CS-TSSOS: Correlative and Term Sparsity for Large-Scale Polynomial Optimization

JIE WANG, Academy of Mathematics and Systems Science, CAS, Beijing, China
VICTOR MAGRON, J. B. LASSERRE, and NGOC HOANG ANH MAI, Laboratory for Analysis and Architecture of Systems, Université de Toulouse, Toulouse, France

This work proposes a new moment-SOS hierarchy, called *CS-TSSOS*, for solving large-scale sparse polynomial optimization problems. Its novelty is to exploit simultaneously *correlative sparsity* and *term sparsity* by combining advantages of two existing frameworks for sparse polynomial optimization. The former is due to Waki et al. [40] while the latter was initially proposed by Wang et al. [42] and later exploited in the TSSOS hierarchy [46, 47]. In doing so we obtain CS-TSSOS—a two-level hierarchy of semidefinite programming relaxations with (i) the crucial property to involve blocks of SDP matrices and (ii) the guarantee of convergence to the global optimum under certain conditions. We demonstrate its efficiency and scalability on several large-scale instances of the celebrated Max-Cut problem and the important industrial optimal power flow problem, involving up to six thousand variables and tens of thousands of constraints.

CCS Concepts: • **Theory of computation** → **Semidefinite programming**; **Convex optimization**; **Numeric approximation algorithms;**

Additional Key Words and Phrases: Moment-SOS hierarchy, Lasserre's hierarchy, correlative sparsity, term sparsity, TSSOS, large-scale polynomial optimization, optimal power flow

ACM Reference format:
Jie Wang, Victor Magron, J. B. Lasserre, and Ngoc Hoang Anh Mai. 2022. CS-TSSOS: Correlative and Term Sparsity for Large-Scale Polynomial Optimization. *ACM Trans. Math. Softw.* 48, 4, Article 42 (December 2022), 26 pages.
https://doi.org/10.1145/3569709

Jie Wang and Victor Magron were supported by the Tremplin ERC Stg Grant ANR-18-ERC2-0004-01 (T-COPS project). Victor Magron was supported by the FMJH Program PGMO (EPICS project) and EDF, Thales, Orange et Criteo. This work has benefited from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Actions, grant agreement 813211 (POEMA) as well as from the AI Interdisciplinary Institute ANITI funding, through the French "Investing for the Future PIA3" program under the Grant agreement n°ANR-19-PI3A-0004. J. B. Lasserre was also supported by the European Research Council (ERC) under the European's Union Horizon 2020 research and innovation program (grant agreement 666981 TAMING).

Authors' addresses: J. Wang, 北京市海淀区中关村东路55号, 思源楼; email: wangjie212@amss.ac.cn; V. Magron, J. B. Lasserre, and N. H. A. Mai, LAAS CNRS, 7 avenue du Colonel Roche F-31031 Toulouse, France; emails: {lasserre, vmagron, nhmai}@laas.fr.

0098-3500/2022/12-ART42 $15.00
https://doi.org/10.1145/3569709

1 INTRODUCTION

This article is concerned with solving large-scale polynomial optimization problems. As is often the case, the polynomials in the problem description involve only a few monomials of low degree and the ultimate goal is to exploit this crucial feature to provide semidefinite relaxations that are computationally much cheaper than those of the standard SOS-based hierarchy [20] or its sparse version [21, 40] based on correlative sparsity.

Throughout this article, we consider large-scale instances of the following **polynomial optimization problem (POP)**:

$$(\mathrm{Q}): \quad \rho^* = \inf_{\mathbf{x}} \{ f(\mathbf{x}) : \mathbf{x} \in \mathbf{K} \}, \tag{1.1}$$

where the objective function f is assumed to be a polynomial in n variables $\mathbf{x} = (x_1, \dots, x_n)$ and the feasible set $\mathbf{K} \subseteq \mathbb{R}^n$ is assumed to be defined by a finite conjunction of m polynomial inequalities, namely

$$\mathbf{K} := \{\mathbf{x} \in \mathbb{R}^n : g_1(\mathbf{x}) \geq 0, \dots, g_m(\mathbf{x}) \geq 0\}, \tag{1.2}$$

for some polynomials $g_1, \dots, g_m$ in $\mathbf{x}$. Here "large-scale" means that the magnitude of the number of variables n and the number of inequalities m can be both proportional to several thousands. A nowadays well-established scheme to handle (Q) is the *moment-SOS hierarchy* [20], where SOS is the abbreviation of *sum of squares*. The moment-SOS hierarchy provides a sequence of semidefinite programming (SDP) relaxations, whose optimal values are non-decreasing lower bounds of the global optimum ρ^* of (Q). Under some mild assumption slightly stronger than compactness, the sequence generically converges to the global optimum in finitely many steps [31]. SDP solvers [48] address a specific class of convex optimization problems, with linear cost and linear matrix inequalities. With a priory fixed precision, an SDP can be solved in polynomial time with respect to its input size. Modern SDP solvers via the interior-point method (e.g., `Mosek` [3]) can solve an SDP problem involving matrices of moderate size (say, ≤ 5,000) and equality constraints of moderate number (say, ≤ 20,000) in reasonable time on a standard laptop [38]. The SDP relaxations arising from the moment-SOS hierarchy typically involve matrices of size $\binom{n+d}{d}$ and equality constraints of number $\binom{n+2d}{2d}$, where d is the relaxation order. For problems with $n \simeq 200$, it is thus possible to compute the first-order SDP relaxation of a **quadratically constrained quadratic problem (QCQP)**, as one can take $d = 1$, yielding $\binom{n+d}{d} \simeq 200$ and $\binom{n+2d}{2d} \simeq 20{,}000$ (in this case, this relaxation is also known as Shor's relaxation [34]). However, the quality of the resulting approximation is often not satisfactory and it is then required to go beyond the first-order relaxation. But for solving the second-order relaxation ($d = 2$) one is limited to problems of small size, typically with $\binom{n+4}{4} \leq 20{,}000$ (hence, with $n \leq 23$) on a standard laptop. Therefore, in view of the current state of SDP solvers, the dense moment-SOS hierarchy does not scale well enough.

