SIAM J. OPTIM.
Vol. 31, No. 1, pp. 114–141
© 2021 Society for Industrial and Applied Mathematics

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

CHORDAL-TSSOS: A MOMENT-SOS HIERARCHY THAT EXPLOITS TERM SPARSITY WITH CHORDAL EXTENSION*

JIE WANG†, VICTOR MAGRON†, AND JEAN-BERNARD LASSERRE†

Abstract. This work is a follow-up and a complement to [J. Wang, V. Magron and J. B. Lasserre, preprint, arXiv:1912.08899, 2019] where the TSSOS hierarchy was proposed for solving polynomial optimization problems (POPs). The chordal-TSSOS hierarchy that we propose is a new sparse moment-SOS framework based on *term sparsity* and *chordal extension*. By exploiting term sparsity of the input polynomials we obtain a two-level hierarchy of semidefinite programming relaxations. The novelty and distinguishing feature of such relaxations is to involve block matrices obtained in an iterative procedure that performs chordal extension of certain adjacency graphs. The graphs are related to the terms arising in the original data and *not* to the links between variables. Various numerical examples demonstrate the efficiency and the scalability of this new hierarchy for both unconstrained and constrained POPs. The two hierarchies are complementary. While the former TSSOS [J. Wang, V. Magron and J. B. Lasserre, preprint, arXiv:1912.08899, 2019] has a theoretical convergence guarantee (to the dense moment-SOS relaxation), the chordal-TSSOS has superior performance but lacks this theoretical guarantee.

Key words. sparsity pattern, polynomial optimization, moment relaxations, sum of squares, semidefinite programming

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

DOI. 10.1137/20M1323564

1. Introduction.

Consider the polynomial optimization problem (POP):

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

where $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}] = \mathbb{R}[x_1, \ldots, x_n]$ is a polynomial and $\mathbf{K} \subseteq \mathbb{R}^n$ is the basic semialgebraic set

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

for some polynomials $g_j(\mathbf{x}) \in \mathbb{R}[\mathbf{x}], j = 1, \ldots, m$. The so-called moment-SOS hierarchy [16] (where SOS stands for *sum of squares*) is a powerful approach based on certain specific positivity certificates of real algebraic geometry. It results in solving a hierarchy of semidefinite program (SDP) relaxations of (Q) whose associated monotone sequence of optimal values converges to f^* from below; in fact the convergence is even finite *generically* [25]. However, in view of the present status of SDP solvers, the moment-SOS hierarchy does not scale well and is so far limited to problems of modest size.

To address the issue of scalability, an important research direction is to define alternative relaxations of (Q) with cheaper computational cost and still with good convergence properties. One possibility is to define hierarchies of relaxations of (Q) based on other positivity certificates. Such alternative positivity certificates are in general weaker but their implementation is much easier as it results in LP-relaxations (e.g., as in DSOS [1]), second-order cone relaxations (e.g., as in SDSOS [1]), or cheaper SDP-relaxations (e.g., as in BSOS [18]).

*Received by the editors March 5, 2020; accepted for publication (in revised form) September 22, 2020; published electronically January 12, 2021.
https://doi.org/10.1137/20M1323564

†Laboratoire dAnalyse et dArchitecture des Systèmes (LAAS), Toulouse, France (jwang@laas.fr, vmagron@laas.fr, lasserre@laas.fr).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Another possibility is to address *sparsity* often present in the description of large-scale instances of (Q). One classical approach is to consider so-called *correlative sparsity* patterns, developed in [17, 34]. This is represented by the correlative sparsity pattern (csp) graph, which grasps the links between variables. Concretely, the nodes of a csp graph correspond to the variables and there is an edge between two nodes (variables) if and only if these two variables appear in the same term of the objective polynomial f or appear in the same polynomial g_j involved in the description of $\mathbf{K}$. One then partitions the variables into cliques according to the maximal cliques of a chordal extension of the csp graph to obtain a moment-SOS hierarchy for (Q) with block SDP matrices. If each maximal clique has a small size, this will significantly reduce the computational cost. This approach has been successfully applied for solving optimal powerflow problems [14], roundoff error bound analysis [21, 22], or more recently to approximate the volume of sparse semialgebraic sets [28], to bound Lipschitz constants of deep networks [5], and to represent sparse positive definite forms [20].

Nevertheless many POPs can be fairly sparse, but they do not exhibit a nontrivial csp (i.e., the corresponding csp graph is complete). For instance, if f has a term involving all variables or some constraint $g_j \geq 0$ (e.g., $1 - \|\mathbf{x}\|^2 \geq 0$) involves all variables, then the csp is trivial. Besides, even if a POP admits a nontrivial csp, some maximal cliques of the csp graph (after a chordal extension) may still have a large size (say, over 20), which makes the resulting SDP problem still hard to solve.

However, instead of exploiting sparsity from the perspective of *variables*, one can also exploit sparsity from the perspective of *terms* as described in [31, 32]. This is the route followed in this paper.

Novelty with respect to [32]. In [32] we exploited the term sparsity occurring in the description of (Q) to define a sparsity-adapted version of the moment-SOS hierarchy, which scales much better with the size of the initial problem (Q). Roughly speaking, the sparsity considered in [32] can be also represented by a graph, which is called a *term sparsity pattern (tsp) graph*. But unlike the csp graph, the nodes of a tsp graph correspond to monomials (not variables) and the edges of the graph grasp the links between monomials in the SOS representation of positive polynomials. In [32], we designed an iterative procedure to enlarge the tsp graph in order to iteratively exploit the term sparsity in (Q). Each iteration consists of two steps: (i) a support-extension operation and (ii) a block-closure operation on adjacency matrices.

We first propose to replace the second step from [32] by a chordal-extension operation. In doing so we obtain a sequence

$$G_1 \subseteq G_2 \subseteq \cdots \subseteq G_r$$

of graphs, where "$G_i \subseteq G_{i+1}$" means that G_i is a subgraph of G_{i+1}. The main difference with [32] is that (ii) now consists of performing an approximately minimum chordal extension instead of performing completion of the connected components for each graph. Then combining this iterative procedure with the standard moment-SOS hierarchy results in a two-level moment-SOS hierarchy with block SDP matrices. When the sizes of blocks are small, then the associated SDP relaxations are drastically much cheaper to solve.

To some extent, the term sparsity (focusing on monomials) is finer than the correlative sparsity (focusing on variables). If a POP is sparse in the sense of correlative sparsity (i.e., the csp graph is not complete), then it must be sparse in the sense of term sparsity (i.e., the tsp graph is not complete), while the converse is not necessarily

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

true. So the basic idea for solving large-scale POPs is as follows: first exploit correlative sparsity to obtain a coarse decomposition in terms of variables with cliques, and second exploit term sparsity for subsystems involving each clique of variables. This idea has been successfully carried out in [33].

Contribution. We propose a new sparse moment-SOS framework based on *term sparsity* and *chordal graphs*, following the route of our previous paper [32]. It is in deep contrast to the approach based on the sole correlative sparsity and provides a new item in the arsenal of sparsity-exploiting techniques for moment-SOS hierarchies of POPs. More precisely, note the following:

• We provide an iterative procedure that exploits term sparsity in POPs. The case of unconstrained polynomial optimization is treated in section 3 and the case of constrained polynomial optimization is treated in section 4. In the constrained case, it results in a two-level moment-SOS hierarchy that we call the *chordal-TSSOS hierarchy* (as the "TSSOS" terminology was used in our prior work [32]), which depends on two parameters: the relaxation order d and the sparse order k. The resulting SDP has *block* SDP matrices, which is the crucial feature of the chordal-TSSOS hierarchy. It is shown that the chordal-TSSOS hierarchy always provides tighter lower bounds than the SDSOS relaxation.

• We also give an algorithm in section 4 to reduce any initial monomial basis used in the chordal-TSSOS hierarchy as an optional preprocessing step, which applies to not only unconstrained POPs but also constrained POPs.

• In section 5 we provide a computational cost estimate for the first step (i.e., $k = 1$) of the chordal-TSSOS hierarchy via a careful analysis of the structure of tsp graphs.

• In section 6 we provide various numerical experiments to illustrate that POPs of significantly large size (up to 200 variables) and without correlative sparsity can be solved by our chordal-TSSOS hierarchy.

The chordal-TSSOS hierarchy should be considered as a *complement* to TSSOS [32] rather than just a *variant*. Indeed on the one hand, TSSOS has a *guarantee* that the optimal values converge to the same optimal value with the dense moment-SOS relaxation [32] and has good efficiency reported in [32] when compared to other hierarchies. On the other hand, while chordal-TSSOS lacks a theoretical convergence guarantee (to the dense moment-SOS relaxation), in practice it has superior performance and converges in many cases, as observed in the numerical experiments. Hence a user should start with chordal-TSSOS and possibly turns to TSSOS if convergence does not occur.

2. Notation and preliminaries.

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

For a finite set $\mathscr{A} \subseteq \mathbb{N}^n$, 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 $\mathbf{S}^r$ and the set of $r \times r$ positive semidefinite (PSD) (resp., positive definite) matrices is denoted by $\mathbf{S}^r_+$ (resp., $\mathbf{S}^r_{++}$). Let us denote by $\langle A, B\rangle \in \mathbb{R}$ the trace inner-product, defined by $\langle A, B\rangle = \mathrm{Tr}(A^T B)$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

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, an SOS decomposition of a given polynomial provides a certificate for its global nonnegativity. For $d \in \mathbb{N}$, let $\mathbb{N}^n_d := \{\boldsymbol{\alpha} = (\alpha_i) \in \mathbb{N}^n \mid \sum_{i=1}^n \alpha_i \le d\}$. Assume that $f \in \mathbb{R}[\mathbf{x}]$ is a polynomial of degree $2d$. 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* [6] for f, such that

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

When f is sparse (i.e., f contains only a few amount of monomials in $\mathbf{x}^{\mathbb{N}^n_{2d}}$), the size of Q can be reduced by computing a possibly 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

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

See [26] for a proof. We refer to this as the *Newton polytope method*. For convenience, we abuse notation in what follows and denote by $\mathscr{B}$ ($\boldsymbol{\beta}$) instead of $\mathbf{x}^{\mathscr{B}}$ ($\mathbf{x}^{\boldsymbol{\beta}}$) a monomial basis (a monomial).

Let $f(\mathbf{x}) = \sum_{\boldsymbol{\alpha} \in \mathscr{A}} f_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}}$ with $\mathrm{supp}(f) = \mathscr{A}$ and $\mathscr{B}$ be a monomial basis. For any $\boldsymbol{\alpha} \in \mathscr{B} + \mathscr{B} := \{\boldsymbol{\beta} + \boldsymbol{\gamma} \mid \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathscr{B}\}$, associate it with a $(0,1)$-binary matrix $A_{\boldsymbol{\alpha}} \in \mathbf{S}^{|\mathscr{B}|}$ such that $[A_{\boldsymbol{\alpha}}]_{\boldsymbol{\beta}\boldsymbol{\gamma}} = 1$ if and only if $\boldsymbol{\beta} + \boldsymbol{\gamma} = \boldsymbol{\alpha}$ for all $\boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathscr{B}$. Then f is an SOS polynomial if and only if there exists $Q \in \mathbf{S}^{|\mathscr{B}|}_+$ such that the following coefficient matching condition holds:

$$\langle A_{\boldsymbol{\alpha}}, Q \rangle = f_{\boldsymbol{\alpha}} \ \forall\ \boldsymbol{\alpha} \in \mathscr{B} + \mathscr{B}, \tag{2.4}$$

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

2.2. Moment-SOS relaxations for POPs. 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

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

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

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

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

Consider the unconstrained polynomial optimization problem:

$$(\mathrm{P}_0): \quad \lambda^* := \inf_{\mathbf{x}} \{f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\} \tag{2.5}$$

with $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ of degree $2d$. Let $\mathscr{B}$ be a monomial basis. Then the moment SDP

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

relaxation of (P_0) is [16]

$$(\mathrm{P}): \quad \begin{aligned} \lambda_{mom} := \ & \inf \ L_{\mathbf{y}}(f) \\ & \text{s.t.} \ M_{\mathscr{B}}(\mathbf{y}) \succeq 0, \\ & \quad y_{\mathbf{0}} = 1. \end{aligned} \tag{2.6}$$

The dual SDP problem of (2.6) is

$$(\mathrm{P})^*: \quad \begin{aligned} & \sup \ \lambda, \\ & \text{s.t.} \ \langle Q, A_{\boldsymbol{\alpha}} \rangle + \lambda \delta_{\mathbf{0}\boldsymbol{\alpha}} = f_{\boldsymbol{\alpha}} \quad \forall \boldsymbol{\alpha} \in \mathscr{B} + \mathscr{B}, \\ & \quad Q \succeq 0, \end{aligned} \tag{2.7}$$

where $\delta_{\mathbf{0}\boldsymbol{\alpha}}$ is the usual Kronecker symbol.

