orthogonal complement space $(\mathscr{A})_2^\perp$ in $\mathbb{Z}_2^n$. Hence, the sign-symmetries of $\mathscr{A}$ can be essentially represented by a basis of the subspace $(\mathscr{A})_2^\perp$ in $\mathbb{Z}_2^n$.

LEMMA 6.3. *Let* $S\subseteq\mathbb{Z}_2^n$. *Then* $(S^\perp)^\perp=\overline{S}$.

Proof. It is immediate from the definitions. ∎

For an (undirected) graph $G(V,E)$ with $V\subseteq\mathbb{N}^n$, we define $\text{supp}(G):=\{\boldsymbol{\beta}+\boldsymbol{\gamma}\mid\{\boldsymbol{\beta},\boldsymbol{\gamma}\}\in E\}$, and we also use $E(G)$ to indicate the edge set of G.

LEMMA 6.4. *Suppose* B *is a* $\{0,1\}$*-binary matrix with rows and columns indexed by* $\mathscr{B}\subseteq\mathbb{N}^n$ *and* G *is its adjacency graph. Let* $\overline{G}$ *be the adjacency graph of* $\overline{B}$. *Then* $(\text{supp}(\overline{G}))_2\subseteq\overline{(\text{supp}(G))_2}$.

Proof. By definition, for any $\{\boldsymbol{\beta},\boldsymbol{\gamma}\}\in E(\overline{G})$, we need to show that $(\boldsymbol{\beta}+\boldsymbol{\gamma})_2\in\overline{(\text{supp}(G))_2}$. By virtue of the graphical description of block-closure, $\boldsymbol{\beta},\boldsymbol{\gamma}$ belong to the same connected component of G. Therefore, there is a path connecting $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ in G:

$\{\boldsymbol{\beta}, \boldsymbol{v}_1, \ldots, \boldsymbol{v}_r, \boldsymbol{\gamma}\}$ with $\{\boldsymbol{\beta}, \boldsymbol{v}_1\}, \{\boldsymbol{v}_r, \boldsymbol{\gamma}\} \in E(G)$ and $\{\boldsymbol{v}_i, \boldsymbol{v}_{i+1}\} \in E(G)$, $i = 1, \ldots, r-1$. From $(\boldsymbol{\beta}+\boldsymbol{v}_1)_2, (\boldsymbol{v}_1+\boldsymbol{v}_2)_2 \in (\text{supp}(G))_2$, we deduce that $(\boldsymbol{\beta}+\boldsymbol{v}_2)_2 \in \overline{(\text{supp}(G))_2}$. Likewise, we can prove that $(\boldsymbol{\beta}+\boldsymbol{v}_i)_2 \in \overline{(\text{supp}(G))_2}$ for $i = 3, \ldots, r+1$ with $\boldsymbol{v}_{r+1} := \boldsymbol{\gamma}$. Hence, $(\boldsymbol{\beta}+\boldsymbol{\gamma})_2 \in \overline{(\text{supp}(G))_2}$ as desired. ∎

THEOREM 6.5. *For a positive integer $\hat{d}$, let $\mathscr{A} \subseteq \mathbb{N}^n_{2\hat{d}}$ be defined as in* (5.4) *and assume that the sign-symmetries of $\mathscr{A}$ are given by the columns of the binary matrix R. Let $B^{(*)}_{0,\hat{d}}$ be defined as in section* 5. *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 $R^T(\boldsymbol{\beta}+\boldsymbol{\gamma}) \equiv 0 \pmod 2$.*

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 $\mathbb{N}^n_{\hat{d}}$ induced by $B^{(*)}_{0,\hat{d}}$ corresponds to the connected components of G. Note that every connected component of G is a complete subgraph.

CLAIM I. *If $\boldsymbol{\alpha} \in \text{supp}(G)$, then for any $\boldsymbol{\alpha}' \in \mathbb{N}^n_{2\hat{d}}$ with $(\boldsymbol{\alpha}')_2 = (\boldsymbol{\alpha})_2$, one has $\boldsymbol{\alpha}' \in \text{supp}(G)$.*