One possible remedy is to rely on alternative weaker positivity certificates, such as the hierarchy of **linear programming (LP)** relaxations based on Krivine-Stengle's certificates [19, 23, 35] or the second-order cone programming (SOCP) relaxation based on (scaled) diagonally dominant sums of squares (DSOS/SDSOS) [2] to approximate/bound from below the optimum of (Q). Even though modern LP/SOCP solvers can handle much larger problems by comparison with SDP solvers, they have been shown to provide less accurate bounds, in particular for combinatorial problems [24], and do not have the property of finite convergence for continuous problems (not even for convex QCQP problems [22, Section 9.3]). Another important methodology is to reduce the size of SDPs arising in the moment-SOS hierarchy via exploiting structure of POPs.

Related Work for Unconstructed POPs. A first option is to exploit *term sparsity* for sparse unconstrained problems, i.e., when $\mathbf{K} = \mathbb{R}^n$, and the objective f involves a few terms (monomials). The algorithm consists of automatically reducing the size of the corresponding SDP matrix by eliminating the monomial terms which never appear among the support of SOS decompositions [32]. Other classes of positivity certificates have been recently developed with a specific focus on sparse unconstrained problems. Instead of trying to decompose a positive polynomial as an SOS, one can try to decompose it as a **sum of nonnegative circuits (SONC)**, by solving a geometric program [16] or a second-order cone program [5, 43], or alternatively as a sum of arithmetic-geometric-mean-exponentials (SAGE) [8] by solving a relative entropy program. Despite their potential efficiency on certain sub-classes of POPs (e.g., sparse POPs with a small number of variables and a high degree), these methods share the common drawback of not providing systematic guarantees of convergence for constrained problems.

Related Work on Correlative Sparsity. In order to reduce the computational burden associated with the dense moment-SOS hierarchy while keeping its nice convergence properties, one possibility is to take into account the sparsity pattern satisfied by the variables of the POP [21, 40]. The resulting algorithm has been implemented in the `SparsePOP` solver [41] and can handle sparse problems with up to several hundred variables. Many applications of interest have been successfully handled thanks to this framework, for instance certified roundoff error bounds in computer arithmetics [25, 26] with up to several hundred variables and constraints, optimal power flow problems [17] (where a multi-ordered Lasserre hierarchy was proposed) with up to several thousand variables and constraints. More recent extensions have been developed for volume computation of sparse semialgebraic sets [37], approximating regions of attraction of sparse polynomial systems [36], noncommutative POPs [18], Lipschitz constant estimation of deep networks [9] and for sparse positive definite functions [28]. In these applications, the cost polynomial and the constraint polynomials possess a specific *correlative sparsity pattern*. The resulting sparse moment-SOS hierarchy is obtained by building blocks of SDP matrices with respect to some subsets or *cliques* of the input variables. When the sizes of these cliques are reasonably small, one can expect to handle problems with a large number of variables. For instance, the maximal size of cliques is less than 10 for some unconstrained problems in [40] or roundoff error problems in [26], and is less than 20 for the optimal power flow problems handled in [17]. Even though correlative sparsity has been successfully used to tackle several interesting applications, there are still many POPs that cannot be handled by considering merely correlative sparsity. For instance, there are POPs for which the correlative sparsity pattern is (nearly) dense or which admits a correlative sparsity pattern with variable cliques of large cardinality (say, > 20), yielding untractable SDPs.

Related Work on Term Sparsity. To overcome these issues, one can exploit *term sparsity* as described in [42], [46], and [47]. The *TSSOS hierarchy* from [47] as well as the complementary *Chordal-TSSOS* from [46] offers some alternative to problems for which the correlative sparsity pattern is dense or nearly dense. In both TSSOS and Chordal-TSSOS frameworks a so-called *term sparsity pattern (tsp) graph* is associated with the POP. The nodes of this tsp graph are monomials (from a monomial basis) needed to construct SOS relaxations of the POP. Two nodes are connected via an edge whenever the product of the corresponding monomials appears in the supports of polynomials involved in the POP or is a monomial square. Note that this graph differs from the *correlative sparsity pattern (csp) graph* used in [40] where the nodes are the input variables and the edges connect two nodes whenever the corresponding variables appear in the same term of the objective function or in the same constraint. A two-step iterative algorithm takes as input the tsp graph and enlarges it to exploit the term sparsity in (Q). Each iteration consists of two successive operations: (i) a support extension operation and (ii) either a block closure operation on adjacency matrices in the case of TSSOS [47] or a chordal extension operation in the case of Chordal-TSSOS

[46]. In doing so, one obtains a two-level moment-SOS hierarchy with blocks of SDP matrices. If the sizes of blocks are relatively small, then the resulting SDP relaxations become more tractable as their computational cost is significantly reduced. Another interesting feature of TSSOS is that the block structure obtained at the end of the iterative algorithm automatically induces a partition of the monomial basis, which can be interpreted in terms of sign symmetries of the initial POP. TSSOS and Chordal-TSSOS allow one to solve POPs with several hundred variables for which there is no or little correlative sparsity to exploit; see and [47], for numerous numerical examples. One can also rely on symmetry exploitation as in [33] but this requires quite strong assumptions on the input data, such as invariance of each polynomial $f, g_1, \ldots, g_m$ under the action of a finite group.