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 any positive integer d, the *localizing* matrix $M_d(g\mathbf{y})$ associated with g and $\mathbf{y}$ is the matrix with rows and columns indexed by $\mathbb{N}^n_d$ such that

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

Consider the constrained polynomial optimization problem:

$$(\mathrm{Q}_0): \quad \lambda^* := \inf_{\mathbf{x}} \{ f(\mathbf{x}) : \mathbf{x} \in \mathbf{K} \}, \tag{2.8}$$

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

$$\mathbf{K} = \{ \mathbf{x} \in \mathbb{R}^n : g_j(\mathbf{x}) \geq 0, j = 1, \ldots, m \} \tag{2.9}$$

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

Let $d_j := \lceil \deg(g_j)/2 \rceil, j = 1, \ldots, m$ and let $\hat{d} \geq \max\{\lceil \deg(f)/2 \rceil, d_1, \ldots, d_m\}$ be a positive integer. Then the Lasserre's hierarchy indexed by $\hat{d}$ of primal moment SDP relaxations of (Q_0) is defined by [16]:

$$(\mathrm{Q}_{\hat{d}}): \quad \begin{aligned} \lambda_{\hat{d}} := \ & \inf \ L_{\mathbf{y}}(f) \\ & \text{s.t.} \ M_{\hat{d}}(\mathbf{y}) \succeq 0, \\ & \quad M_{\hat{d}-d_j}(g_j \mathbf{y}) \succeq 0, \quad j = 1, \ldots, m, \\ & \quad y_{\mathbf{0}} = 1. \end{aligned} \tag{2.10}$$

We call $\hat{d}$ the *relaxation order*.

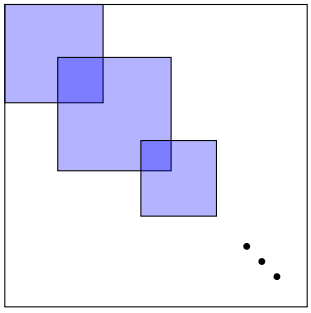
Set $g_0 := 1$ and $d_0 := 0$. For each j, writing $M_{\hat{d}-d_j}(g_j \mathbf{y}) = \sum_{\boldsymbol{\alpha}} D^j_{\boldsymbol{\alpha}} y_{\boldsymbol{\alpha}}$ for appropriate symmetric matrices $\{D^j_{\boldsymbol{\alpha}}\}$, we can write the dual of (2.10) as

$$(\mathrm{Q}_{\hat{d}})^*: \quad \begin{aligned} & \sup \ \lambda, \\ & \text{s.t.} \ \sum_{j=0}^{m} \langle Q_j, D^j_{\boldsymbol{\alpha}} \rangle + \lambda \delta_{\mathbf{0}\boldsymbol{\alpha}} = f_{\boldsymbol{\alpha}} \quad \forall \boldsymbol{\alpha} \in \mathbb{N}^n_{2\hat{d}}, \\ & \quad Q_j \succeq 0, \quad j = 0, \ldots, m. \end{aligned} \tag{2.11}$$

2.3. Chordal graphs and sparse matrices. We introduce some basic notions from graph theory. An (undirected) *graph* $G(V, E)$ or simply G consists of a set of nodes V and a set of edges $E \subseteq \{\{v_i, v_j\} \mid (v_i, v_j) \in V \times V\}$. If G is a graph, we also use $V(G)$ and $E(G)$ to indicate the set of nodes of G and the set of edges of G,

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



The blue area indicates the positions of possible nonzero entries.

FIG. 1. *The block structure of matrices in* $\mathbf{S}_G$.

respectively. For two graphs G, H, we say that G is a *subgraph* of H if $V(G) \subseteq V(H)$ and $E(G) \subseteq E(H)$, denoted by $G \subseteq H$. For a graph $G(V, E)$, a *cycle* of length k is a set of nodes $\{v_1, v_2, \ldots, v_k\} \subseteq V$ with $\{v_k, v_1\} \in E$ and $\{v_i, v_{i+1}\} \in E$ for $i = 1, \ldots, k-1$. A *chord* in a cycle $\{v_1, v_2, \ldots, v_k\}$ is an edge $\{v_i, v_j\}$ that joins two nonconsecutive nodes in the cycle.

A graph is called a *chordal graph* if all its cycles of length at least four have a chord. Chordal graphs include some common classes of graphs, such as complete graphs, line graphs, and trees, and have applications in sparse matrix theory. Note that any nonchordal graph $G(V, E)$ can always be extended to a chordal graph $\overline{G}(V, \overline{E})$ by adding appropriate edges to E, which is called a *chordal extension* of $G(V, E)$. A *clique* $C \subseteq V$ of G is a subset of nodes where $\{v_i, v_j\} \in E$ for any $v_i, v_j \in C$. If a clique C is not a subset of any other clique, then it is called a *maximal clique*. It is known that maximal cliques of a chordal graph can be enumerated efficiently in linear time in the number of nodes and edges of the graph. See, e.g., [3, 8, 10] for details.

Given a graph $G(V, E)$, a symmetric matrix Q with row and column indices labeled by V is said to have sparsity pattern G if $Q_{\boldsymbol{\beta}\boldsymbol{\gamma}} = Q_{\boldsymbol{\gamma}\boldsymbol{\beta}} = 0$ whenever $\boldsymbol{\beta} \neq \boldsymbol{\gamma}$ and $\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \notin E$. Let $\mathbf{S}_G$ be the set of symmetric matrices with sparsity pattern G. A matrix in $\mathbf{S}_G$ exhibits a *block* structure (after an appropriate permutation of rows and columns) as illustrated in Figure 1. Each block corresponds to a maximal clique of G. The maximal block size is the maximal size of maximal cliques of G, namely, the *clique number* of G. Note that there might be overlaps between blocks because different maximal cliques may share nodes.

Given a maximal clique C of $G(V, E)$, we define a matrix $P_C \in \mathbb{R}^{|C| \times |V|}$ as

$$[P_C]_{i\boldsymbol{\beta}} = \begin{cases} 1 & \text{if } C(i) = \boldsymbol{\beta}, \\ 0 & \text{otherwise}, \end{cases} \tag{2.12}$$

where $C(i)$ denotes the ith node in C, sorted in the ordering compatibly with V. Note that $Q_C = P_C Q P_C^T \in \mathbf{S}^{|C|}$ extracts a principal submatrix Q_C defined by the indices in the clique C from a symmetry matrix Q, and $Q = P_C^T Q_C P_C$ inflates a $|C| \times |C|$ matrix Q_C into a sparse $|V| \times |V|$ matrix Q.

The PSD matrices with sparsity pattern G form a convex cone

$$\mathbf{S}_+^{|V|} \cap \mathbf{S}_G = \{Q \in \mathbf{S}_G \mid Q \succeq 0\}. \tag{2.13}$$

When the sparsity pattern graph G is chordal, the cone $\mathbf{S}_+^{|V|} \cap \mathbf{S}_G$ can be decomposed as a sum of simple convex cones, as stated in the following theorem.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

THEOREM 2.1 (see [30, Theorem 9.2]). *Let $G(V,E)$ be a chordal graph and assume that $C_1,\dots,C_t$ are all the maximal cliques of $G(V,E)$. Then a matrix $Q \in \mathbf{S}_+^{|V|} \cap \mathbf{S}_G$ if and only if there exist $Q_k \in \mathbf{S}_+^{|C_k|}$ for $k=1,\dots,t$ such that $Q = \sum_{k=1}^t P_{C_k}^T Q_k P_{C_k}$.*

Given a graph $G(V,E)$, let Π_G be the projection from $\mathbf{S}^{|V|}$ to the subspace $\mathbf{S}_G$, i.e., for $Q \in \mathbf{S}^{|V|}$,

$$\Pi_G(Q)_{\boldsymbol{\beta}\boldsymbol{\gamma}} = \begin{cases} Q_{\boldsymbol{\beta}\boldsymbol{\gamma}} & \text{if } \{\boldsymbol{\beta},\boldsymbol{\gamma}\} \in E \text{ or } \boldsymbol{\beta}=\boldsymbol{\gamma}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.14}$$

We denote by $\Pi_G(\mathbf{S}_+^{|V|})$ the set of matrices in $\mathbf{S}_G$ that have a PSD completion, i.e.,

$$\Pi_G(\mathbf{S}_+^{|V|}) = \{\Pi_G(Q) \mid Q \in \mathbf{S}_+^{|V|}\}. \tag{2.15}$$

One can check that the PSD completable cone $\Pi_G(\mathbf{S}_+^{|V|})$ and the PSD cone $\mathbf{S}_+^{|V|} \cap \mathbf{S}_G$ form a pair of dual cones in $\mathbf{S}_G$; see [30, section 10.1] for a proof. Moreover, for a chordal graph G, the decomposition result for the cone $\mathbf{S}_+^{|V|} \cap \mathbf{S}_G$ in Theorem 2.1 leads to the following characterization of the PSD completable cone $\Pi_G(\mathbf{S}_+^{|V|})$.

THEOREM 2.2 (see [30, Theorem 10.1]). *Let $G(V,E)$ be a chordal graph and assume that $C_1,\dots,C_t$ are all the maximal cliques of $G(V,E)$. Then a matrix $Q \in \Pi_G(\mathbf{S}_+^{|V|})$ if and only if $Q_k = P_{C_k} Q P_{C_k}^T \succeq 0$ for $k=1,\dots,t$. Moreover, a matrix $Q \in \Pi_G(\mathbf{S}_+^{|V|})$ if and only if $Q_k = P_{C_k} Q P_{C_k}^T \succ 0$ for $k=1,\dots,t$.*

For more details about sparse matrices and chordal graphs, the reader may refer to [30].

3. The chordal-TSSOS hierarchy: Unconstrained case. In this section, we describe an iterative procedure to exploit term sparsity for the primal (2.6) and dual (2.7) SDP relaxations of unconstrained POPs.