Proof of Claim I. Suppose $\boldsymbol{\alpha} \in \text{supp}(G)$. If $(\boldsymbol{\alpha})_2 = (\boldsymbol{\alpha}')_2 = \mathbf{0}$, assume $\boldsymbol{\alpha}' = \boldsymbol{\beta}+\boldsymbol{\gamma}$ for some $\boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{N}^n_{\hat{d}}$. Then $\boldsymbol{\beta}+\boldsymbol{\gamma} \in (2\mathbb{N})^n$. Hence, $\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \in E(G)$ and it follows that $\boldsymbol{\alpha}' \in \text{supp}(G)$. Now assume $(\boldsymbol{\alpha})_2 \neq \mathbf{0}$. For $\mathbf{s} = (s_i), \mathbf{s}' = (s'_i) \in \mathbb{Z}^n_2$, 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 $\tau((\boldsymbol{\alpha})_2)$ is odd, let $\mathbf{s}_1, \mathbf{s}_2 \in \mathbb{Z}^n_2 \cap \mathbb{N}^n_{\hat{d}}$ such that $(\boldsymbol{\alpha})_2 = \mathbf{s}_1+\mathbf{s}_2$ and $\mathbf{s}_1 \perp \mathbf{s}_2$. If $\tau((\boldsymbol{\alpha})_2)$ 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 $\{\mathbf{s}_1+\boldsymbol{\beta}_1, \mathbf{s}_2+\boldsymbol{\beta}_2\} \in E(G)$ since $\boldsymbol{\alpha} \in \text{supp}(G)$ and $B^{(*)}_{0,\hat{d}}$ is stabilized under the support-extension operation. Because $(\boldsymbol{\alpha}')_2 = (\boldsymbol{\alpha})_2$, 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 $(\mathbf{s}_1+\boldsymbol{\beta}_1)+(\mathbf{s}_1+\boldsymbol{\beta}'_1) \in (2\mathbb{N})^n$ and $(\mathbf{s}_2+\boldsymbol{\beta}_2)+(\mathbf{s}_2+\boldsymbol{\beta}'_2) \in (2\mathbb{N})^n$. Hence, $\{\mathbf{s}_1+\boldsymbol{\beta}_1, \mathbf{s}_1+\boldsymbol{\beta}'_1\}, \{\mathbf{s}_2+\boldsymbol{\beta}_2, \mathbf{s}_2+\boldsymbol{\beta}'_2\} \in E(G)$, which together with $\{\mathbf{s}_1+\boldsymbol{\beta}_1, \mathbf{s}_2+\boldsymbol{\beta}_2\} \in E(G)$ 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 $\{\mathbf{s}_1+\boldsymbol{\beta}'_1, \mathbf{s}_2+\boldsymbol{\beta}'_2\} \in E(G)$ and $\boldsymbol{\alpha}' \in \text{supp}(G)$. The proof of Claim I is finished. ∎

CLAIM II. *Let $S = (\mathscr{A})_2$. The edge set of G is*

$$E = \{\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \in V^2 \mid (\boldsymbol{\beta}+\boldsymbol{\gamma})_2 \in \overline{S}\},$$

which is equivalent to (by Claim I *and the fact that $B^{(*)}_{0,\hat{d}}$ is stabilized under the support-extension operation)*

$$\text{supp}(G) = \{\boldsymbol{\alpha} \in \mathbb{N}^n_{2\hat{d}} \mid (\boldsymbol{\alpha})_2 \in \overline{S}\}.$$

Proof of Claim II. First we prove that $\text{supp}(G) \subseteq \{\boldsymbol{\alpha} \in \mathbb{N}^n_{2\hat{d}} \mid (\boldsymbol{\alpha})_2 \in \overline{S}\}$. 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 section 5 and let H^k_j, G^k_j be the adjacency graphs of $C^{(k)}_{j,\hat{d}}, B^{(k)}_{j,\hat{d}}$, respectively. By construction, one has $\text{supp}(G) =$

$\bigcup_{k\ge0}\bigcup_{j=0}^{m}\mathscr{S}_{j,\hat d}^{(k)}$. It suffices to prove

$$\bigcup_{j=0}^{m}\mathscr{S}_{j,\hat d}^{(k)}\subseteq\{\boldsymbol{\alpha}\in\mathbb{N}^n_{2\hat d}\mid(\boldsymbol{\alpha})_2\in\overline{S}\}\tag{6.1}$$

for all k. Let us do induction on $k\ge0$. It is obvious that (6.1) is valid for $k=0$. Now assume that (6.1) holds for a given $k\ge0$. For $0\le j\le m$ and for any $\boldsymbol{\alpha}'\in\operatorname{supp}(H_j^{k+1})$, by (5.5) we have $(\operatorname{supp}(g_j)+\boldsymbol{\alpha}')\cap\bigcup_{j=0}^m\mathscr{S}_{j,\hat d}^{(k)}\ne\emptyset$, which implies that $(\operatorname{supp}(g_j)+\boldsymbol{\alpha}')\cap\{\boldsymbol{\alpha}\in\mathbb{N}^n_{2\hat d}\mid(\boldsymbol{\alpha})_2\in\overline{S}\}\ne\emptyset$ by the induction hypothesis. It follows that $(\boldsymbol{\alpha}')_2\in\overline{S}$. Thus, $\operatorname{supp}(H_j^{k+1})\subseteq\{\boldsymbol{\alpha}\in\mathbb{N}^n_{2\hat d}\mid(\boldsymbol{\alpha})_2\in\overline{S}\}$. Then by Lemma 6.4, $\operatorname{supp}(G_j^{k+1})\subseteq\{\boldsymbol{\alpha}\in\mathbb{N}^n_{2\hat d}\mid(\boldsymbol{\alpha})_2\in\overline{S}\}$. By (5.6), $\mathscr{S}_{j,\hat d}^{(k+1)}=\operatorname{supp}(g_j)+\operatorname{supp}(G_j^{k+1})$. Hence, $\mathscr{S}_{j,\hat d}^{(k+1)}\subseteq\{\boldsymbol{\alpha}\in\mathbb{N}^n_{2\hat d}\mid(\boldsymbol{\alpha})_2\in\overline{S}\}$ for all j. This completes the induction.