To tackle large-scale POPs, a natural idea is to simultaneously benefit from correlative and term sparsity patterns. This is the spirit of our contribution. Also in the same vein the work in [30] combines the (S)DSOS framework [2] with the TSSOS hierarchy but does not provide systematic convergence guarantees.

Contribution. Our main contribution is as follows:

- For large-scale POPs with a correlative sparsity pattern, we first apply the usual sparse polynomial optimization framework [21, 40] to get a coarse decomposition in terms of cliques of variables. Next we apply the term sparsity strategy (either TSSOS or Chordal-TSSOS) to each subsystem (which involves only one clique of variables) to reduce the size of SDPs even further. While the overall strategy is quite clear and simple, its implementation is not trivial and needs some care. Indeed for its coherency one needs to take extra care of the monomials which involve variables that belong to intersections of variable cliques (those obtained from correlative sparsity). The resulting combination of correlative sparsity (CS for short) and term sparsity produces what we call the *CS-TSSOS* hierarchy—a two-level hierarchy of SDP relaxations with blocks of SDP matrices, which yields a converging sequence of certified approximations for POPs. Under certain conditions, we prove that the corresponding sequence of optimal values converges to the global optimum of the POP.
- Our algorithmic development of the CS-TSSOS hierarchy is fully implemented in the `TSSOS` tool [27]. The most recent version of `TSSOS` has been released within the Julia programming language, which is freely available online and documented.[1] With `TSSOS`, the accuracy and scalability of the CS-TSSOS hierarchy are evaluated on several large-scale benchmarks coming from the continuous and combinatorial optimization literature. In particular, numerical experiments demonstrate that the CS-TSSOS hierarchy is able to handle challenging Max-Cut instances and optimal power flow instances with several thousand (≃ 6,000) variables on a laptop whenever appropriate sparsity patterns are accessible. We remark that the CS-TSSOS framework has been recently extended to handle noncommutative polynomial optimization [45] and complex polynomial optimization [44].

The rest of this aticle is organized as follows: in Section 2, we provide preliminary background on SOS polynomials, the moment-SOS hierarchy, correlative sparsity and the (Chordal-)TSSOS hierarchy. In Section 3, we explain how to combine correlative sparsity and term sparsity to obtain a two-level CS-TSSOS hierarchy. Its convergence is analyzed in Section 4. Eventually, we provide numerical experiments for large-scale POP instances in Section 5. Discussions and conclusions are made in Section 6.

[1] https://github.com/wangjie212/TSSOS.

2 NOTATION AND PRELIMINARIES

2.1 Notation and SOS Polynomials

Let $\mathbf{x} = (x_1, \dots, x_n)$ be a tuple of variables and $\mathbb{R}[\mathbf{x}] = \mathbb{R}[x_1, \dots, x_n]$ be the ring of real n-variate polynomials. For $d \in \mathbb{N}$, the set of polynomials of degree no more than $2d$ is denoted by $\mathbb{R}_{2d}[\mathbf{x}]$. 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 $\text{supp}(f) := \{\boldsymbol{\alpha} \in \mathscr{A} \mid f_{\boldsymbol{\alpha}} \neq 0\}$. We 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)** matrices is denoted by $\mathbf{S}_+^r$. A matrix $A \in \mathbf{S}_+^r$ is written as $A \succeq 0$. For matrices $A, B \in \mathbf{S}^r$, let $\langle A, B \rangle \in \mathbb{R}$ denote the trace inner-product, defined by $\langle A, B \rangle = \text{Tr}(A^T B)$, and let $A \circ B \in \mathbf{S}^r$ denote the Hadamard product, defined by $[A \circ B]_{ij} = A_{ij} B_{ij}$. For $d \in \mathbb{N}$, let $\mathbb{N}_d^n := \{\boldsymbol{\alpha} = (\alpha_i)_{i=1}^n \in \mathbb{N}^n \mid \sum_{i=1}^n \alpha_i \leq d\}$. For $\boldsymbol{\beta} = (\beta_i) \in \mathbb{N}^n, \boldsymbol{\gamma} = (\gamma_i) \in \mathbb{N}^n$, let $\boldsymbol{\beta} + \boldsymbol{\gamma} := (\beta_i + \gamma_i) \in \mathbb{N}^n$. For $\boldsymbol{\alpha} \in \mathbb{N}^n, \mathscr{A}, \mathscr{B} \subseteq \mathbb{N}^n$, let $\boldsymbol{\alpha} + \mathscr{B} := \{\boldsymbol{\alpha} + \boldsymbol{\beta} \mid \boldsymbol{\beta} \in \mathscr{B}\}$ and $\mathscr{A} + \mathscr{B} := \{\boldsymbol{\alpha} + \boldsymbol{\beta} \mid \boldsymbol{\alpha} \in \mathscr{A}, \boldsymbol{\beta} \in \mathscr{B}\}$. For $m \in \mathbb{N} \backslash \{0\}$, let $[m] := \{1, 2, \dots, m\}$.