Let $f(\mathbf{x}) = \sum_{\boldsymbol{\alpha}\in\mathscr{A}} f_{\boldsymbol{\alpha}} \mathbf{x}^{\boldsymbol{\alpha}} \in \mathbb{R}[\mathbf{x}]$ with $\operatorname{supp}(f) = \mathscr{A}$ (without loss of generality assuming $\mathbf{0} \in \mathscr{A}$) and $\mathscr{B}$ be a monomial basis. In the following, we will consider graphs with $V := \mathscr{B}$ as the set of nodes. Suppose that $G(V,E)$ is such a graph. We define the *support* of G by

$$\operatorname{supp}(G) := \{\boldsymbol{\beta}+\boldsymbol{\gamma} \mid (\boldsymbol{\beta},\boldsymbol{\gamma}) \in V\times V,\ \{\boldsymbol{\beta},\boldsymbol{\gamma}\} \in E\}.$$

We further define two operations on G: *support extension* and *chordal extension*. The support extension of G, denoted by $\operatorname{SE}(G)$, is the graph with nodes $\mathscr{B}$ and with edges

$$E(\operatorname{SE}(G)) := \{\{\boldsymbol{\beta},\boldsymbol{\gamma}\} \mid (\boldsymbol{\beta},\boldsymbol{\gamma}) \in V\times V,\ \boldsymbol{\beta}\neq\boldsymbol{\gamma},\ \boldsymbol{\beta}+\boldsymbol{\gamma} \in \operatorname{supp}(G)\}.$$

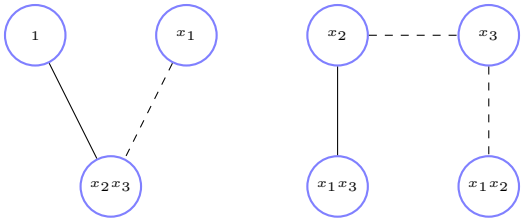
Example 3.1. Consider the graph $G(V,E)$ with

$$V = \{1, x_1, x_2, x_3, x_2x_3, x_1x_3, x_1x_2\} \text{ and } E = \{\{1, x_2x_3\}, \{x_2, x_1x_3\}\}.$$

Then $E(\operatorname{SE}(G)) = \{\{1,x_2x_3\},\{x_2,x_1x_3\},\{x_2,x_3\},\{x_1,x_2x_3\},\{x_3,x_1x_2\}\}$. See Figure 2 for the support extension $\operatorname{SE}(G)$ of G.

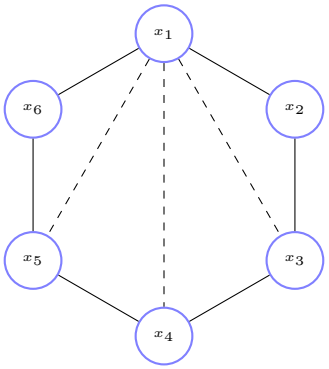
Any specific chordal extension of G is denoted by $\overline{G}$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.



The dashed edges are added after support extension.

FIG. 2. *The support extension* $\mathrm{SE}(G)$ *of* G.



The dashed edges are added after chordal extension.

FIG. 3. *The chordal extension* $\overline{G}$ *of* G.

Example 3.2. Consider the graph $G(V,E)$ with $V = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ and $E = \{\{x_1, x_2\}, \{x_2, x_3\}, \{x_3, x_4\}, \{x_4, x_5\}, \{x_5, x_6\}, \{x_6, x_1\}\}$. See Figure 3 for the chordal extension $\overline{G}$ of G.

Remark 3.3. For a graph $G(V,E)$, the chordal extension of G is usually not unique. A chordal extension with the least number of edges is called a *minimum chordal extension*. Finding a minimum chordal extension of a graph is an NP-complete problem in general. Fortunately, several heuristic algorithms, such as the minimum degree ordering, are known to efficiently produce a good approximation [2, 11]. In this paper, we always use an approximately minimum chordal extension for a graph.

In what follows, we assume that for graphs G, H with the same set of nodes, if $E(G) \subseteq E(H)$, then $E(\overline{G}) \subseteq E(\overline{H})$. This assumption is reasonable since any chordal extension of H must be also a chordal extension of G.

Let $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$ with $\mathrm{supp}(f) = \mathscr{A}$ (without loss of generality assuming $\mathbf{0} \in \mathscr{A}$) and $\mathscr{B}$ be a monomial basis with $r = |\mathscr{B}|$. We define $G_0(V, E_0)$ to be the graph with $V = \mathscr{B}$ and

$$E_0 = \{\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \mid (\boldsymbol{\beta}, \boldsymbol{\gamma}) \in V \times V,\ \boldsymbol{\beta} \neq \boldsymbol{\gamma},\ \boldsymbol{\beta} + \boldsymbol{\gamma} \in \mathscr{A} \cup (2\mathscr{B})\}, \tag{3.1}$$

where $2\mathscr{B} = \{2\boldsymbol{\beta} \mid \boldsymbol{\beta} \in \mathscr{B}\}$. We call G_0 the *tsp graph* associated with f.

For $k \geq 1$, we recursively define a sequence of graphs $(G_k(V, E_k))_{k\geq 1}$ by

$$G_k := \overline{\mathrm{SE}(G_{k-1})}. \tag{3.2}$$

If f is sparse, by replacing $M_{\mathscr{B}}(\mathbf{y}) \succeq 0$ with the weaker condition $M_{\mathscr{B}}(\mathbf{y}) \in \Pi_{G_k}(\mathbf{S}^r_+)$ in (2.6), we obtain a sparse moment SDP relaxation of (P_0) (2.5) for each

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

$k \geq 1$:

$$(\mathrm{P}^k): \quad \lambda_k := \begin{array}{ll} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & M_{\mathscr{B}}(\mathbf{y}) \in \Pi_{G_k}(\mathbf{S}^r_+), \\ & y_{\mathbf{0}} = 1. \end{array} \tag{3.3}$$

We call k the *sparse order*. By construction, one has $G_k \subseteq G_{k+1}$ for all $k \geq 1$ and therefore the sequence of graphs $(G_k(V, E_k))_{k\geq 1}$ stabilizes after a finite number of steps.

Remark 3.4. The intuition behind the support-extension operation is that once one position related to $y_{\boldsymbol{\alpha}}$ in the moment matrix $M_{\mathscr{B}}(\mathbf{y})$ is "activated" in the sparsity pattern, then all positions related to $y_{\boldsymbol{\alpha}}$ in $M_{\mathscr{B}}(\mathbf{y})$ should be "activated." Theorems 2.1 and 2.2 provide the rationale behind the mechanism of the chordal-extension operation.

THEOREM 3.5. *The sequence* $(\lambda_k)_{k\geq 1}$ *is monotone nondecreasing and* $\lambda_k \leq \lambda_{mom}$ *for all* k.

Proof. Because $G_k \subseteq G_{k+1}$, each maximal clique of G_k is a subset of some maximal clique of G_{k+1}. Thus by Theorem 2.2, we have that (P^k) is a relaxation of (P^{k+1}) (and also a relaxation of (P)). This yields the desired conclusions. ∎

As a consequence of Theorem 3.5, we obtain the following hierarchy of lower bounds for the optimum of the original problem (P_0):

$$\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{mom} \leq \lambda^*. \tag{3.4}$$

We say that (3.3) (and its associated sequence (3.4)) is the *chordal*-TSSOS hierarchy for (P_0).

The *maximal* chordal extension of a graph is the one that completes every connected component of the graph. If we use the maximal chordal extension in (3.2), then we retrieve the TSSOS hierarchy developed in [32] (which will be referred to as the block-TSSOS hierarchy in this paper). It was shown in [32] that the sequence of optima of the block-TSSOS hierarchy always converges to the optimum of the dense moment-SOS relaxation.

Unlike the block-TSSOS hierarchy, in theory there is no guarantee that the chordal-TSSOS hierarchy of lower bounds $(\lambda_k)_{k\geq 1}$ converges to the value λ_{mom}. The following is an example.

Example 3.6. Consider the polynomial $f = x_1^2 - 2x_1x_2 + 3x_2^2 - 2x_1^2x_2 + 2x_1^2x_2^2 - 2x_2x_3 + 6x_3^2 + 18x_2^2x_3 - 54x_2x_3^2 + 142x_2^2x_3^2$ [15]. The monomial basis computed by the Newton polytope method is $\{1, x_1, x_2, x_3, x_1x_2, x_2x_3\}$. We have $E_0 = \{\{1, x_1x_2\}, \{1, x_2x_3\}, \{x_1, x_1x_2\}, \{x_1, x_2\}, \{x_2, x_3\}, \{x_2, x_2x_3\}, \{x_3, x_2x_3\}\}$. Figure 4 shows the tsp graph G_0 (without dashed edges) and its chordal extension G_1 (with dashed edges) for f. The graph sequence $(G_k)_{k\geq 1}$ stabilizes at $k = 1$. Solving the SDP problem (P^1) associated with G_1, we obtain $\lambda_1 \approx -0.00355$ while we have $\lambda_{mom} = \lambda^* = 0$.

Remark 3.7. Even though there is no theoretical guarantee that the sequence of optimal values of the chordal-TSSOS hierarchy converges to the optimal value of the dense moment-SOS relaxation for (P_0), in practice the convergence takes place in many cases as we shall see in the numerical experiments.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

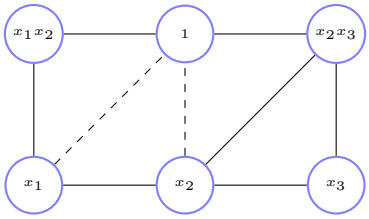


FIG. 4. *The tsp graph G_0 and its chordal extension G_1 for Example* 3.6.

For each $k \geq 1$, the dual SDP problem of (3.3) is

$$(\mathrm{P}^k)^* : \quad \begin{array}{ll} \sup & \lambda, \\ \text{s.t.} & \langle Q, A_{\boldsymbol{\alpha}} \rangle + \lambda \delta_{\mathbf{0}\boldsymbol{\alpha}} = f_{\boldsymbol{\alpha}} \quad \forall \boldsymbol{\alpha} \in \mathrm{supp}(G_k) \cup (2\mathscr{B}), \\ & Q \in \mathbf{S}^r_+ \cap \mathbf{S}_{G_k}, \end{array} \tag{3.5}$$

where $A_{\boldsymbol{\alpha}}$ is defined by (2.4).

PROPOSITION 3.8. *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 [16] for the dense case and Theorem 2.2. ∎

Comparison with SDSOS [1]. The following definition of SDSOS polynomials has been introduced and studied in [1]. A symmetric matrix $Q \in \mathbf{S}^r$ is *diagonally dominant* if $Q_{ii} \geq \sum_{j \neq i} |Q_{ij}|$ for $i = 1, \ldots, r$ and is *scaled diagonally dominant* if there exists a positive definite $r \times r$ diagonal matrix D such that DQD 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 $SDSOS$.

Following [1], by replacing the nonnegativity condition in (P_0) with the SDSOS condition, one obtains the SDSOS relaxation of (P) and (P_0):

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

THEOREM 3.9. *With the above notation, one has $\lambda_1 \geq \lambda_{sdsos}$.*

Proof. Let $f \in \mathbb{R}[\mathbf{x}]$ with $\mathscr{A} = \mathrm{supp}(f)$ and $\mathscr{B}$ be a monomial basis with $r = |\mathscr{B}|$. Assume that $f \in \mathrm{SDSOS}$, i.e., f admits a scaled diagonally dominant Gram matrix $Q \in \mathbb{S}^r_+$ indexed by $\mathscr{B}$. We then construct a Gram matrix $\tilde{Q}$ for f by

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

It is easy to see that we still have $f = (\mathbf{x}^{\mathscr{B}})^T \tilde{Q} \mathbf{x}^{\mathscr{B}}$. Note that we only replace off-diagonal entries by zeros in Q to obtain $\tilde{Q}$ 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 $\tilde{Q} \in \mathbf{S}^r_+ \cap \mathbf{S}_{G_1}$ by construction. It follows that (SDSOS) is a relaxation of $(\mathrm{P}^1)^*$. Hence $\lambda_1 \geq \lambda_{sdsos}$. ∎

The next result states that $\lambda_1 = \lambda_{mom}$ always holds in the quadratic case.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

THEOREM 3.10. *Suppose that the objective function $f \in \mathbb{R}[\mathbf{x}]$ in (P_0) is a quadratic polynomial. Then $\lambda_1 = \lambda_{mom}$.*

Proof. Assume that $\mathrm{supp}(f) = \mathscr{A}$. Since f is quadratic, we may take $\mathscr{B} = \{\mathbf{e}_0, \mathbf{e}_1, \ldots, \mathbf{e}_n\}$ as a monomial basis, where $\mathbf{e}_0 = \mathbf{0}$ and $\{\mathbf{e}_i\}_{i=1}^n$ is the standard basis of $\mathbb{R}^n$. Let G_0 be the tsp graph associated with f. We only need to prove that if f admits a PSD Gram matrix, then f admits a Gram matrix in $\mathbf{S}_+^{n+1} \cap \mathbf{S}_{G_0}$. Suppose that $Q = [q_{ij}]_{i,j=0}^n$ is a PSD Gram matrix for f indexed by $\mathscr{B}$. Note that for all i, j, if $\{\mathbf{e}_i, \mathbf{e}_j\} \notin E(G_0)$, then we must have $\mathbf{e}_i + \mathbf{e}_j \notin \mathscr{A}$, which implies $q_{ij} = 0$. It follows that $Q \in \mathbf{S}_{G_0}$ as desired. ∎