Next we need to prove that $\{\boldsymbol{\alpha}\in\mathbb{N}^n_{2\hat d}\mid(\boldsymbol{\alpha})_2\in\overline{S}\}\subseteq\operatorname{supp}(G)$, or equivalently

$$\overline{S}\cap\mathbb{N}^n_{2\hat d}\subseteq(\operatorname{supp}(G))_2.\tag{6.2}$$

For any $\mathbf{s}\in\overline{S}\cap\mathbb{N}^n_{2\hat d}$, we can write $\mathbf{s}=(\sum_{i=1}^l\mathbf{s}_i)_2$ for some $\{\mathbf{s}_i\}_i\subseteq S$. Let us prove (6.2) by induction on l. The case of $l=1$ follows from $\mathbf{s}_1\in S\subseteq(\operatorname{supp}(G))_2$. Now assume that $(\sum_{i=1}^l\mathbf{s}_i)_2\in(\operatorname{supp}(G))_2$. Suppose $(\sum_{i=1}^l\mathbf{s}_i)_2=(p_s)_{s=1}^n$ and $\mathbf{s}_{l+1}=(q_s)_{s=1}^n$. Let $J_1=\{s\mid p_s=1,q_s=0\}$, $J_2=\{s\mid p_s=q_s=1\}$, and $J_3=\{s\mid p_s=0,q_s=1\}$. If $|J_1|,|J_2|,|J_3|\le\hat d$, let $I=J_2$; if $|J_2|>\hat d$, let I be any $\hat d$-subset of J_2; if $|J_1|>\hat d$, let K be any $\hat d$-subset of J_1 and $I=J_1\backslash K$; if $|J_3|>\hat d$, let K be any $\hat d$-subset of J_3 and $I=J_3\backslash K$. Then define $\mathbf{u}=(u_s)\in\mathbb{Z}_2^n\cap\mathbb{N}^n_{\hat d}$ by

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

and let $\boldsymbol{v}=(\sum_{i=1}^l\mathbf{s}_i+\mathbf{u})_2$, $\boldsymbol{\omega}=(\mathbf{s}_{l+1}+\mathbf{u})_2$. Then $(\sum_{i=1}^l\mathbf{s}_i)_2=(\mathbf{u}+\boldsymbol{v})_2$ and $(\mathbf{s}_{l+1})_2=(\mathbf{u}+\boldsymbol{\omega})_2$. In the case of $|J_1|,|J_2|,|J_3|\le\hat d$, one has $\tau(\boldsymbol{v})=|J_1|\le\hat d$ and $\tau(\boldsymbol{\omega})=|J_3|\le\hat d$; in the case of $|J_2|>\hat d$, one has $\tau(\boldsymbol{v})=|J_1|+|J_2|-\hat d\le\hat d$ and $\tau(\boldsymbol{\omega})=|J_3|+|J_2|-\hat d\le\hat d$ because $(\sum_{i=1}^l\mathbf{s}_i)_2,\mathbf{s}_{l+1}\in\mathbb{N}^n_{2\hat d}$; in the case of $|J_1|>\hat d$, one has $\tau(\boldsymbol{v})=|J_1|+|J_2|-\hat d\le\hat d$ and $\tau(\boldsymbol{\omega})=|J_3|+|J_1|-\hat d\le\hat d$ because $(\sum_{i=1}^l\mathbf{s}_i)_2,\mathbf{s}=(\sum_{i=1}^{l+1}\mathbf{s}_i)_2\in\mathbb{N}^n_{2\hat d}$; in the case of $|J_3|>\hat d$, one has $\tau(\boldsymbol{v})=|J_1|+|J_3|-\hat d\le\hat d$ and $\tau(\boldsymbol{\omega})=|J_3|+|J_2|-\hat d\le\hat d$ because $\mathbf{s}=(\sum_{i=1}^{l+1}\mathbf{s}_i)_2,\mathbf{s}_{l+1}\in\mathbb{N}^n_{2\hat d}$. Consequently, $\boldsymbol{v},\boldsymbol{\omega}\in\mathbb{N}^n_{\hat d}$. By the induction hypothesis, $(\mathbf{u}+\boldsymbol{v})_2\in(\operatorname{supp}(G))_2$, which implies $\mathbf{u}+\boldsymbol{v}\in\operatorname{supp}(G)$ by Claim I and hence $\{\mathbf{u},\boldsymbol{v}\}\in E$ (because $B_{0,\hat d}^{(*)}$ is stabilized under the support-extension operation). We also have $(\mathbf{u}+\boldsymbol{\omega})_2\in S\subseteq(\operatorname{supp}(G))_2$, which implies $\mathbf{u}+\boldsymbol{\omega}\in\operatorname{supp}(G)$ by Claim I and hence $\{\mathbf{u},\boldsymbol{\omega}\}\in E$. It follows that $\{\boldsymbol{v},\boldsymbol{\omega}\}\in E$ and $\boldsymbol{v}+\boldsymbol{\omega}\in\operatorname{supp}(G)$. Thus, $(\sum_{i=1}^{l+1}\mathbf{s}_i)_2=(\boldsymbol{v}+\boldsymbol{\omega})_2\in(\operatorname{supp}(G))_2$, which completes the induction and also completes the proof of Claim II. ∎

By Lemma 6.3, *we have* $\overline{S}=R^{\perp}$. *Thus,* $\boldsymbol{\beta},\boldsymbol{\gamma}$ *belong to the same connected component of* G *if and only if* $(\boldsymbol{\beta}+\boldsymbol{\gamma})_2\in\overline{S}$ *by Claim* II, *which is equivalent to* $R^T(\boldsymbol{\beta}+\boldsymbol{\gamma})\equiv0\pmod 2$. ∎

Remark 6.6. Note that Theorem 6.5 is applied for the standard monomial basis $\mathbb{N}^n_{\hat d}$. If a smaller monomial basis is chosen, then we only have the "only if" part of the conclusion in Theorem 6.5. See Example 6.7.