Given a polynomial $f(\mathbf{x}) \in \mathbb{R}[\mathbf{x}]$, if there exist polynomials $f_1(\mathbf{x}), \dots, f_t(\mathbf{x})$ such that $f(\mathbf{x}) = \sum_{i=1}^t f_i(\mathbf{x})^2$, then we call $f(\mathbf{x})$ a *sum of squares (SOS)* polynomial. The set of SOS polynomials is denoted by $\Sigma[\mathbf{x}]$. Assume that $f \in \Sigma_{2d}[\mathbf{x}] := \Sigma[\mathbf{x}] \cap \mathbb{R}_{2d}[\mathbf{x}]$ and $\mathbf{x}^{\mathbb{N}_d^n}$ is the *standard monomial basis*. Then, the SOS condition for f is equivalent to the existence of a PSD matrix Q, which is called a *Gram matrix* [32], such that $f = (\mathbf{x}^{\mathbb{N}_d^n})^T Q \mathbf{x}^{\mathbb{N}_d^n}$. For convenience, we abuse notation in the sequel and denote by $\mathbb{N}_d^n$ instead of $\mathbf{x}^{\mathbb{N}_d^n}$ the standard monomial basis and use the exponent $\boldsymbol{\alpha}$ to represent a monomial $\mathbf{x}^{\boldsymbol{\alpha}}$.

2.2 The Moment-SOS Hierarchy for POPs

With $\mathbf{y} = (y_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}}$ 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 $d \in \mathbb{N}$, the *moment matrix* $M_d(\mathbf{y})$ of order d associated with $\mathbf{y}$ is the matrix with rows and columns indexed by the standard monomial basis $\mathbb{N}_d^n$ such that

$$M_d(\mathbf{y})_{\boldsymbol{\beta}\boldsymbol{\gamma}} := L_{\mathbf{y}}(\mathbf{x}^{\boldsymbol{\beta}} \mathbf{x}^{\boldsymbol{\gamma}}) = y_{\boldsymbol{\beta}+\boldsymbol{\gamma}}, \quad \forall \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{N}_d^n.$$

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

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

Consider the POP (Q) defined by (1.1) and (1.2). Throughout this article, let $d_j := \lceil \deg(g_j)/2 \rceil, j = 1, \dots, m$ and $d_{\min} := \max\{\lceil \deg(f)/2 \rceil, d_1, \dots, d_m\}$. Then the moment hierarchy for (Q) indexed by integer $d \geq d_{\min}$ is defined by ([20]):

$$(\mathrm{Q}_d): \begin{cases} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & M_d(\mathbf{y}) \succeq 0, \\ & M_{d-d_j}(g_j \mathbf{y}) \succeq 0, \quad j = 1, \dots, m, \\ & y_{\mathbf{0}} = 1. \end{cases} \tag{2.1}$$

We call d the *relaxation order*.

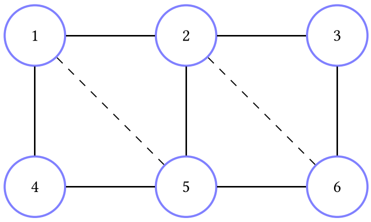


Fig. 1. An example of chordal extension.

For the sake of convenience, we set $g_0 := 1$ and $d_0 := 0$ throughout this article. For each j, writing $M_{d-d_j}(g_j\mathbf{y}) = \sum_{\boldsymbol{\alpha}} D^j_{\boldsymbol{\alpha}} y_{\boldsymbol{\alpha}}$ for appropriate symmetry matrices $\{D^j_{\boldsymbol{\alpha}}\}$, the dual of (2.1) reads as

$$(\mathrm{Q}_d)^* : \quad \begin{cases} \sup & \rho \\ \text{s.t.} & \sum_{j=0}^{m} \langle Q_j, D^j_{\boldsymbol{\alpha}} \rangle + \rho\delta_{\mathbf{0}\boldsymbol{\alpha}} = f_{\boldsymbol{\alpha}}, \quad \forall \boldsymbol{\alpha} \in \mathbb{N}^n_{2d}, \\ & Q_j \succeq 0, \quad j = 0, \ldots, m, \end{cases} \tag{2.2}$$

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

2.3 Chordal Graphs and Sparse Matrices

In this subsection, we recall some basic results on chordal graphs and sparse matrices which are crucial for our subsequent development. Our notation and definitions here mostly follow from [39].

An *(undirected) graph* $G(V, E)$ or simply G consists of a set of nodes V and a set of edges $E \subseteq \{\{v_i, v_j\} \mid v_i \neq v_j, (v_i, v_j) \in V \times V\}$. For a graph G, we use $V(G)$ and $E(G)$ to indicate the node set of G and the edge set of G, respectively. The *adjacency matrix* of a graph G is denoted by B_G for which we put ones on its diagonal. For two graphs G, H, we say that G is a *subgraph* of H, denoted by $G \subseteq H$, if both $V(G) \subseteq V(H)$ and $E(G) \subseteq E(H)$ hold.

Definition 2.1. A graph is called a *chordal graph* if all its cycles of length at least four have a chord.[2]

The notion of chordal graphs plays an important role in sparse matrix theory. Any non-chordal graph $G(V, E)$ can be always extended to a chordal graph $G'(V, E')$ by adding appropriate edges to E, which is called a *chordal extension* of $G(V, E)$. As an example, in Figure 1 the two dashed edges are added to obtain a chordal extension. The chordal extension of G is usually not unique and the symbol G' is used to represent any specific chordal extension of G throughout this article. For graphs $G \subseteq H$, we assume that $G' \subseteq H'$ always holds in this article.