4. The chordal-TSSOS hierarchy: Constrained case. In this section, we describe an iterative procedure to exploit term sparsity for the primal-dual moment-SOS hierarchy (2.10)–(2.11) of the constrained POP (Q_0) defined in (2.8)–(2.9). Let

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

Let $d_j := \lceil \deg(g_j)/2 \rceil, j = 1, \ldots, m$ and $d := \max\{\lceil \deg(f)/2 \rceil, d_1, \ldots, d_m\}$. Fix a relaxation order $\hat{d} \geq d$ in Lasserre's hierarchy (2.10). Let $g_0 = 1$, $d_0 = 0$, and $\mathscr{B}_{j,\hat{d}} = \mathbb{N}^n_{\hat{d}-d_j}$ be the standard monomial basis for $j = 0, \ldots, m$. We define a graph $G^{(0)}_{0,\hat{d}}(V_{0,\hat{d}}, E^{(0)}_{0,\hat{d}})$ with $V_{0,\hat{d}} = \mathscr{B}_{0,\hat{d}}$ and

$$E^{(0)}_{0,\hat{d}} = \{\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \mid (\boldsymbol{\beta}, \boldsymbol{\gamma}) \in V_{0,\hat{d}} \times V_{0,\hat{d}},\ \boldsymbol{\beta} \neq \boldsymbol{\gamma},\ \boldsymbol{\beta} + \boldsymbol{\gamma} \in \mathscr{A} \cup (2\mathscr{B}_{0,\hat{d}})\}. \tag{4.1}$$

We call G_0 the *tsp graph* associated with (Q_0).

For $k \geq 1$, we recursively define a sequence of graphs $(G^{(k)}_{j,\hat{d}}(V_{j,\hat{d}}, E^{(k)}_{j,\hat{d}}))_{k\geq 1}$ with $V_{j,\hat{d}} = \mathscr{B}_{j,\hat{d}}$ for $j = 0, \ldots, m$ by

$$G^{(k)}_{0,\hat{d}} := \overline{\mathrm{SE}(G^{(k-1)}_{0,\hat{d}})} \text{ and } G^{(k)}_{j,\hat{d}} := \overline{F^{(k)}_{j,\hat{d}}}, \quad j = 1, \ldots, m, \tag{4.2}$$

where $F^{(k)}_{j,\hat{d}}$ is the graph with $V(F^{(k)}_{j,\hat{d}}) = \mathscr{B}_{j,\hat{d}}$ and

$$\begin{aligned} E(F^{(k)}_{j,\hat{d}}) = &\{\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \mid (\boldsymbol{\beta}, \boldsymbol{\gamma}) \in \mathscr{B}_{j,\hat{d}} \times \mathscr{B}_{j,\hat{d}},\ \boldsymbol{\beta} \neq \boldsymbol{\gamma},\\ &(\mathrm{supp}(g_j) + \boldsymbol{\beta} + \boldsymbol{\gamma}) \cap (\mathrm{supp}(G^{(k-1)}_{0,\hat{d}}) \cup (2\mathscr{B}_{0,\hat{d}})) \neq \emptyset\}, \quad j = 1, \ldots, m. \end{aligned} \tag{4.3}$$

Let $r_j := \binom{n+\hat{d}-d_j}{\hat{d}-d_j}$. Therefore by replacing $M_{\hat{d}-d_j}(g_j\mathbf{y}) \succeq 0$ with the weaker condition $M_{\hat{d}-d_j}(g_j\mathbf{y}) \in \Pi_{G^{(k)}_{j,\hat{d}}}(\mathbf{S}_+^{r_j})$ for $j = 0, \ldots, m$ in (2.10), we obtain the following sparse SDP relaxation of $(\mathrm{Q}_{\hat{d}})$ and (Q_0) for each $k \geq 1$:

$$(\mathrm{Q}^k_{\hat{d}}): \quad \begin{aligned} \lambda^{(k)}_{\hat{d}} := \ & \inf \quad L_{\mathbf{y}}(f),\\ & \text{s.t.} \quad M_{\hat{d}}(\mathbf{y}) \in \Pi_{G^{(k)}_{0,\hat{d}}}(\mathbf{S}_+^{r_0}),\\ & \qquad M_{\hat{d}-d_j}(g_j\mathbf{y}) \in \Pi_{G^{(k)}_{j,\hat{d}}}(\mathbf{S}_+^{r_j}), \quad j = 1, \ldots, m,\\ & \qquad y_{\mathbf{0}} = 1. \end{aligned} \tag{4.4}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

We call k the *sparse order*. By construction, one has $G_{j,\hat{d}}^{(k)} \subseteq G_{j,\hat{d}}^{(k+1)}$ for all j,k. Therefore, for every j, the sequence of graphs $(G_{j,\hat{d}}^{(k)})_{k\geq 1}$ stabilizes after a finite number of steps.

THEOREM 4.1. *Fixing a relaxation order $\hat{d} \geq d$, the sequence $(\lambda_{\hat{d}}^{(k)})_{k\geq 1}$ is monotone nondecreasing and $\lambda_{\hat{d}}^{(k)} \leq \lambda_{\hat{d}}$ for all k (with $\lambda_{\hat{d}}$ as in (2.10)).*

Proof. For all j,k, because $G_{j,\hat{d}}^{(k)} \subseteq G_{j,\hat{d}}^{(k+1)}$, each maximal clique of $G_{j,\hat{d}}^{(k)}$ is a subset of some maximal clique of $G_{j,\hat{d}}^{(k+1)}$. Hence by Theorem 2.2, $(\mathrm{Q}_{\hat{d}}^{k})$ is a relaxation of $(\mathrm{Q}_{\hat{d}}^{k+1})$ (and also a relaxation of $(\mathrm{Q}_{\hat{d}})$). Therefore, $(\lambda_{\hat{d}}^{(k)})_{k\geq 1}$ is monotone nondecreasing and $\lambda_{\hat{d}}^{(k)} \leq \lambda_{\hat{d}}$ for all k. ∎

THEOREM 4.2. *Fixing a sparse order $k \geq 1$, the sequence $(\lambda_{\hat{d}}^{(k)})_{\hat{d}\geq d}$ is monotone nondecreasing.*

Proof. The conclusion follows if we can show that $G_{j,\hat{d}}^{(k)} \subseteq G_{j,\hat{d}+1}^{(k)}$ for all $j,\hat{d}$ since by Theorem 2.2 this implies that $(\mathrm{Q}_{\hat{d}}^{k})$ is a relaxation of $(\mathrm{Q}_{\hat{d}+1}^{k})$. Let us prove $G_{j,\hat{d}}^{(k)} \subseteq G_{j,\hat{d}+1}^{(k)}$ by induction on k. For $k=1$, from (4.1), we have $E_{0,\hat{d}}^{(0)} \subseteq E_{0,\hat{d}+1}^{(0)}$, which implies that $G_{j,\hat{d}}^{(1)} \subseteq G_{j,\hat{d}+1}^{(1)}$ for $j=0,\ldots,m$. Now assume that $G_{j,\hat{d}}^{(k)} \subseteq G_{j,\hat{d}+1}^{(k)}$, $j=0,\ldots,m$ hold for a given $k \geq 1$. Then from (4.2) and (4.3) and by the induction hypothesis, we have $G_{j,\hat{d}}^{(k+1)} \subseteq G_{j,\hat{d}+1}^{(k+1)}$ for $j=0,\ldots,m$, which completes the induction and also completes the proof. ∎

Combining Theorem 4.1 with Theorem 4.2, we have the following two-level hierarchy of lower bounds for the optimum of (Q_0):

$$
\begin{array}{ccccccc}
\lambda_d^{(1)} & \leq & \lambda_d^{(2)} & \leq & \cdots & \leq & \lambda_d \\
\wedge| & & \wedge| & & & & \wedge| \\
\lambda_{d+1}^{(1)} & \leq & \lambda_{d+1}^{(2)} & \leq & \cdots & \leq & \lambda_{d+1} \\
\wedge| & & \wedge| & & & & \wedge| \\
\vdots & & \vdots & & \vdots & & \vdots \\
\wedge| & & \wedge| & & & & \wedge| \\
\lambda_{\hat{d}}^{(1)} & \leq & \lambda_{\hat{d}}^{(2)} & \leq & \cdots & \leq & \lambda_{\hat{d}} \\
\wedge| & & \wedge| & & & & \wedge| \\
\vdots & & \vdots & & \vdots & & \vdots
\end{array}
\tag{4.5}
$$

The array of lower bounds (4.5) (and its associated SDP-relaxations (4.4)) is what we call the *chordal*-TSSOS moment-SOS hierarchy (in short *chordal*-TSSOS hierarchy) associated with (Q_0).

For each $k \geq 1$, the dual of $(\mathrm{Q}_{\hat{d}}^{k})$ reads as

$$
(\mathrm{Q}_{\hat{d}}^{k})^{*}: \quad
\begin{array}{ll}
\sup & \lambda, \\
\text{s.t.} & \sum_{j=0}^{m} \langle Q_j, D_{\boldsymbol{\alpha}}^{j} \rangle + \lambda \delta_{\mathbf{0}\boldsymbol{\alpha}} = f_{\boldsymbol{\alpha}}, \\
& \qquad \forall \boldsymbol{\alpha} \in \bigcup_{j=0}^{m} (\mathrm{supp}(g_j) + \mathrm{supp}(G_{j,\hat{d}}^{(k)})) \cup (2\mathscr{B}_{0,\hat{d}}), \\
& Q_j \in \mathbf{S}_{+}^{r_j} \cap \mathbf{S}_{G_{j,\hat{d}}^{(k)}}, \quad j=0,\ldots,m,
\end{array}
\tag{4.6}
$$

where $D_{\boldsymbol{\alpha}}^{j}$ is defined in section 2.1.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

PROPOSITION 4.3. *Let $f \in \mathbb{R}[\mathbf{x}]$ and $\mathbf{K}$ be as in (2.9). Assume that K has a nonempty interior. Then there is no duality gap between $(Q_{\hat{d}}^k)$ and $(Q_{\hat{d}}^k)^*$ 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 [16] for the dense case and Theorem 2.2. ∎

Remark 4.4. As in the unconstrained case, there is no theoretical guarantee that the sequence of optimal values $(\lambda_{\hat{d}}^{(k)})_{k\geq 1}$ of the chordal-TSSOS hierarchy converges to the optimal value $\lambda_{\hat{d}}$ of the dense moment-SOS relaxation for (Q_0). However, as we shall observe in the numerical experiments, the convergence takes place in many cases.

By the same argument as for Theorem 3.10, we can prove the following.

PROPOSITION 4.5. *For quadratically constrained quadratic programs (QCQP), the equality $\lambda_1^{(1)} = \lambda_1$ holds.*

Remark 4.6. Our above treatment easily extends to include equality constraints.

Obtaining a possibly smaller monomial basis $\mathscr{B}$ ($\mathscr{B}_{0,\hat{d}}$). The chordal-TSSOS hierarchy depends on the chosen monomial basis $\mathscr{B}$ ($\mathscr{B}_{0,\hat{d}}$) and therefore its choice can affect the overall efficiency of the hierarchy. For instance, for unconstrained POPs, the Newton polytope method usually provides a monomial basis smaller than the standard monomial basis. However, this method does not apply to constrained POPs. Here as an optional pretreatment of POPs, we provide an iterative procedure which not only enables us to obtain a monomial basis $\mathscr{B}$ smaller than the one given by the Newton polytope method for unconstrained POPs in many cases, but can also be applied to constrained POPs. It sometimes leads to a monomial basis $\mathscr{B}_{0,\hat{d}}$ smaller than the standard one.