EXAMPLE 6.7. *Let* $f = 1+x^2y^4+x^4y^2+x^4y^4-xy^2-3x^2y^2$ *and* $\mathscr{A} = \mathrm{supp}(f)$*. The monomial basis given by the Newton polytope method is* $\mathscr{B} = \{1, xy, xy^2, x^2y, x^2y^2\}$*. The sign-symmetries of* $\mathscr{A}$ *consist of two elements:* $(0, 0)$ *and* $(0, 1)$*. According to the sign-symmetries,* $\mathscr{B}$ *is partitioned into* $\{1, xy^2, x^2y^2\}$ *and* $\{xy, x^2y\}$ *(recall that* $\boldsymbol{\beta}, \boldsymbol{\gamma}$ *belong to the same block in the partition induced by the sign-symmetries R if and only if* $R^T(\boldsymbol{\beta}+\boldsymbol{\gamma}) \equiv 0 \pmod 2$*). On the other hand, we have*

$$C_{\mathscr{A}}^{(1)} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad B_{\mathscr{A}}^{(*)} = B_{\mathscr{A}}^{(1)} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Thus, the partition of $\mathscr{B}$ *induced by* $B_{\mathscr{A}}^{(*)}$ *is* $\{1, xy^2, x^2y^2\}$*,* $\{xy\}$ *and* $\{x^2y\}$*, which is a refinement of the partition determined by the sign-symmetries.*

By virtue of Theorem 6.5, the partition of the monomial basis $\mathbb{N}^n_{\hat{d}-d_j}$ induced by $B^{(*)}_{j,\hat{d}}$, $j = 1\ldots, m$, can also be characterized using sign-symmetries.

COROLLARY 6.8. *Notations are as in Theorem* 6.5*, and assume* $\mathscr{A} = \mathrm{supp}(f) \cup \bigcup_{j=1}^m \mathrm{supp}(g_j)$*. Let* $B^{(*)}_{j,\hat{d}}$ *be defined as in section* 5*. Then* $\boldsymbol{\beta}, \boldsymbol{\gamma}$ *belong to the same block in the partition of* $\mathbb{N}^n_{\hat{d}-d_j}$ *induced by* $B^{(*)}_{j,\hat{d}}$ *if and only if* $R^T(\boldsymbol{\beta}+\boldsymbol{\gamma}) \equiv 0 \pmod 2$*,* $j = 1\ldots, m$*.*

Proof. Let $G_j(V_j, E_j)$ be the adjacency graph of $B^{(*)}_{j,\hat{d}}$ with vertices $V_j = \mathbb{N}^n_{\hat{d}-d_j}$ and edges $E_j = \{\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \mid [B^{(*)}_{j,\hat{d}}]_{\boldsymbol{\beta}\boldsymbol{\gamma}} = 1\}$, $j = 1, \ldots, m$. Then the partition of $\mathbb{N}^n_{\hat{d}-d_j}$ induced by $B^{(*)}_{j,\hat{d}}$ corresponds to the connected components of G_j. Note also that every connected component of G_j is a complete subgraph.

If $\boldsymbol{\beta}, \boldsymbol{\gamma}$ belong to the same connected component of G_j, then $\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \in E_j$. So $\boldsymbol{\beta}+\boldsymbol{\gamma}+\mathrm{supp}(g_j) \subseteq \mathrm{supp}(G)$, which implies $(\boldsymbol{\beta}+\boldsymbol{\gamma}+\mathrm{supp}(g_j))_2 \subseteq (\mathrm{supp}(G))_2 \subseteq \overline{S}$ by Claim II in the proof of Theorem 6.5. Since $(\mathrm{supp}(g_j))_2 \subseteq (\mathscr{A})_2 \subseteq \overline{S}$, we have $(\boldsymbol{\beta}+\boldsymbol{\gamma})_2 \in \overline{S}$ and it follows that $R^T(\boldsymbol{\beta}+\boldsymbol{\gamma}) \equiv 0 \pmod 2$.

If $\boldsymbol{\beta}, \boldsymbol{\gamma}$ don't belong to the same connected component of G_j, then $\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \notin E_j$. So $\boldsymbol{\beta}+\boldsymbol{\gamma}+\mathrm{supp}(g_j) \not\subseteq \mathrm{supp}(G)$, which implies $(\boldsymbol{\beta}+\boldsymbol{\gamma})_2 \notin \overline{S}$ by Claim II in the proof of Theorem 6.5. Thus, $R^T(\boldsymbol{\beta}+\boldsymbol{\gamma}) \not\equiv 0 \pmod 2$. ∎

Remark 6.9. The assumption that $\mathscr{A} = \mathrm{supp}(f) \cup \bigcup_{j=1}^m \mathrm{supp}(g_j)$ in Corollary 6.8 relies on the fact that "1" is part of the monomial basis $\mathbf{x}^{\mathbb{N}^n_{\hat{d}-d_j}}$ of the SOS multiplier s_j associated to g_j; see Remark 5.1.

Theorem 6.5 together with Corollary 6.8 implies that the block structure of the TSSOS hierarchy at each relaxation order converges to the block structure determined by the sign-symmetries related to the support of the input data, under the assumption that the standard monomial bases are used.