A *complete graph* is a graph in which any two nodes have an edge. A *clique* of a graph is a subset of nodes that induces a complete subgraph. A *maximal clique* is a clique that is not contained in any other clique. It is known that for a chordal graph, its maximal cliques can be enumerated efficiently in linear time in terms of the number of nodes and edges. See, e.g., [7], [11], and [12] for the details.

From now on, we consider graphs with the node set $V \subseteq \mathbb{N}^n$. Given a graph $G(V, E)$, a symmetric matrix Q with rows and columns indexed 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.

[2] A chord is an edge that joins two nonconsecutive nodes in a cycle.

For a matrix in $\mathbf{S}_G$, its submatrices/blocks indexed by the maximal cliques of G play a crucial role, especially in the case when G is a chordal graph (see Theorems 2.3 and 2.4). The maximal size of blocks is the maximal size of maximal cliques of G, namely, the *clique number* of G.

Remark 2.2. For a graph G, among all chordal extensions of G, there is a particular one G' which makes every connected component of G to be a complete subgraph. Accordingly, the matrix with sparsity pattern G' is block diagonal (after an appropriate permutation on rows and columns): each block corresponds to a connected component of G. We call this chordal extension the *maximal* chordal extension. In this article, we only consider chordal extensions that are subgraphs of the maximal chordal extension.

Given a graph $G(V, E)$, the PSD matrices with sparsity pattern G form a convex cone

$$\mathbf{S}_+^{|V|} \cap \mathbf{S}_G = \{Q \in \mathbf{S}_G \mid Q \succeq 0\}. \tag{2.3}$$

Once the sparsity pattern graph $G(V, E)$ is a chordal graph, the cone $\mathbf{S}_+^{|V|} \cap \mathbf{S}_G$ can be decomposed as a sum of simple convex cones thanks to the following theorem and hence the related optimization problem can be solved more efficiently.

Theorem 2.3 ([1], Theorem 2.3). *Let $G(V, E)$ be a chordal graph and assume that $C_1, \ldots, C_t$ are the list of maximal cliques of $G(V, E)$. Then a matrix $Q \in \mathbf{S}_+^{|V|} \cap \mathbf{S}_G$ if and only if Q can be written as $Q = \sum_{i=1}^t Q_i$, where $Q_i \in \mathbf{S}_+^{|V|}$ has nonzero entries only with row and column indices coming from C_i for $i = 1, \ldots, t$.*

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 $Q \in \mathbf{S}^{|V|}$,

$$\Pi_G(Q)_{\boldsymbol{\beta}\boldsymbol{\gamma}} = \begin{cases} Q_{\boldsymbol{\beta}\boldsymbol{\gamma}}, & \text{if } \boldsymbol{\beta} = \boldsymbol{\gamma} \text{ or } \{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \in E, \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

The set $\Pi_G(\mathbf{S}_+^{|V|})$ denotes matrices that are projections of PSD matrices onto $\mathbf{S}_G$. More precisely,

$$\Pi_G(\mathbf{S}_+^{|V|}) = \{\Pi_G(Q) \mid Q \in \mathbf{S}_+^{|V|}\}. \tag{2.5}$$

One can easily check that the cone $\Pi_G(\mathbf{S}_+^{|V|})$ and the cone $\mathbf{S}_+^{|V|} \cap \mathbf{S}_G$ form a pair of dual cones in $\mathbf{S}_G$ (see [39, Chapter 10]). Moreover, for a chordal graph G, the decomposition result for matrices in $\mathbf{S}_+^{|V|} \cap \mathbf{S}_G$ given in Theorem 2.3 leads to the following characterization of matrices in the cone $\Pi_G(\mathbf{S}_+^{|V|})$.

Theorem 2.4 ([14], Theorem 7). *Let $G(V, E)$ be a chordal graph and assume that $C_1, \ldots, C_t$ are the list of maximal cliques of $G(V, E)$. Then, a matrix $Q \in \Pi_G(\mathbf{S}_+^{|V|})$ if and only if $Q[C_i] \succeq 0$ for $i = 1, \ldots, t$, where $Q[C_i]$ denotes the principal submatrix of Q indexed by the clique C_i.*

By Theorem 2.4, to check $Q \in \Pi_G(\mathbf{S}_+^{|V|})$, it suffices to check the positive semidefiniteness of certain blocks of Q. For more details on chordal graphs and sparse matrices, the reader may refer to [39].

2.4 Correlative Sparsity

To exploit correlative sparsity in the moment-SOS hierarchy for POPs, one proceeds in two steps: (1) decompose the set of variables into cliques according to the links between variables emerging in the input polynomial system, and (2) construct a sparse moment-SOS hierarchy with respect to the former decomposition of variables [40].

More concretely, we define the *correlative sparsity pattern (csp) graph* associated with POP (1.1) to be the graph G^{csp} with nodes $V = [n]$ and edges E satisfying $\{i, j\} \in E$ if one of following holds:

(i) there exists $\boldsymbol{\alpha} \in \mathrm{supp}(f)$ s.t. $\alpha_i > 0, \alpha_j > 0$;
(ii) there exists $k \in [m]$ such that $x_i, x_j \in \mathrm{var}(g_k)$, where $\mathrm{var}(g_k)$ is the set of variables involved in g_k.