We start with the unconstrained case. Let $f \in \mathbb{R}[\mathbf{x}]$ with $\mathscr{A} = \mathrm{supp}(f)$ and $\mathscr{B}$ the monomial basis given by the Newton polytope method. Set $\mathscr{B}_0 := \emptyset$. For $p \geq 1$, we iteratively define a sequence of monomial sets $(\mathscr{B}_p)_{p\geq 1}$ by

$$\mathscr{B}_p := \{\boldsymbol{\beta} \in \mathscr{B} \mid \exists \boldsymbol{\gamma} \in \mathscr{B} \text{ s.t. } \boldsymbol{\beta} + \boldsymbol{\gamma} \in \mathscr{A} \cup (2\mathscr{B}_{p-1})\}. \tag{4.7}$$

Consequently, we obtain an increasing chain of monomial sets:

$$\mathscr{B}_1 \subseteq \mathscr{B}_2 \subseteq \mathscr{B}_3 \subseteq \cdots \subseteq \mathscr{B}.$$

Clearly, the above chain will stabilize in a finite number of steps. Each $\mathscr{B}_p$ can serve as a candidate monomial basis. Particularly, we have the following.

PROPOSITION 4.7. *Let $f \in \mathbb{R}[\mathbf{x}]$ and $\mathscr{B}_* = \cup_{p\geq 1}\mathscr{B}_p$. If $f \in$ SDSOS, then f is an SDSOS polynomial in the monomial basis $\mathscr{B}_*$.*

Proof. Let $\mathscr{B}$ be the monomial basis given by the Newton polytope method with $r = |\mathscr{B}|$. If $f \in$ SDSOS, then there exists a scaled diagonally dominant Gram matrix $Q \in \mathbb{S}_+^r$ indexed by $\mathscr{B}$ such that $f = (\mathbf{x}^{\mathscr{B}})^T Q \mathbf{x}^{\mathscr{B}}$. Let $s = |\mathscr{B}_*|$. We then construct a Gram matrix $\tilde{Q} \in \mathbf{S}_+^s$ indexed by $\mathscr{B}_*$ for f as follows:

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

One can easily check that we still have $f = (\mathbf{x}^{\mathscr{B}_*})^T \tilde{Q} \mathbf{x}^{\mathscr{B}_*}$. Let $\hat{Q}$ be the principal submatrix of Q by deleting the rows and columns whose indices are not in $\mathscr{B}_*$, which

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

is also a scaled diagonally dominant matrix. By construction, $\tilde{Q}$ is obtained from $\hat{Q}$ by replacing certain off-diagonal entries by zeros. Since replacing off-diagonal entries by zeros does not affect the scaled diagonal dominance of a matrix, $\tilde{Q}$ is also a scaled diagonally dominant matrix. It follows that f is an SDSOS polynomial in the monomial basis $\mathscr{B}_*$. ∎

Remark 4.8. By Proposition 4.7, if we use the monomial basis $\mathscr{B}_*$ for (P^k) (3.3) and (P^k)* (3.5), we still have the hierarchy of optimal values:

$$\lambda_{sdsos} \le \lambda_1 \le \lambda_2 \le \cdots \le \lambda_{mom} \le \lambda^*.$$

Remark 4.9. For unconstrained POPs, one may take an initial monomial basis $\mathscr{B}$ via the Newton polytope method in Algorithm 4.1.

Algorithm 4.1 `GenerateBasis`.

Require: $\mathscr{A}$ and an initial monomial basis $\mathscr{B}$
Ensure: An increasing chain of potential monomial bases $(\mathscr{B}_p)_{p\ge1}$

```
Set B_0 := ∅;
Let p = 0;
while p = 0 or B_p ≠ B_{p−1} do
   p := p + 1;
   Set B_p := ∅;
   for each pair {β, γ} of B do
      if β + γ ∈ A ∪ (2B_{p−1}) then
         B_p := B_p ∪ {β, γ};
      end if
   end for
end while
return (B_p)_{p≥1};
```

This method enables us to obtain a monomial basis that may be strictly smaller than the monomial basis given by the Newton polytope method as the following example shows.

Example 4.10. Consider the polynomial $f = 1 + x + x^8$. The monomial basis given by the Newton polytope method is $\mathscr{B} = \{1, x, x^2, x^3, x^4\}$. By the above iterative procedure, we compute that $\mathscr{B}_1 = \{1, x, x^4\}$ and $\mathscr{B}_2 = \{1, x, x^2, x^4\}$. It turns out that f has no SOS representations with $\mathscr{B}_1$ while $\mathscr{B}_2$ can serve as a monomial basis to represent f as an SOS.

For the constrained case we use the notation of section 4. Fix a relaxation order $\hat{d}$ and a sparse order k of the chordal-TSSOS hierarchy. Then each iteration breaks into two steps.

For step 1, let the maximal cliques of $G^{(k)}_{j,\hat{d}}$ (4.2) be $C^{(k)}_{j,1}, C^{(k)}_{j,2}, \ldots, C^{(k)}_{j,l_j}$ for $j = 0, \ldots, m$. Let

$$\mathscr{F} = \mathrm{supp}(f) \cup \bigcup_{j=1}^{m} (\mathrm{supp}(g_j) + \bigcup_{i=1}^{l_j} (C^{(k)}_{j,i} + C^{(k)}_{j,i})). \tag{4.8}$$

Then call the algorithm `GenerateBasis` with $\mathscr{A} = \mathscr{F}$ and $\mathscr{B} = \mathscr{B}_{0,\hat{d}}$ to generate a new monomial basis $\mathscr{B}'_{0,\hat{d}}$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

For step 2, with the new monomial basis $\mathscr{B}'_{0,\hat{d}}$, we compute a new sparsity pattern graph $(G^{(k)}_{j,\hat{d}})'$ for $j = 0, \ldots, m$. Then go back to step 1.

Continue the iterative procedure until $\mathscr{B}'_{0,\hat{d}} = \mathscr{B}_{0,\hat{d}}$.

5. Computational cost discussion. In this section, we provide an estimate for the computational cost associated with the chordal-TSSOS hierarchy of POPs with sparse order $k = 1$. For ease of exposition we only consider the unconstrained case (3.5). We use the standard monomial basis $\mathscr{B} = \mathbb{N}^n_d$ for the discussion in order to obtain a quantitative comparison of computational costs.

Let $f \in \mathbb{R}[\mathbf{x}]$ be of degree $2d$ and G_0 the tsp graph associated with f. Let $(G_k)_{k\geq 1}$ be defined by (3.2). By Theorem 2.1, the complexity of (3.5) depends on the sizes of maximal cliques of G_k and the number of equality constraints, i.e., $|\mathrm{supp}(G_k) \cup (2\mathscr{B})|$. Since we rely on an approximately minimum chordal extension, only a small number of edges are added to G_0 in the process of obtaining G_1 from the chordal extension of G_0. In this case, the complexity of (3.5) for $k = 1$ depends mainly on the sizes of maximal cliques of G_0 and $|\mathrm{supp}(G_0) \cup (2\mathscr{B})|$.

For any $\boldsymbol{\alpha} = (\alpha_i) \in \mathbb{N}^n$, we call $\boldsymbol{\alpha} \,(\mathrm{mod}\ 2) = (\alpha_i \,(\mathrm{mod}\ 2)) \in \{0,1\}^n$ the *sign type* of $\boldsymbol{\alpha}$. We say that $\boldsymbol{\alpha}$ is *even* if the sign type of $\boldsymbol{\alpha}$ is $\mathbf{0}$ and is *odd* otherwise.

PROPOSITION 5.1. *Let $G_0(V, E_0)$ be the tsp graph associated with f. Then for any $\boldsymbol{\beta}, \boldsymbol{\gamma} \in V$ with the same sign type, one has $\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \in E_0$.*

Proof. Since $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ have the same sign type, $\boldsymbol{\beta} + \boldsymbol{\gamma}$ is even, i.e., $\boldsymbol{\beta} + \boldsymbol{\gamma} \in 2\mathscr{B}$, and therefore $\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \in E_0$ by (3.1). ∎

Proposition 5.1 implies that the nodes of G_0 with the same sign type have pairwise edges and hence form a clique of G_0.

PROPOSITION 5.2. *Let C be a subset of $\mathbb{N}^n_d$ such that the elements of C have the same sign type. Then $|C| \leq \binom{n+\lfloor \frac{d}{2} \rfloor}{\lfloor \frac{d}{2} \rfloor}$.*

Proof. Let $\mathbf{y} = (y_1, \ldots, y_n)$ be a set of variables and $\mathbf{s}$ be the sign type of the elements in C. It follows that any $\boldsymbol{\alpha} \in C$ is a nonnegative integer solution of the following system:

$$\begin{cases} y_1 + y_2 + \cdots + y_n \leq d, \\ \mathbf{y} \,(\mathrm{mod}\ 2) = \mathbf{s}. \end{cases} \tag{5.1}$$

Define $\bar{\mathbf{y}} = (\bar{y}_1, \cdots, \bar{y}_n)$ by

$$\bar{y}_i = \begin{cases} \frac{y_i}{2} & \text{if } y_i \,(\mathrm{mod}\ 2) = 0, \\ \frac{y_i - 1}{2} & \text{if } y_i \,(\mathrm{mod}\ 2) = 1. \end{cases}$$

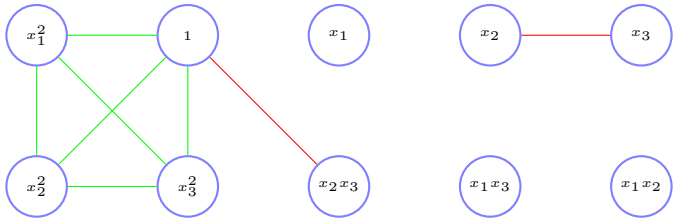
Let o be the number of subscripts i such that $s_i = 1$. Then the nonnegative integer solution of (5.1) is in a one-to-one correspondence to the nonnegative integer solution of the following system:

$$\bar{y}_1 + \bar{y}_2 + \cdots + \bar{y}_n \leq \left\lfloor \frac{d-o}{2} \right\rfloor. \tag{5.2}$$

Since the number of nonnegative integer solutions to (5.2) is $\binom{n+\lfloor \frac{d-o}{2} \rfloor}{\lfloor \frac{d-o}{2} \rfloor}$, we conclude that the number of nonnegative integer solutions to (5.1) is also $\binom{n+\lfloor \frac{d-o}{2} \rfloor}{\lfloor \frac{d-o}{2} \rfloor}$. Therefore, we have $|C| \leq \binom{n+\lfloor \frac{d-o}{2} \rfloor}{\lfloor \frac{d-o}{2} \rfloor} \leq \binom{n+\lfloor \frac{d}{2} \rfloor}{\lfloor \frac{d}{2} \rfloor}$ as o is a nonnegative integer. ∎

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



The green edges are even and the red edges are odd.

FIG. 5. *The tsp graph G_0 of f.*

TABLE 1
Computational cost comparison for the sparse and dense SDP relaxations of unconstrained POPs.

	Maximal size of SDP blocks	#SDP blocks	#Equality constraints
Sparse	$\sim \binom{n+\lfloor \frac{d}{2} \rfloor}{\lfloor \frac{d}{2} \rfloor}$	$\leq \binom{n+d}{d}$	$\sim \lvert\mathrm{supp}(f)\rvert + \binom{n+d}{d}$
Dense	$\binom{n+d}{d}$	1	$\binom{n+2d}{2d}$

Combining Proposition 5.1 with Proposition 5.2, we conclude that the size of any clique of G_0 whose nodes have the same sign type is no more than $\binom{n+\lfloor \frac{d}{2} \rfloor}{\lfloor \frac{d}{2} \rfloor}$.