Remark 6.10. Though it is guaranteed that at the final step of the TSSOS hierarchy an equivalent SDP (with block structure determined by sign-symmetries if the standard monomial bases are used) is retrieved, in practice it frequently happens that the same optimal value as the dense moment-SOS relaxation is achieved at an earlier step, even at the first step, but with a much cheaper computational cost, as we can see in section 7.

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

$$\mathcal{Q}(\mathbf{g}) := \left\{ s_0 + \sum_{j=1}^{m} s_j g_j \mid s_j \text{ is an SOS, } j = 0, \ldots, m \right\}. \tag{6.3}$$

The quadratic module $\mathcal{Q}(\mathbf{g})$ associated with $\mathbf{K}$ in (5.1) is said to be *Archimedean* if there exists $N > 0$ such that the quadratic polynomial $\mathbf{x} \mapsto N - \|\mathbf{x}\|^2$ belongs to $\mathcal{Q}(\mathbf{g})$.

As a corollary of Theorem 6.5 and Corollary 6.8, we obtain the following sparse representation theorem for positive polynomials over basic compact semialgebraic sets.

THEOREM 6.11. *Let $f \in \mathbb{R}[\mathbf{x}]$, and let $\mathbf{K}$ be as in* (5.1). *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 columns of 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 [32], there exist SOS polynomials $t_0, t_1, \ldots, t_m$ such that

$$f = t_0 + \sum_{j=1}^{m} t_j g_j. \tag{6.4}$$

Let $d_j = \lceil \deg(g_j)/2 \rceil$, $j = 0, \ldots, m$, and $\hat{d} = \max\{\lceil \deg(t_j g_j)/2 \rceil : j = 0, 1, \ldots, m\}$, with $g_0 = 1$. Let Q_j be a Gram matrix associated with t_j and indexed by the monomial basis $\mathbb{N}^n_{\hat{d}-d_j}$, $j = 0, \ldots, m$. Then set $s_j = (\mathbf{x}^{\mathbb{N}^n_{\hat{d}-d_j}})^T (B^{(*)}_{j,\hat{d}} \circ Q_j) \mathbf{x}^{\mathbb{N}^n_{\hat{d}-d_j}}$ for $j = 0, \ldots, m$, where $B^{(*)}_{j,\hat{d}}$ is defined as in section 5. For all $j = 0, \ldots, m$, $B^{(*)}_{j,\hat{d}} \circ Q_j$ is block-diagonal up to permutation and Q_j is positive semidefinite, and thus s_j is an SOS polynomial.

Following the notation from Theorem 6.5 and Corollary 6.8, let G be the adjacency graph of $B^{(*)}_{0,\hat{d}}$. By construction, $\operatorname{supp}(s_0) \subseteq \operatorname{supp}(G)$. For $j = 1, \ldots, m$, let $B^{(k)}_{j,\hat{d}}, B^{(*)}_{j,\hat{d}}$ be defined as in section 5 and let G_j^k, G_j be the adjacency graphs of $B^{(k)}_{j,\hat{d}}, B^{(*)}_{j,\hat{d}}$, respectively. By construction, $\operatorname{supp}(G_j) = \bigcup_{k \geq 1} \operatorname{supp}(G_j^k)$. By the proof of Claim II in Theorem 6.5, $\operatorname{supp}(G_j^k) \subseteq \operatorname{supp}(G)$ 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 $(\boldsymbol{\alpha})_2 \in \overline{S}$ by Claim II in the proof of Theorem 6.5, which implies $R^T \boldsymbol{\alpha} \equiv 0 \pmod 2$. Moreover, for any $\boldsymbol{\alpha}' \in \operatorname{supp}(g_j)$, we have $(\boldsymbol{\alpha} + \boldsymbol{\alpha}')_2 \in \overline{S}$ and for any $\boldsymbol{\alpha}'' \in \mathbb{N}^n_{2\hat{d}} \backslash \operatorname{supp}(G)$, we have $(\boldsymbol{\alpha}'')_2 \notin \overline{S}$ by Claim II in the proof of Theorem 6.5 and hence $(\boldsymbol{\alpha}'' + \boldsymbol{\alpha}')_2 \notin \overline{S}$. From these facts we deduce that substituting t_i by s_i in (6.4) is just removing the terms whose exponents modulo 2 are not in $\overline{S}$ from the right-hand side of (6.4). 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. ∎

7. Numerical experiments. In this section, we present numerical results of the proposed primal-dual hierarchies (4.1)–(4.3) and (5.7)–(5.9) 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 (4.1) and (5.9) and then relies on MOSEK [27] to solve them. TSSOS utilizes the Julia packages LightGraphs [3] to handle graphs and JuMP [6] to model SDP. In the following subsections, we compare the performance of TSSOS with that of GloptiPoly [10] and Yalmip [19]. As for TSSOS, GloptiPoly and Yalmip also rely on MOSEK to solve SDP problems.

Our TSSOS tool can be downloaded at https://github.com/wangjie212/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 timing includes the time for preprocessing (to get the block structure in TSSOS), the time for modeling SDP, and the time for solving SDP. Although the modeling part in Julia is usually faster than the one in MATLAB, typically the time for solving SDP is dominant on the tested examples in this paper and exceeds the preprocessing time and the modeling time by one order of magnitude.

The notations that we use are listed in Table 1.