Let $(G^{\mathrm{csp}})'$ be a chordal extension of G^{csp} and $\{I_l\}_{l=1}^p$ be the list of maximal cliques of $(G^{\mathrm{csp}})'$ with $n_l := |I_l|$. Let $\mathbb{R}[\mathbf{x}(I_l)]$ denote the ring of polynomials in the n_l variables $\mathbf{x}(I_l) = \{x_i \mid i \in I_l\}$. We then partition the constraint polynomials $g_1, \dots, g_m$ into groups $\{g_j \mid j \in J_l\}, l = 1, \dots, p$ which satisfy

(i) $J_1, \dots, J_p \subseteq [m]$ are pairwise disjoint and $\cup_{l=1}^p J_l = [m]$;
(ii) for any $j \in J_l$, $\mathrm{var}(g_j) \subseteq I_l$, $l = 1, \dots, p$.

Next, with $l \in \{1, \dots, p\}$ fixed, for $d \in \mathbb{N}$ and $g \in \mathbb{R}[\mathbf{x}(I_l)]$, let $M_d(\mathbf{y}, I_l)$ (respectively, $M_d(g\mathbf{y}, I_l)$) be the moment (respectively, localizing) submatrix obtained from $M_d(\mathbf{y})$ (respectively, $M_d(g\mathbf{y})$) by retaining only those rows and columns indexed by $\boldsymbol{\beta} = (\beta_i) \in \mathbb{N}_d^n$ of $M_d(\mathbf{y})$ (respectively, $M_d(g\mathbf{y})$) with $\mathrm{supp}(\boldsymbol{\beta}) \subseteq I_l$, where $\mathrm{supp}(\boldsymbol{\beta}) := \{i \mid \beta_i \neq 0\}$.

Then, with $d \geq d_{\min}$, the moment hierarchy based on correlative sparsity for POP (1.1) is defined as

$$(\mathrm{Q}_d^{\mathrm{cs}}): \quad \begin{cases} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & M_d(\mathbf{y}, I_l) \succeq 0, \quad l = 1, \dots, p, \\ & M_{d-d_j}(g_j\mathbf{y}, I_l) \succeq 0, \quad j \in J_l, l = 1 \dots, p, \\ & y_{\mathbf{0}} = 1, \end{cases} \tag{2.6}$$

with optimal value denoted by ρ_d. In the following, we refer to $(\mathrm{Q}_d^{\mathrm{cs}})$ (2.6) as the *CSSOS* hierarchy for POP (1.1).

Remark 2.5. As shown in [21] under some compactness assumption, the sequence $(\rho_d)_{d \geq d_{\min}}$ monotonically converges to the global optimum ρ^* of POP (1.1).

2.5 Term Sparsity

In contrast to the correlative sparsity pattern which focuses on links between *variables*, the term sparsity pattern focuses on links between *monomials* (or terms). To exploit term sparsity in the moment-SOS hierarchy one also proceeds in two steps: (1) decompose each involved monomial basis into blocks according to the links between monomials emerging in the input polynomial system, and (2) construct a sparse moment-SOS hierarchy with respect to the former decomposition of monomial bases [46, 47].

More concretely, let $\mathscr{A} = \mathrm{supp}(f) \cup \bigcup_{j=1}^m \mathrm{supp}(g_j)$ and $\mathbb{N}_{d-d_j}^n$ be the standard monomial basis for $j = 0, \dots, m$ (recall $d_0 = 0$). Fixing a relaxation order $d \geq d_{\min}$, we define the *term sparsity pattern (tsp) graph* associated with POP (1.1) or the support set $\mathscr{A}$, to be the graph G_d^{tsp} with node set $V_{d,0} := \mathbb{N}_d^n$ and edge set

$$E(G_d^{\mathrm{tsp}}) := \{\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \mid \boldsymbol{\beta} \neq \boldsymbol{\gamma} \in V_{d,0}, \boldsymbol{\beta} + \boldsymbol{\gamma} \in \mathscr{A} \cup (2\mathbb{N})^n\}, \tag{2.7}$$

where $(2\mathbb{N})^n := \{2\boldsymbol{\alpha} \mid \boldsymbol{\alpha} \in \mathbb{N}^n\}$.

For a graph $G(V, E)$ with $V \subseteq \mathbb{N}^n$, let $\mathrm{supp}(G) := \{\boldsymbol{\beta} + \boldsymbol{\gamma} \mid \boldsymbol{\beta} = \boldsymbol{\gamma} \text{ or } \{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \in E\}$. We define the graphs $G_{d,0}^{(0)} := G_d^{\mathrm{tsp}}$ and, for $j = 1, \dots, m$, $G_{d,j}^{(0)}$ is the empty graph with node set $V_{d,j} := \mathbb{N}_{d-d_j}^n$ and empty edge set. Note that $\mathrm{supp}(G_{d,0}^{(0)}) = \mathscr{A} \cup 2\mathbb{N}_d^n$ and $\mathrm{supp}(G_{d,j}^{(0)}) = \emptyset$ for $j \geq 1$. Now for each $j \in \{0\} \cup [m]$, we iteratively define an ascending chain of graphs $(G_{d,j}^{(k)}(V_{d,j}, E_{d,j}^{(k)}))_{k \geq 1}$. To this end, we start with the initial graph $G_{d,j}^{(0)}$ and each iteration consists of two successive operations:

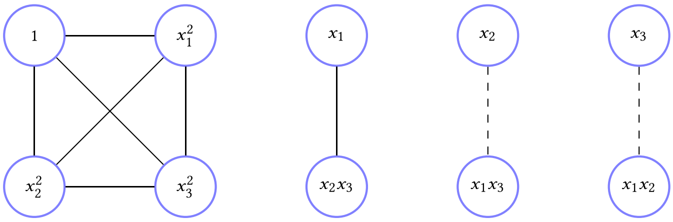


Fig. 2. The support extension of G.