Suppose $G(V, E)$ is a graph with $V = \mathscr{B}$. We say that an edge $\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \in E$ is *even* if $\boldsymbol{\beta} + \boldsymbol{\gamma}$ is even and is *odd* if $\boldsymbol{\beta} + \boldsymbol{\gamma}$ is odd. In other words, an even edge connects two nodes of the same sign type and an odd edge connects two nodes of different sign types. From the definition of G_0, odd edges of G_0 correspond to odd vectors in $\mathrm{supp}(f)$. If f is sufficiently sparse such that the odd edges of G_0 are not too many, then because of the above discussion, the maximal size of maximal cliques of G_0 is close to $\binom{n+\lfloor \frac{d}{2} \rfloor}{\lfloor \frac{d}{2} \rfloor}$.

It is known that for a chordal graph whose edge set is nonempty, the number of maximal cliques is less than the number of nodes [9]. Therefore, the number of maximal cliques of G_0 is bounded by $|\mathscr{B}| = \binom{n+d}{d}$.

By construction, we have $|\mathrm{supp}(G_0) \cup (2\mathscr{B})| = |\mathrm{supp}(f) \cup (2\mathscr{B})| \leq |\mathrm{supp}(f)| + \binom{n+d}{d}$.

Example 5.3. Consider the polynomial $f = 1 + x_1^4 + x_2^4 + x_3^4 - x_1^2x_2^2 - x_1^2x_3^2 - x_2^2x_3^2 + x_2x_3$. See Figure 5 for the tsp graph G_0 of f. There are six maximal cliques for G_0, which are of sizes $4, 2, 2, 1, 1, 1$, respectively.

On the other hand, for the dense SDP relaxation (2.7) of (3.5), there is only one SDP matrix which is of size $\binom{n+d}{d}$ and the number of equality constraints is $\binom{n+2d}{2d}$.

Thus we obtain Table 1 for the computational cost of the sparse (with sparse order $k = 1$) and dense SDP relaxations of (3.5).

We illustrate the above discussion by an explicit example.

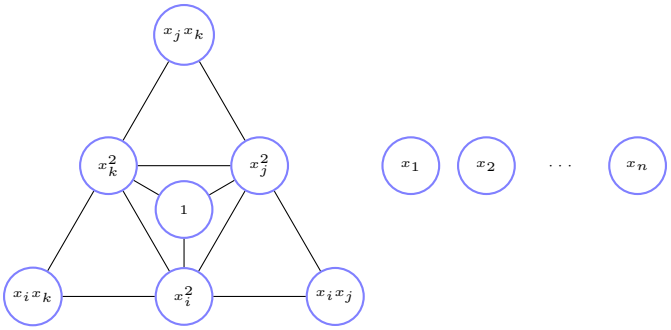
Example 5.4. For $n \geq 1$, let

$$f_n = \sum_{i=1}^{n}(x_i^2 + x_i^4) + \sum_{i=1}^{n}\sum_{k=1}^{n}(x_i - x_k)^4. \tag{5.3}$$

The tsp graph G_0 for f_n (see Figure 6) has 1 maximal clique of size $n+1$ (involving the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



This is a subgraph of G_0. The whole graph G_0 is obtained by putting all such subgraphs together.

FIG. 6. *The tsp graph G_0 of f_n.*

TABLE 2
Computational cost comparison for the sparse and dense SDP relaxations of minimizing f_n.

	#SDP blocks	#Equality constraints
Sparse	$3 \times \frac{n(n-1)}{2}, 1 \times n, (n+1) \times 1$	$\frac{3n(n-1)}{2} + 2n + 1$
Dense	$\binom{n+2}{2} \times 1$	$\binom{n+4}{4}$

nodes $1, x_1^2, \ldots, x_n^2$), $\frac{n(n-1)}{2}$ maximal cliques of size 3 (involving the nodes x_i^2, x_j^2, x_ix_j for each pair $\{i,j\}, i \neq j$), and n maximal cliques of size 1 (involving the node x_i for each i). Note that G_0 is already a chordal graph. So we have $G_1 = G_0$.

The computational cost for the sparse (with sparse order $k = 1$) and dense SDP relaxations of minimizing f_n is displayed in Table 2. In the column "#SDP blocks," $i \times j$ means j SDP blocks of size i.

6. Numerical experiments. In this section, we present numerical results of the proposed chordal-TSSOS hierarchies (3.3)–(3.5) and (4.4)–(4.6) for both unconstrained and constrained POPs, respectively. Our tool, named TSSOS, is implemented in Julia and constructs instances of the dual SDP problems (3.5) and (4.6) via JuMP [7], then relies on MOSEK [24] to solve them. TSSOS utilizes the Julia packages LightGraphs [4] to handle graphs. TSSOS also implements the block-TSSOS hierarchy developed in [32]. In the following subsections, we compare the performance of TSSOS with that of GloptiPoly [12], Yalmip [19], and SparsePOP [35]. As for TSSOS, GloptiPoly and Yalmip use MOSEK as an SDP solver. SparsePOP uses SDPT3 [29] as an SDP solver. We use the default accuracy 1×10^{-8} for both MOSEK and SDPT3.

Our TSSOS tool is available on the following website:

https://github.com/wangjie212/TSSOS.

All numerical examples were computed on an Intel Core i5-8265U@1.60GHz CPU with 8GB RAM memory. 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 that 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 notation that we use is listed in Table 3.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

TABLE 3
The notation.

n	the number of variables
$2d$	the degree
s	the number of terms
$\hat{d}$	the relaxation order of Lasserre's hierarchy
k	the sparse order of the (block-) chordal-TSSOS hierarchy
bs	the size of initial monomial bases by the Newton polytope method
rbs	the size of reduced monomial bases by Algorithm 4.1
mb	the maximal size of SDP blocks or whose kth entry is the maximal size of SDP blocks of the chordal-TSSOS hierarchy at sparse order k
opt	the optimal value or whose kth entry is the optimal value of the chordal-TSSOS hierarchy at sparse order k
time	the running time (in seconds) or whose kth entry is the running time (in seconds) of the chordal-TSSOS hierarchy at sparse order k
0	a number whose absolute value is less than 1×10^{-5}
-	out of memory

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

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

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

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

Table 4 displays the numerical results on these polynomials. We only present the results of the first three steps (i.e., $k = 1, 2, 3$) of the chordal-TSSOS hierarchy since for all instances except F_3, the sequence of graphs $(G_k)_{k\geq 1}$ stabilizes in three steps. Note that the time spent to compute a monomial basis is included in the running time of the chordal-TSSOS hierarchy at sparse order $k = 1$. This explains why the running time is less important at $k = 2$ for F_2, F_3, and F_4.

In Table 5, we compare the performance of TSSOS, GloptiPoly, Yalmip, and SparsePOP when achieving the same optimum on these polynomials. We present

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

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

TABLE 4
The results for randomly generated polynomials of type I.

	n	$2d$	s	bs	rbs	mb	opt	Time
F_1	8	8	64	106	49	4	0	0.24
F_2	8	8	102	122	76	6, 6	0, 0	0.34, 0.08
F_3	8	8	104	150	95	6, 8, 11	0, 0, 0	0.36, 0.05, 0.08
F_4	8	10	103	202	75	5, 5	0, 0	0.58, 0.04
F_5	8	10	85	201	70	4	0	0.53
F_6	8	10	111	128	60	7	0	0.38
F_7	9	10	101	145	66	5	0	0.50
F_8	9	10	166	178	81	5	0	0.72
F_9	9	10	161	171	78	8	0	0.79
F_{10}	10	12	271	223	94	8	0	2.2
F_{11}	10	12	253	176	88	9	0	1.6
F_{12}	10	12	261	204	94	8	0	1.8
F_{13}	10	16	370	1098	125	9	0	15
F_{14}	10	16	412	800	139	9	0	14
F_{15}	10	16	436	618	146	9	0	12
F_{16}	12	12	488	330	187	8	0	8.4
F_{17}	12	12	351	264	150	8	0	5.7
F_{18}	12	12	464	316	179	8	0	7.4

TABLE 5
Comparison with GloptiPoly, Yalmip, and SparsePOP for randomly generated polynomials of type I.

	TSSOS		GloptiPoly	Yalmip	SparsePOP	
	Chordal	Block				
	Time				mb	Time
F_1	0.24	1.7	306	10	105	24
F_2	0.34	4.6	348	13	140	130
F_3	0.36	8.8	326	19	153	175
F_4	0.58	4.8	-	92	233	323
F_5	0.53	4.2	-	72	201	1526
F_6	0.38	5.2	-	22	140	134
F_7	0.50	3.2	-	44	153	324
F_8	0.72	6.5	-	143	-	-
F_9	0.79	5.9	-	109	186	284
F_{10}	2.2	12	-	474	-	-
F_{11}	1.6	9.2	-	147	160	318
F_{12}	1.8	12	-	350	168	404
F_{13}	15	36	-	-	-	-
F_{14}	14	305	-	-	-	-
F_{15}	12	207	-	-	-	-
F_{16}	8.4	61	-	-	-	-
F_{17}	5.7	17	-	-	-	-
F_{18}	7.4	22	-	-	-	-

the performance of TSSOS for both "chordal" and "block" approaches. In Yalmip, we turn the option "sos.newton" on to compute a monomial basis by the Newton polytope method. Since SparsePOP uses a different SDP solver, we also provide the data of maximal sizes of SDP blocks produced by SparsePOP for comparison.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

As the tables illustrate, TSSOS is significantly faster and scales better than GloptiPoly, Yalmip, and SparsePOP. Note also that TSSOS produces much smaller SDP blocks than SparsePOP. Moreover, one can see that the chordal-TSSOS hierarchy performs much better than the block-TSSOS hierarchy.

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

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

Table 6 displays the numerical results on these polynomials. We only present the results of the first three steps (i.e., $k = 1, 2, 3$) of the chordal-TSSOS hierarchy since it always converges to the same optimum with the dense moment-SOS relaxation in three steps. It happens that the second (or the third) step of the chordal-TSSOS hierarchy spends less time than the first one because it involves fewer SDP blocks than the first one while the sizes of maximal SDP blocks are close.

In Table 7, we compare the performance of TSSOS, GloptiPoly, Yalmip, and SparsePOP when achieving the same optimum on these polynomials. We present the performance of TSSOS for both "chordal" and "block" approaches. 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.

Again as illustrated in the tables, TSSOS is significantly faster and scales better than GloptiPoly, Yalmip, and SparsePOP. As above, TSSOS produces much smaller SDP blocks than SparsePOP. In addition, the chordal-TSSOS hierarchy performs much better than the block-TSSOS hierarchy.

The Broyden banded function [34] is defined by

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

where $J_i = \{j \mid j \neq i, \max(1, i-5) \leq j \leq \min(n, i+1)\}$. Table 8 displays the results of the Broyden banded function for the chordal-TSSOS hierarchy and SparsePOP. The optimums are always 0. One can see that the tsp leads to much smaller SDP blocks than the csp (19 against 120 when $n \geq 8$) and thus saves more computational cost.