TABLE 1
Notations.

n	the number of variables
$2d$	the degree
s	the number of terms
$\hat{d}$	the relaxation order of the Lasserre hierarchy
k	the sparse order of the TSSOS hierarchy
bs	the size of monomial bases
mb	the maximal size of blocks (or a vector whose kth entry is the maximal size of blocks obtained from the TSSOS hierarchy at sparse order k in Tables 2 and 4)
Opt	the optimal value (or a vector whose kth entry is the optimal value obtained from the TSSOS hierarchy at sparse order k in Tables 2 and 4)
Time	running time in seconds (or a vector whose kth entry is the time for computing the TSSOS hierarchy at sparse order k in Tables 2 and 4)
0	a number whose absolute value is less than 1×10^{-5}
#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*

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

The polynomial f has 8 variables and is of degree 20. We compute a basis by the Newton polytope method (2.3) which has 1284 monomials. The first step of the TSSOS hierarchy gives us a block structure 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 the blocks and the second line is the number of blocks of the corresponding size. We obtain the optimal value -2.1617×10^{-6} at the first step of the TSSOS hierarchy. 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 $\mathbf{x}^{\mathbb{N}^n_d}$ 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,[1] where

$$F_1, F_2, F_3 \in \mathbf{randpoly1}(8, 8, 30, 0.1),$$

$$F_4, F_5, F_6 \in \mathbf{randpoly1}(8, 10, 25, 0.04),$$

$$F_7, F_8, F_9 \in \mathbf{randpoly1}(9, 10, 30, 0.03),$$

$$F_{10}, F_{11}, F_{12} \in \mathbf{randpoly1}(10, 12, 20, 0.01),$$

$$F_{13}, F_{14}, F_{15} \in \mathbf{randpoly1}(10, 16, 30, 0.003),$$

$$F_{16}, F_{17}, F_{18} \in \mathbf{randpoly1}(12, 12, 50, 0.01).$$

For these polynomials, the sign-symmetry is always trivial. We compute a monomial basis using the Newton polytope method (2.3). Table 2 displays the numerical results on these polynomials. Note that the time for computing a monomial basis is included in the time of the first step of the TSSOS hierarchy. In Table 3, we compare the performance of TSSOS ($k = 1$), GloptiPoly, and Yalmip on these polynomials. In

[1]The polynomials can be downloaded from https://wangjie212.github.io/jiewang/code.html.

Yalmip, we turn the option "sos.newton" on to compute a monomial basis also by the Newton polytope method.

For these examples, TSSOS always provides a nice block structure at sparse order $k = 1$ and retrieves the same optimum as the dense moment-SOS relaxation in much less time. TSSOS is also significantly faster than Yalmip. 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
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				Time		
	TSSOS	GloptiPoly	Yalmip		TSSOS	GloptiPoly	Yalmip
F_1	1.7	306	4.9	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,[2] where

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

Table 4 displays the numerical results on these polynomials. Table 5 indicates similar efficiency and accuracy results on the comparison with GloptiPoly and Yalmip 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
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 [9], 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 [9]:

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

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 (4.1) at $k = 1$ with TSSOS for $N = 10, 20, 30, 40, 50, 60, 70, 80$. The results are listed in Table 6.

[2]The polynomials can be downloaded from https://wangjie212.github.io/jiewang/code.html.

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

	TSSOS		GloptiPoly		Yalmip	
	Opt	Time	Opt	Time (s)	Opt	Time
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	-	-	-	-

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	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 [9] 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 [9]:

$$V = \sum_{i=1}^{N} a_i \left(\frac{1}{2}x_i^2 - \frac{1}{4}x_i^4\right) + \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{N} b_{ik}\frac{1}{4}(x_i - x_k)^4, \tag{7.1}$$

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:

$$f = V - \sum_{i=1}^{N} \lambda_i x_i^2 (g - x_i^2) \geq 0\,, \tag{7.2}$$

where $\lambda_i > 0$ are scalar decision variables and g is a fixed positive scalar. Clearly, the condition (7.2) ensures that V is positive definite when $x_i^2 < g$. Here we solve SDP (4.1) at $k = 1$ with TSSOS for $N = 10, 20, 30, 40, 50$. For this example, graphs arising in the TSSOS hierarchy are naturally chordal, so we simply exploit chordal decompositions. This example was also examined in [23] to demonstrate the advantage of SDSOS programming compared to dense SOS programming. The method based on SDSOS programming was executed in SPOT [25] with MOSEK as a second-order cone programming solver. The results are listed in Table 7. The row "#var" in Table 7 indicates the number of decision variables.

Table 7
The results for network problem II.

N		10	20	30	40	50
#block	TSSOS	3×45, 1×10, 11×1	3×190, 1×20, 21×1	3×435, 1×30, 31×1	3×780, 1×40, 41×1	3×1225, 1×50, 51×1
	SDSOS	2×2145	2×26565	2×122760	2×370230	2×878475
#var	TSSOS	346	1391	3136	5581	8726
	SDSOS	6435	79695	368280	1110690	2635425
Time	TSSOS	0.01	0.06	0.17	0.50	0.89
	SDSOS	0.47	1.14	5.47	20	70

For this example, TSSOS uses many fewer decision variables than SDSOS programming and hence spends less time compared to SDSOS programming. On the other hand, TSSOS computes a positive definite form V after selecting a value for g up to 2 (which is the same as the maximal value obtained by the dense SOS), while the method in [9] 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 [37] is defined by

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

where $J_i = \{j \mid j \neq i, \max(1, i-5) \leq j \leq \min(n, i+1)\}$. We prove that f_{Bb} is nonnegative by solving SDP (4.1) at $k = 1$ with TSSOS for $n = 6, 7, 8, 9, 10$. We make a comparison between TSSOS and SparsePOP [38] which exploits correlative sparsity and uses SeDuMi [35] 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 decision variables involved in TSSOS and SparsePOP, respectively. The results are displayed in Table 8. The row "#var" in Table 8 indicates the number of decision variables.