(1) *Support Extension.* Define $F_{d,j}^{(k)}$ to be the graph with nodes $V_{d,j}$ and with (recall $g_0 = 1$)

$$E\left(F_{d,j}^{(k)}\right) = \Big\{\{\boldsymbol{\beta},\boldsymbol{\gamma}\} \mid \boldsymbol{\beta} \neq \boldsymbol{\gamma} \in V_{d,j}, \\ (\boldsymbol{\beta}+\boldsymbol{\gamma}+\operatorname{supp}(g_j)) \cap \left(\cup_{i=0}^m \operatorname{supp}\left(G_{d,i}^{(k-1)}\right)\right) \neq \emptyset\Big\}, \quad j \in \{0\} \cup [m]. \tag{2.8}$$

(2) Chordal Extension. Let

$$G_{d,j}^{(k)} := \left(F_{d,j}^{(k)}\right)', \quad j \in \{0\} \cup [m]. \tag{2.9}$$

Note that $F_{d,0}^{(1)}$ has edges $\{\boldsymbol{\beta},\boldsymbol{\gamma}\}$ with $\boldsymbol{\beta}+\boldsymbol{\gamma} \in \mathscr{A} \cup (2\mathbb{N})^n$. To summarise, the iterative process is

$$G_{d,j}^{(0)} \to \cdots \to G_{d,j}^{(k-1)} \xrightarrow{\text{support extension}} F_{d,j}^{(k)} \xrightarrow{\text{chordal extension}} G_{d,j}^{(k)} \to \cdots,$$

for each $j \in \{0\} \cup [m]$.

Example 2.6 (Support Extension). Assume $m = 0, d = 2$, and consider the graph G with solid edges shown in Figure 2. Then, by support extension, the two dashed edges are added to G for $x_1x_2x_3 \in \operatorname{supp}(G)$.

Let $r_j := |\mathbb{N}^n_{d-d_j}| = \binom{n+d-d_j}{d-d_j}, j = 0, \dots, m$. Then, with $d \geq d_{\min}$ and $k \geq 1$, the moment hierarchy based on term sparsity for POP (1.1) is defined as

$$(\mathrm{Q}^{\mathrm{ts}}_{d,k}): \begin{cases} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & B_{G_{d,0}^{(k)}} \circ M_d(\mathbf{y}) \in \Pi_{G_{d,0}^{(k)}}(\mathbf{S}_+^{r_0}), \\ & B_{G_{d,j}^{(k)}} \circ M_{d-d_j}(g_j\mathbf{y}) \in \Pi_{G_{d,j}^{(k)}}(\mathbf{S}_+^{r_j}), \quad j = 1, \dots, m, \\ & y_0 = 1. \end{cases} \tag{2.10}$$

The notation $B_G \circ A$ in (2.10) refers to a matrix whose $(\boldsymbol{\beta},\boldsymbol{\gamma})$-entry is $A_{\boldsymbol{\beta}\boldsymbol{\gamma}}$ if $\boldsymbol{\beta} = \boldsymbol{\gamma}$ or $\{\boldsymbol{\beta},\boldsymbol{\gamma}\} \in E(G)$, and 0 otherwise. We call k the *sparse order* and, in the remainder of this article, the *TSSOS* hierarchy for POP (1.1) refers to the hierarchy $(\mathrm{Q}^{\mathrm{ts}}_{d,k})$.

Remark 2.7. In $(Q^{ts}_{d,k})$, one has the freedom to choose a specific chordal extension for any involved graph $G_{d,j}^{(k)}$. As shown in [47], if one chooses the maximal chordal extension then with d fixed, the resulting sequence of optimal values of the TSSOS hierarchy (as k increases) monotonically converges in finitely many steps to the optimal value of the corresponding dense moment relaxation for POP (1.1).

3 THE CS-TSSOS HIERARCHY

When applicable, one can significantly improve the scalability of the moment-SOS hierarchy by exploiting correlative sparsity or term sparsity. For large-scale POPs, it is then natural to ask whether one can combine correlative sparsity and term sparsity to further reduce the size of SDPs involved in the moment-SOS hierarchy and to improve its scalability even more. As we shall see in the following sections, the answer is affirmative.

3.1 The CS-TSSOS Hierarchy for General POPs

Let us continue considering POP (1.1).[3] The first natural idea to combine correlative sparsity and term sparsity would be to apply the TSSOS hierarchy for each subsystem (involving one variable clique) *separately*, once the cliques have been obtained from the csp graph of POP (1.1). However, with this naive approach convergence may be lost and, in the following, we take extra care to avoid this annoying consequence.

Let G^{csp} be the csp graph associated with POP (1.1), $(G^{\text{csp}})'$ a chordal extension of G^{csp} and $\{I_l\}_{l=1}^{p}$ be the list of maximal cliques of $(G^{\text{csp}})'$ with $n_l := |I_l|$. As in Section 2.4, the set of variables $\mathbf{x}$ is decomposed into $\mathbf{x}(I_1), \mathbf{x}(I_2), \dots, \mathbf{x}(I_p)$. Let $J_1, \dots, J_p$ be defined as in Section 2.4.