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

TABLE 6
The results for randomly generated polynomials of type II.

	n	$2d$	s	bs	rbs	mb	opt	Time
G_1	8	8	15	495	235	21, 23, 28	−0.5758, −0.5758, −0.5758	0.36, 0.21, 0.28
G_2	8	8	15	495	328	31, 33, 37	−34.69, −34.69, −34.69	0.51, 0.57, 1.7
G_3	8	8	15	495	258	21, 23, 31	0.7073, 0.7073, 0.7073	0.31, 0.23, 0.36
G_4	9	8	20	715	415	31, 41, 127	−801.7, −801.7, −801.7	1.0, 1.8, 184
G_5	9	8	20	715	342	28, 31, 40	−0.8064, −0.8064, −0.8064	0.63, 0.45, 1.1
G_6	9	8	20	715	340	25, 43, 61	−1.698, −1.698, −1.698	0.76, 1.4, 3.6
G_7	9	10	15	2002	1254	38, 41, 55	−1.295, −1.295, −1.295	6.6, 5.2, 15
G_8	9	10	15	2002	894	26, 28, 40	−0.6622, −0.6622, −0.6622	5.0, 1.8, 3.5
G_9	9	10	15	2002	888	28, 31, 31	0.5180, 0.5180, 0.5180	4.9, 1.4, 2.3
G_{10}	10	8	20	1001	454	31, 36, 42	−0.4895, −0.4895, −0.4895	1.2, 0.73, 0.92
G_{11}	10	8	20	1001	414	26, 33, 60	0.1732, 0.1798, 0.1867	1.1, 0.87, 6.0
G_{12}	10	8	20	1001	387	23, 37, 52	0.4943, 0.4943, 0.4943	1.0, 1.2, 2.4
G_{13}	11	8	20	1365	299	21, 22, 22	−3.963, −3.963, −3.963	1.7, 0.26, 0.30
G_{14}	11	8	20	1365	412	27, 33, 42	−2.184, −2.184, −2.184	1.8, 0.68, 3.3
G_{15}	11	8	20	1365	458	27, 30, 37	0.0588, 0.0588, 0.0588	1.9, 0.59, 0.76
G_{16}	12	8	25	1820	744	39, 58, 81	−758.6, −688.0, −688.0	4.1, 5.8, 73
G_{17}	12	8	25	1820	694	37, 51, 76	−40.89, −40.22, −40.22	3.7, 3.7, 31
G_{18}	12	8	25	1820	581	31, 40, 48	−14.27, −14.27, −14.27	2.9, 1.3, 1.8

TABLE 7
Comparison with GloptiPoly, Yalmip, and SparsePOP for randomly generated polynomials of type II.

	TSSOS		GloptiPoly	Yalmip	SparsePOP	
	Chordal	Block				
	Time				mb	Time
G_1	0.36	8.5	346	31	330	271
G_2	0.51	2.6	447	24	330	496
G_3	0.31	1.0	257	6.0	330	178
G_4	1.0	40	-	-	-	-
G_5	0.63	24	-	363	330	611
G_6	0.76	31	-	141	330	578
G_7	6.6	24	-	322	-	-
G_8	5.0	28	-	233	-	-
G_9	4.9	21	-	249	-	-
G_{10}	1.2	13	-	-	-	-
G_{11}	8.0	86	-	536	-	-
G_{12}	1.0	66	-	-	-	-
G_{13}	1.7	13	-	655	330	398
G_{14}	1.8	37	-	-	210	221
G_{15}	1.9	36	-	340	330	293
G_{16}	10	693	-	-	-	-
G_{17}	7.4	333	-	-	-	-
G_{18}	2.9	393	-	-	-	-

TABLE 8
The results for the Broyden banded function.

n		6	7	8	9	10	11	12	13	14	15
mb	TSSOS	15	17	19	19	19	19	19	19	19	19
	SparsePOP	84	120	120	120	120	120	120	120	120	120
Time	TSSOS	0.14	0.18	0.22	0.26	0.30	0.35	0.43	0.54	0.62	0.76
	SparsePOP	1.4	4.0	11	18	28	48	69	126	190	230

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 9
The results for the modified generalized Rosenbrock function.

n	TSSOS					
	Chordal			Block		
	mb	opt	Time	mb	opt	Time
10	11	8.45	0.03	28, 56	8.45, 8.45	0.06, 0.32
20	21	18.35	0.12	58, 211	18.35, 18.35	1.2, 50
30	31	28.25	0.34	88, 466	28.25, -	9, -
40	41	38.15	0.87	118, 821	38.15, -	42, -
50	51	48.05	2.5	148, 1276	48.05, -	146, -
60	61	57.95	4.1	178, 1831	57.95, -	382, -
70	71	67.85	9.5	218, 2486	67.85, -	786, -
80	81	77.75	22	248, 3241	77.75, -	4467, -
90	91	87.65	28	278, 4096	-	-
100	101	97.55	46	308, 5051	-	-
120	121	117.35	105	368, 7261	-	-
140	141	137.15	243	428, 9871	-	-
160	161	156.95	504	488, 12881	-	-
180	181	176.75	820	548, 16291	-	-
200	201	196.55	1792	608, 20101	-	-

In the final part of this subsection, we present the numerical results for the following two functions:

• The modified generalized Rosenbrock function

$$f_{\mathrm{mgR}}(\mathbf{x}) = 1 + \sum_{i=1}^{n}(100(x_i - x_{i-1}^2)^2 + (1-x_i)^2) + \sum_{i=1}^{n}\sum_{j=i+1}^{n} x_i^2 x_j^2,$$

which is obtained from the generalized Rosenbrock function by adding monomial terms such that the csp graph is complete.

• The modified chained singular function

$$\begin{aligned} f_{\mathrm{mcs}}(\mathbf{x}) = &\sum_{i\in J}((x_i + 10x_{i+1})^2 + 5(x_{i+2} - x_{i+3})^2 + (x_{i+1} - 2x_{i+2})^4 \\ &+ 10(x_i - 10x_{i+3})^4 + \sum_{i=1}^{n}\sum_{j=i+1}^{n} x_i^2 x_j^2 \end{aligned}$$

with $J = \{1, 3, 5, \ldots, n-3\}$, which is obtained from the chained singular function by adding the same monomial terms as above.

The results for the modified generalized Rosenbrock function and the chained singular function are displayed in Tables 9 and 10, respectively. As the tables illustrate, the chordal-TSSOS hierarchy can handle these functions with variables up to 200 while the block-TSSOS hierarchy can handle these functions with variables no more than 100 due to the memory constraint.

6.2. Constrained polynomial optimization problems. Now we present the numerical results for constrained polynomial optimization problems. We first consider six randomly generated polynomials $H_1, \ldots, H_6$ of type II[3] as objective functions f

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

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

TABLE 10
The results for the modified chained singular function.

n	TSSOS					
	Chordal			Block		
	mb	opt	Time	mb	opt	Time
10	11	−0.0003	0.06	24, 56	−0.0006, −0.0007	0.07, 0.42
20	21	−0.0013	0.11	49, 211	−0.0006, −0.0007	0.77, 78
30	31	−0.0004	0.37	74, 466	−0.0002, -	3.9, -
40	41	−0.0007	0.85	99, 821	−0.0001, -	15, -
50	51	−0.0021	2.1	124, 1276	−0.0006, -	45, -
60	61	−0.0021	4.7	149, 1831	−0.0002, -	112, -
70	71	−0.0030	7.6	174, 2486	−0.0005, -	282, -
80	81	−0.0040	19	199, 3241	−0.0002, -	670, -
90	91	−0.0034	23	224, 4096	−0.0004, -	1768, -
100	101	−0.0038	37	249, 5051	-	-
120	121	−0.0014	88	274, 7261	-	-
140	141	−0.0011	199	299, 9871	-	-
160	161	−0.0015	390	324, 12881	-	-
180	181	−0.0057	682	349, 16291	-	-
200	201	−0.0083	1126	374, 20101	-	-

and minimize them over the two following semialgebraic sets: 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

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

The results for the unit ball case are displayed in Table 11 (whose csp is clearly trivial because the constraint of unit balls involves all the variables) and the results for the unit hypercube case are displayed in Table 12 (only the results of the first three steps of the chordal-TSSOS hierarchy are displayed). We compare the performance of the chordal-TSSOS hierarchy with that of GloptiPoly (and SparsePOP in the unit hypercube case). It can be seen that for each instance TSSOS is significantly faster than GloptiPoly (and faster than SparsePOP in the unit hypercube case) without compromising accuracy.

Next we present the numerical results of the following two functions over the unit ball.

- The Broyden tridiagonal function

$$f_{\mathrm{Bt}}(\mathbf{x}) = ((3 - 2x_1)x_1 - 2x_2 + 1)^2 + \sum_{i=2}^{n-1}((3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1)^2 + ((3 - 2x_n)x_n - x_{n-1} + 1)^2.$$

- The generalized Rosenbrock function

$$f_{\mathrm{gR}}(\mathbf{x}) = 1 + \sum_{i=1}^{n}(100(x_i - x_{i-1}^2)^2 + (1 - x_i)^2).$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 11

The results for minimizing randomly generated polynomials of type II *over unit balls. In this table, the first entry of "mb" is the maximal size of SDP blocks corresponding to the moment matrix* $M_{\hat{d}}(\mathbf{y})$ *and the second entry is the maximal size of SDP blocks corresponding to the localizing matrix* $M_{\hat{d}-d_1}(g_1\mathbf{y})$.

	$(n, 2d, s)$	$\hat{d}$	TSSOS				GloptiPoly	
			k	mb	opt	Time	opt	Time
H_1	(6,8,10)	4	1	(28, 7)	0.1362	0.20	0.1362	8.0
			2	(32, 12)		0.52		
			3	(37, 20)		0.86		
		5	1	(29, 28)		0.91		80
			2	(35, 30)		3.0		
			3	(48, 45)		9.0		
H_2	(7,8,12)	4	1	(36, 8)	0.1373	0.36	0.1373	34
			2	(36, 10)		0.52		
			3	(38, 15)		1.6		
		5	1	(36, 36)		1.9	-	-
			2	(45, 36)		3.9		
			3	(59, 49)		34		
H_3	(8,8,15)	4	1	(45, 9)	0.1212	0.75	0.1212	225
			2	(45, 10)		1.3		
			3	(53, 25)		20		
		5	1	(45, 45)		5.3	-	-
			2	(45, 45)		7.5		
			3	(59, 46)		94		
H_4	(9,6,15)	3	1	(10, 10)	0.8704	0.15	0.8704	16
			2	(10, 10)		0.22		
			3	(10, 10)		0.25		
		4	1	(55, 10)		1.3	-	-
			2	(55, 13)		2.0		
			3	(56, 19)		2.8		
H_5	(10,6,20)	3	1	(12, 11)	0.5966	0.22	0.5966	48
			2	(13, 14)		0.42		
			3	(19, 16)		0.95		
		4	1	(66, 13)		2.5	-	-
			2	(66, 23)		10		
			3	(75, 44)		88		
H_6	(11,6,20)	3	1	(12, 12)	0.1171	0.28	0.1171	115
			2	(15, 12)		0.36		
			3	(16, 13)		0.60		
		4	1	(78, 14)		4.4	-	-
			2	(78, 15)		4.7		
			3	(78, 13)		7.5		

Since the constraint of unit balls involves all the variables, the csp for these problems is clearly trivial. The results for the generalized Rosenbrock function are displayed in Table 13 and the results for the Broyden tridiagonal function are displayed in Table 14. The relaxation order $\hat{d}$ is 2. We present the performance of TSSOS for

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 12

The results for minimizing randomly generated polynomials of type II *over unit hypercubes. In this table, the first entry of the fifth column "mb" is the maximal size of SDP blocks corresponding to the moment matrix $M_{\hat{d}}(\mathbf{y})$ and the second entry is the maximal size of SDP blocks corresponding to the localizing matrices $M_{\hat{d}-d_j}(g_j\mathbf{y}), j = 1, \ldots, n$.*

	$(n, 2d, s)$	$\hat{d}$	TSSOS				GloptiPoly		SparsePOP		
			k	mb	opt	Time	opt	Time	mb	opt	Time
H_1	(6,8,10)	4	1	(28, 8)	-0.4400	0.33	-0.4400	19	126	-0.4400	9.8
			2	(32, 12)		0.86					
			3	(37, 19)		1.4					
		5	1	(29, 28)		1.7		237	252		83
			2	(35, 30)		6.1					
			3	(48, 45)		16					
H_2	(7,8,12)	4	1	(36, 9)	-0.1289	0.80	-0.1289	101	126	-0.1289	9.1
			2	(36, 10)		0.94					
			3	(38, 13)		1.9					
		5	1	(36, 36)		3.6	-	-	252		129
			2	(45, 36)		6.8					
			3	(59, 49)		61					
H_3	(8,8,15)	4	1	(45, 10)	-0.1465	1.0	-0.1465	433	210	-0.1465	103
			2	(45, 11)		2.0					
			3	(53, 22)		24					
		5	1	(45, 45)		9.6	-	-	-	-	-
			2	(45, 45)		13					
			3	(59, 50)		112					
H_4	(9,6,15)	3	1	(10, 10)	0.1199	0.27	0.1199	27	35	0.1199	1.3
			2	(10, 10)		0.30					
			3	(10, 10)		0.32					
		4	1	(55, 11)		2.1	-	-	70		6.0
			2	(55, 13)		2.8					
			3	(56, 19)		4.0					
H_5	(10,6,20)	3	1	(12, 11)	-0.2813	0.38	-0.2813	69	84	-0.2822	14
			2	(13, 12)		0.50					
			3	(19, 13)		0.70					
		4	1	(66, 14)		4.4	-	-	-	-	-
			2	(66, 23)		15					
			3	(75, 44)		80					
H_6	(11,6,20)	3	1	(12, 12)	-0.2316	0.47	-0.2316	211	84	-0.2316	5.9
			2	(15, 12)		0.61					
			3	(16, 13)		0.76					
		4	1	(78, 13)		7.5	-	-	210		162
			2	(78, 15)		9.9					
			3	(78, 13)		13					

both "chordal" and "block" approaches. As the tables illustrate, again the chordal-TSSOS hierarchy performs much better than the block-TSSOS hierarchy.