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
#var	TSSOS	2100	3690	5943	8998	13006
	SparsePOP	3570	7260	14520	21780	29040
Time	TSSOS	0.27	0.76	1.9	5.3	13
	SparsePOP	2.0	9.0	20	30	42

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

Example 7.2. *Consider the following problem:*

$$\begin{cases} \min & f = 27 - ((x_1-x_2)^2+(y_1-y_2)^2)((x_1-x_3)^2+(y_1-y_3)^2) \\ & \qquad ((x_2-x_3)^2+(y_2-y_3)^2) \\ s.t. & g_1 = ((x_1^2+y_1^2)+(x_2^2+y_2^2)+(x_3^2+y_3^2)) - 3 \geq 0, \\ & g_2 = 3 - ((x_1^2+y_1^2)+(x_2^2+y_2^2)+(x_3^2+y_3^2)) \geq 0. \end{cases}$$

We consider the TSSOS hierarchy with $\hat{d} = 3$ and $\hat{d} = 4$. For $\hat{d} = 3$ and $k = 1$, we obtain the block structure

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

and we obtain an optimal value -5.0324×10^{-8}. 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 -1.6016×10^{-7}. For $\hat{d} = 3$, the hierarchy converges at $k = 2$.

For $\hat{d} = 4$, the hierarchy immediately converges at $k = 1$, yielding the block structure

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

and an optimal value -2.5791×10^{-10}.

Now we present the numerical results for constrained polynomial optimization problems. We generate six randomly generated polynomials $H_1, \ldots, H_6$ of type II[3] 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. The related numerical results are outputted in Tables 9 and 10. As in the unconstrained case, Tables 9 and 10 show that TSSOS performs much 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 theoretical convergence guarantees with the standard one, and our numerical benchmarks demonstrate the performance speedup which can be achieved in both the unconstrained and the constrained cases.

One direction of further research is to investigate whether 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 noncommuting variables.

[3]The polynomials can be downloaded from https://wangjie212.github.io/jiewang/code.html.

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

	$(n, 2d, s)$	$\hat{d}$	k	mb	TSSOS Opt	TSSOS Time	GloptiPoly Opt	GloptiPoly Time
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 $M_{\hat{d}-d_1}(g_1\mathbf{y})$.

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

	$(n, 2d, s)$	$\hat{d}$	k	mb	TSSOS Opt	TSSOS Time	GloptiPoly Opt	GloptiPoly Time
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 $M_{\hat{d}-d_j}(g_j\mathbf{y}), j = 1, \ldots, m$.

REFERENCES

[1] J. Agler, W. Helton, S. McCullough, and L. Rodman, *Positive semidefinite matrices with a given sparsity pattern*, Linear Algebra. Appl., 107 (1988), pp. 101–149.

[2] A. A. Ahmadi and A. Majumdar, *DSOS and SDSOS optimization: LP and SOCP-based alternatives to sum of squares optimization*, in Proceedings of the 48th Annual Conference on Information Sciences and Systems (CISS), 2014, pp. 1–5.

[3] S. Bromberger, J. Fairbanks, et al., *JuliaGraphs/LightGraphs.jl: An Optimized Graphs Package for the Julia Programming Language*, https://doi.org/10.5281/zenodo.889971, 2017.

[4] V. Chandrasekaran and P. Shah, *Relative entropy relaxations for signomial optimization*, SIAM J. Optim., 26 (2016), pp. 1147–1173, https://doi.org/10.1137/140988978.

[5] M. D. Choi, T. Y. Lam, and B. Reznick, *Sums of squares of real polynomials*, in K-Theory and Algebraic Geometry: Connections with Quadratic Forms and Division Algebras, Proc. Sympos. Pure Math. 58, AMS, Providence, RI, 1995, pp. 103–126.

[6] I. Dunning, J. Huchette, and M. Lubin, JuMP: A modeling language for mathematical optimization, SIAM Rev., 59 (2017), pp. 295–320, https://doi.org/10.1137/15M1020575.

[7] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, *Exploiting sparsity in semidefinite programming via matrix completion* I: *General framework*, SIAM J. Optim., 11 (2000), pp. 647–674, https://doi.org/10.1137/S1052623400366218.

[8] M. Ghasemi and M. Marshall, *Lower bounds for polynomials using geometric programming*, SIAM J. Optim., 22 (2012), pp. 460–473, https://doi.org/10.1137/110836869.

[9] E. J. Hancock and A. Papachristodoulou, *Structured sum of squares for networked systems analysis*, in Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC).

[10] D. Henrion and J. B. Lasserre, *GloptiPoly: Global optimization over polynomials with Matlab and SeDuMi*, ACM Trans. Math. Software, 29 (2003), pp. 165–194.

[11] S. Iliman and T. de Wolff, *Amoebas, nonnegative polynomials and sums of squares supported on circuits*, Res. Math. Sci., 3 (2016), 9.

[12] C. Josz, *Application of Polynomial Optimization to Electricity Transmission Networks*, Theses, Université Pierre et Marie Curie - Paris VI, Paris, France, 2016.

[13] I. Klep, V. Magron, and J. Povh, *Sparse Noncommutative Polynomial Optimization*, preprint, https://arxiv.org/abs/1909.00569, 2019.

[14] M. Kojima, S. Kim, and H. Waki, *Sparsity in sums of squares of polynomials*, Math. Program., 103 (2005), pp. 45–62.

[15] J. B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., 11 (2001), pp. 796–817, https://doi.org/10.1137/S1052623400366802.

[16] J. B. Lasserre, *Convergent SDP-relaxations in polynomial optimization with sparsity*, SIAM J. Optim., 17 (2006), pp. 822–843, https://doi.org/10.1137/05064504X.

[17] J.-B. Lasserre, K.-C. Toh, and S. Yang, *A bounded degree SOS hierarchy for polynomial optimization*, EURO J. Comput. Optim., 5 (2017), pp. 87–117.

[18] M. Laurent, *Sums of squares, moment matrices and optimization over polynomials*, in Emerging Applications of Algebraic Geometry, IMA Vol. Math. Appl. 149, Springer, New York, 2009, pp. 157–270.

[19] J. Löfberg, *YALMIP: A toolbox for modeling and optimization in MATLAB*, in Proceedings of the IEEE International Conference on Robotics and Automation, 2004, pp. 284–289.

[20] J. Löfberg, *Pre- and post-processing sum-of-squares programs in practice*, IEEE Trans. Automat. Control, 54 (2009), pp. 1007–1011.

[21] V. Magron, *Interval enclosures of upper bounds of roundoff errors using semidefinite programming*, ACM Trans. Math. Software, 44 (2018), 41.

[22] V. Magron, G. Constantinides, and A. Donaldson, *Certified roundoff error bounds using semidefinite programming*, ACM Trans. Math. Software, 43 (2017), 34.

[23] A. Majumdar, A. A. Ahmadi, and R. Tedrake, *Control and verification of high-dimensional systems with DSOS and SDSOS programming*, in Proceedings of the 53rd IEEE Conference on Decision and Control, 2014, pp. 394–401.

[24] 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 Appl. Math., 275 (2020), pp. 95–110.

[25] A. Megretski, *Systems Polynomial Optimization Tools (SPOT)*, https://github.com/spot-toolbox/spotless, 2010.

[26] J. Miller, Y. Zheng, M. Sznaier, and A. Papachristodoulou, *Decomposed Structured Subsets for Semidefinite and Sum-of-Squares Optimization*, preprint, https://arxiv.org/

abs/1911.12859, 2019.
[27] MOSEK ApS, *The MOSEK Optimization Toolbox, Version* 8.1, https://docs.mosek.com/8.1/toolbox/index.html, 2017.
[28] 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 (2003), pp. 303–327.
[29] P. A. Parrilo, *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*, Ph.D. thesis, California Institute of Technology, Pasadena, CA, 2000.
[30] F. Permenter and P. A. Parrilo, *Basis selection for SOS programs via facial reduction and polyhedral approximations*, in Proceedings of the 53rd IEEE Conference on Decision and Control, 2014, pp. 6615–6620.
[31] F. Permenter and P. A. Parrilo, *Finding sparse, equivalent SDPs using minimal coordinate projections*, in Proceedings of the 54th IEEE Conference on Decision and Control, 2015, pp. 7274–7279.
[32] M. Putinar, *Positive polynomials on compact semialgebraic sets*, Indiana Univ. Math. J., 42 (1993), pp. 969–984.
[33] B. Reznick, *Extremal PSD forms with few terms*, Duke Math. J., 45 (1978), pp. 363–374.
[34] C. Riener, T. Theobald, L. J. Andrén, and J.-B. Lasserre, *Exploiting symmetries in SDP-relaxations for polynomial optimization*, Math. Oper. Res., 38 (2013), pp. 122–141.
[35] J. F. Sturm, *Using SeDuMi* 1.02*, a MATLAB toolbox for optimization over symmetric cones*, Optim. Methods Softw., 11 (1999), pp. 625–653.
[36] M. Tacchi, T. Weisser, J.-B. Lasserre, and D. Henrion, *Exploiting Sparsity for Semi-algebraic Set Volume Computation*, preprint, https://arxiv.org/abs/1902.02976, 2019.
[37] H. Waki, S. Kim, M. Kojima, and M. Muramatsu, *Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity*, SIAM J. Optim., 17 (2006), pp. 218–242, https://doi.org/10.1137/050623802.
[38] 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 (2009), 15.
[39] J. Wang, H. Li, and B. Xia, *A new sparse SOS decomposition algorithm based on term sparsity*, in Proceedings of the ACM International Symposium on Symbolic and Algebraic Computation, 2019, pp. 347–354.
[40] J. Wang, V. Magron, and J.-B. Lasserre, *Chordal-TSSOS: A moment-SOS hierarchy that exploits term sparsity with chordal extension*, SIAM J. Optim., to appear.
[41] J. Wang, V. Magron, J.-B. Lasserre, and N. H. A. Mai, *CS-TSSOS: Correlative and Term Sparsity for Large-Scale Polynomial Optimization*, preprint, https://arxiv.org/abs/2005.02828, 2020.
[42] T. Weisser, J. B. Lasserre, and K. C. Toh, *Sparse-BSOS: A bounded degree SOS hierarchy for large scale polynomial optimization with sparsity*, Math. Program. Comput., 10 (2018), pp. 1–32.