7. Conclusions and outlook. In this paper, a follow-up on our previous work [32], we continue to exploit the tsp for a POP. Through the support-extension and chordal-extension operations, we iteratively enlarge the tsp graph to obtain a chordal-TSSOS hierarchy of sparse SDP relaxations for a POP. Various numerical examples demonstrate the efficiency and the scalability of this new hierarchy for unconstrained and constrained POPs.

There are still many questions left for further investigations:

(1) When relying on the dense moment-SOS hierarchy, one can extract the global optimizers under certain flatness conditions of the moment matrix [13]. A similar procedure exists for the sparse moment-SOS hierarchy based on correlative sparsity [15, 17]. It is worth looking for a similar condition when using the chordal-TSSOS hierarchy.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 13

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

n	TSSOS							
	Chordal				Block			
	k	mb	opt	Time	k	mb	opt	Time
10	1	(11, 2)	8.35	0.05	1	(28, 10)	8.35	0.22
					2	(56, 10)	8.35	0.32
20	1	(21, 2)	18.25	0.19	1	(58, 20)	18.25	8.2
					2	(211, 20)	18.25	45
30	1	(31, 2)	28.15	0.49	1	(88, 30)	28.15	203
					2	(466, 30)	-	-
40	1	(41, 2)	38.05	1.3	1	(118, 40)	-	-
					2	(821, 40)	-	-
50	1	(51, 2)	47.95	4.0	1	(148, 50)	-	-
					2	(1276, 50)	-	-
60	1	(61, 2)	57.85	6.6	1	(178, 60)	-	-
					2	(1831, 60)	-	-
70	1	(71, 2)	67.75	18	1	(218, 70)	-	-
					2	(2486, 70)	-	-
80	1	(81, 2)	77.65	26	1	(248, 78)	-	-
					2	3241, 80)	-	-
90	1	(91, 2)	87.55	50	1	(278, 90)	-	-
					2	(4096, 90)	-	-
100	1	(101, 2)	97.45	85	1	(308, 100)	-	-
					2	(5051, 100)	-	-
120	1	(121, 2)	117.25	186	1	(368, 120)	-	-
					2	(7261, 120)	-	-
140	1	(141, 2)	137.05	448	1	(428, 140)	-	-
					2	(9871, 140)	-	-
160	1	(161, 2)	156.85	841	1	(488, 160)	-	-
					2	(12881, 160)	-	-
180	1	(181, 2)	176.65	1495	1	(548, 180)	-	-
					2	(16291, 180)	-	-

(2) We have the freedom to choose a specific chordal extension for the chordal-extension operation. The computational cost of the chordal-TSSOS hierarchy as well as its convergence (to the optimum of the dense relaxation) and convergence rate highly depend on this choice. A good chordal extension leads to a hierarchy converging quickly together with a low computational cost. In this paper, we have used an approximately minimum chordal extension, which performs fairly well, at least on the examples tested in this paper. However, as [27] suggests, an approximately minimum chordal extension is not always optimal. Therefore one could explore more general choices of chordal extensions to find a good one for specific POPs.

(3) The chordal-TSSOS hierarchy can be combined with other techniques for handling large-scale POPs, e.g., correlative sparsity [34] or structured subsets [23]. This issue is the focus of our recent work [33].

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

TABLE 14

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

n	TSSOS							
	Chordal				Block			
	k	mb	opt	Time	k	mb	opt	Time
10	1	(13, 5)	5.15	0.07	1	(38, 11)	5.15	0.20
	2	(13, 5)	5.15	0.08	2	(66, 11)	5.15	0.32
20	1	(23, 5)	15.04	0.37	1	(78, 21)	15.04	11
	2	(23, 7)	15.04	0.50	2	(231, 21)	15.04	73
30	1	(33, 5)	25.01	1.2	1	(118, 31)	25.01	174
	2	(33, 7)	25.01	2.1	2	(496, 31)	-	-
40	1	(43, 5)	35.00	3.6	1	(158, 41)	-	-
	2	(43, 7)	35.00	7.8	2	(861, 41)	-	-
50	1	(53, 5)	44.99	9.0	1	(198, 51)	-	-
	2	(53, 7)	44.99	21	2	(1326, 51)	-	-
60	1	(65, 5)	54.99	18	1	(238, 61)	-	-
	2	(65, 7)	54.99	48	2	(1891, 61)	-	-
70	1	(73, 5)	64.99	41	1	(278, 71)	-	-
	2	(73, 7)	64.99	138	2	(2556, 71)	-	-
80	1	(83, 5)	74.99	67	1	(318, 81)	-	-
	2	(83, 7)	74.99	240	2	(3321, 81)	-	-
90	1	(93, 5)	84.99	110	1	(348, 91)	-	-
	2	(93, 7)	84.99	193	2	(4186, 91)	-	-
100	1	(103, 5)	94.98	188	1	(378, 101)	-	-
	2	(103, 7)	94.98	299	2	(5151, 101)	-	-
120	1	(123, 4)	114.98	374	1	(458, 121)	-	-
	2	(123, 8)	114.98	864	2	(7381, 121)	-	-

REFERENCES

[1] 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.

[2] P. R. AMESTOY, T. A. DAVIS, AND I. S. DUFF, *Algorithm* 837: *AMD, an approximate minimum degree ordering algorithm*, ACM Trans. Math. Software, 30 (2004), pp. 381–388.

[3] J. R. S. BLAIR AND B. PEYTON, *An introduction to chordal graphs and clique trees*, in Graph Theory and Sparse Matrix Computation, A. George, J. R. Gilbert, and J. W. H. Liu, eds., Springer, New York, 1993, pp. 1–29.

[4] S. BROMBERGER, J. FAIRBANKS, AND OTHER CONTRIBUTORS, *JuliaGraphs/LightGraphs.jl: An Optimized Graphs Package for the Julia Programming Language*, preprint, https://doi.org/10.5281/zenodo.889971, 2017.

[5] T. CHEN, J. B. LASSERRE, V. MAGRON, AND E. PAUWELS, *Semialgebraic optimization for bounding Lipschitz constants of ReLU networks*, in Proceeding of Advances in Neural Information Processing Systems 33 (NIPS 2020), to appear.

[6] M. D. CHOI, T. Y. LAM, AND B. REZNICK, *Sums of squares of real polynomials*, Proc. Sympos. Pure Math., 58 (1995), pp. 103–126.

[7] I. DUNNING, J. HUCHETTE, AND M. LUBIN, *JuMP: A modeling language for mathematical optimization*, SIAM Rev., 59 (2017), pp. 295–320.

[8] D. R. FULKERSON AND O. A. GROSS, *Incidence matrices and interval graphs*, Pacific J. Math., 15 (1963), pp. 835–855.

[9] F. GAVRIL, *Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph*, SIAM J. Comput., 1 (1972), pp. 180–187.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[10] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, 1980.
[11] P. Heggernes, *Minimal triangulations of graphs: A survey*, Discrete Math., 306 (2006), pp. 297–317.
[12] D. Henrion and J. B. Lasserre, *GloptiPoly: Global optimization over polynomials with MATLAB and SeDuMi*, in Proceedings of the IEEE Conference on Decision and Control, Las Vegas, Nevada, 2002, pp. 747–752.
[13] D. Henrion and J. B. Lasserre, *Detecting global optimality and extracting solutions in GloptiPoly*, in Positive Polynomials in Control, Springer, New York, 2005, pp. 293–310.
[14] C. Josz, *Application of Polynomial Optimization to Electricity Transmission Networks*, Theses, Université Pierre et Marie Curie–Paris VI, 2016.
[15] I. Klep, V. Magron, and J. Povh, *Sparse noncommutative polynomial optimization*, Math. Program, to appear.
[16] J. B. Lasserre, *Global optimization with polynomials and the problem of moments*, SIAM J. Optim., 11 (2001), pp. 796–817.
[17] J. B. Lasserre, *Convergent SDP-relaxations in polynomial optimization with sparsity*, SIAM J. Optim., 17 (2006), pp. 822–843.
[18] 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.
[19] J. Löfberg, *YALMIP: A toolbox for modeling and optimization in MATLAB*, in Proceedings of the 2004 IEEE International Conference on Robotics and Automation, 2004, pp. 284–289.
[20] N. H. A. Mai, J. B. Lasserre, and V. Magron, *A Sparse Version of Reznick's Positivstellensatz*, preprint, http://arxiv.org/abs/2002.05101, 2020.
[21] V. Magron, G. Constantinides, and A. Donaldson, *Certified roundoff error bounds using semidefinite programming*, ACM Trans. Math. Software, 43 (2017), pp. 1–31.
[22] V. Magron, *Interval enclosures of upper bounds of roundoff errors using semidefinite programming*, ACM Trans. Math. Software, 44 (2018), pp. 1–18.
[23] J. Miller, Y. Zheng, M. Sznaier, and A. Papachristodoulou, *Decomposed Structured Subsets for Semidefinite and Sum-of-Squares Optimization*, preprint, http://arxiv.org/abs/1911.12859, 2019.
[24] MOSEK ApS, *The MOSEK Optimization Toolbox Version* 8.1, http://docs.mosek.com/8.1/toolbox/index.html, 2017.
[25] J. Nie, *Optimality conditions and finite convergence of Lasserre's hierarchy*, Math. Program., 146 (2014), pp. 97–121.
[26] B. Reznick, *Extremal PSD forms with few terms*, Duke Math. J., 45 (1978), pp. 363–374.
[27] J. Sliwak, M. Anjos, L. Létocart, J. Maeght, and E. Traversi, *Improving Clique Decompositions of Semidefinite Relaxations for Optimal Power Flow Problems*, preprint, https://arxiv.org/abs/1912.09232, 2019.
[28] M. Tacchi, T. Weisser, J.-B. Lasserre, and D. Henrion, *Exploiting Sparsity for Semi-Algebraic Set Volume Computation*, preprint, http://arxiv.org/abs/1902.02976, 2019.
[29] K. C. Toh, M. J. Todd, and R. H. Tutuncu, *SDPT3—A MATLAB software package for semidefinite programming*, Optim. Methods Softw., 11 (1999), pp. 545–581.
[30] L. Vandenberghe and M. S. Andersen, *Chordal Graphs and Semidefinite Optimization*, Found. Trends Optim. 1, Now Publishers, Hanover, 2015, pp. 241–433.
[31] J. Wang, H. Li and B. Xia, *A new sparse SOS decomposition algorithm based on term sparsity*, in Proceedings of the 2019 on International Symposium on Symbolic and Algebraic Computation, ACM, 2019, pp. 347–354.
[32] J. Wang, V. Magron and J. B. Lasserre, *TSSOS: A Moment-SOS Hierarchy that Exploits Term Sparsity*, SIAM J. Optim., to appear.
[33] 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.
[34] 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 (2016), pp. 218–242.
[35] 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 (2008), pp. 1–13.

Downloaded 03/15/22 to 65.49.38.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.