Now we apply the term sparsity pattern to each subsystem involving variables $\mathbf{x}(I_l)$, $l = 1, \dots, p$ respectively as follows: Let

$$\mathscr{A} := \text{supp}(f) \cup \bigcup_{j=1}^{m} \text{supp}(g_j) \text{ and } \mathscr{A}_l := \{\boldsymbol{\alpha} \in \mathscr{A} \mid \text{supp}(\boldsymbol{\alpha}) \subseteq I_l\} \tag{3.1}$$

for $l = 1, \dots, p$. As before, we set $d_{\min} := \max\{\lceil \deg(f)/2 \rceil, d_1, \dots, d_m\}$, $d_0 := 0$ and $g_0 := 1$. Fix a relaxation order $d \geq d_{\min}$ and let $\mathbb{N}^{n_l}_{d-d_j}$ be the standard monomial basis for $j \in \{0\} \cup J_l$, $l = 1, \dots, p$. Let $G^{\text{tsp}}_{d,l}$ be the tsp graph with nodes $\mathbb{N}^{n_l}_d$ associated with $\mathscr{A}_l$ defined as in Section 2.5, i.e., its node set is $\mathbb{N}^{n_l}_d$ and $\{\boldsymbol{\beta}, \boldsymbol{\gamma}\}$ is an edge if $\boldsymbol{\beta} + \boldsymbol{\gamma} \in \mathscr{A}_l \cup (2\mathbb{N})^{n_l}$. Note that we embed $\mathbb{N}^{n_l}$ into $\mathbb{N}^n$ via the map $\boldsymbol{\alpha} = (\alpha_i) \in \mathbb{N}^{n_l} \mapsto \boldsymbol{\alpha}' = (\alpha_i') \in \mathbb{N}^n$ which satisfies

$$\alpha_i' = \begin{cases} \alpha_i, & \text{if } i \in I_l, \\ 0, & \text{otherwise.} \end{cases}$$

Let us define $G^{(0)}_{d,l,0} := G^{\text{tsp}}_{d,l}$ and $G^{(0)}_{d,l,j}$, $j \in J_l$, $l = 1, \dots, p$ are all empty graphs with nodes $\mathbb{N}^{n_l}_{d-d_j}$. Next, for an integer $k \geq 1$, for each $j \in \{0\} \cup J_l$, $l = 1, \dots, p$, we iteratively define an ascending chain of graphs $(G^{(k)}_{d,l,j}(V_{d,l,j}, E^{(k)}_{d,l,j}))_{k \geq 1}$ with $V_{d,l,j} := \mathbb{N}^{n_l}_{d-d_j}$ via two successive operations:

(1) *Support Extension.* Define $F^{(k)}_{d,l,j}$ to be the graph with nodes $V_{d,l,j}$ and with

$$E\left(F^{(k)}_{d,l,j}\right) = \{\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \mid \boldsymbol{\beta} \neq \boldsymbol{\gamma} \in V_{d,l,j}, (\boldsymbol{\beta} + \boldsymbol{\gamma} + \text{supp}(g_j)) \cap \mathscr{C}^{(k-1)}_d \neq \emptyset\}, \tag{3.2}$$

where

$$\mathscr{C}^{(k-1)}_d := \bigcup_{l=1}^{p} \left(\cup_{j \in \{0\} \cup J_l} \left(\text{supp}(g_j) + \text{supp}\left(G^{(k-1)}_{d,l,j}\right)\right)\right). \tag{3.3}$$

(2) *Chordal Extension.* Let

$$G^{(k)}_{d,l,j} := \left(F^{(k)}_{d,l,j}\right)', \quad j \in \{0\} \cup J_l, l = 1, \dots, p. \tag{3.4}$$

[3] Though we only include inequality constraints in the definition of $\mathbf{K}$ (1.2) for the sake of simplicity, equality constraints can be treated in a similar way.

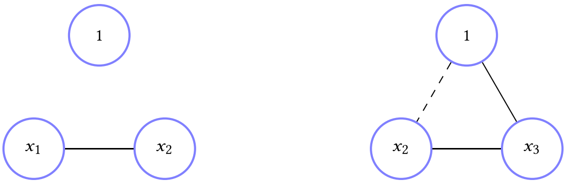


Fig. 3. The tsp graphs of Example 3.1. The dashed edge is added after the maximal chordal extension.

Example 3.1. Let $f = 1 + x_1^2 + x_2^2 + x_3^2 + x_1x_2 + x_2x_3 + x_3$ and consider the unconstrained POP: $\min\{f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\}$. We have $n = 3, m = 0$ and take the relaxation order $d = d_{\min} = 1$. The variables are decomposed into two cliques: $\{x_1, x_2\}$ and $\{x_2, x_3\}$. The tsp graphs with respect to these two cliques are illustrated in Figure 3. The left graph corresponds to the first clique: x_1 and x_2 are connected because of the term x_1x_2. The right graph corresponds to the second clique: 1 and x_3 are connected because of the term x_3; x_2 and x_3 are connected because of the term x_2x_3. If we apply the TSSOS hierarchy (using the maximal chordal extension in (3.4)) separately for each clique, then the graph sequences $(G_{1,l}^{(k)})_{k\ge1}, l = 1, 2$ (the subscript j is omitted here since there is no constraint) stabilize at $k = 1$. However, the added (dashed) edge in the right graph corresponds to the monomial x_2, which only involves the variable x_2 belonging to the first clique. Hence we need to add the edge connecting 1 and x_2 to the left graph in order to get the guarantee of convergence as we shall see in Section 4.1. Consequently, the graph sequences $(G_{1,l}^{(k)})_{k\ge1}, l = 1, 2$ stabilize at $k = 2$.

Let $r_{l,j} := |\mathbb{N}^{n_l}_{d-d_j}| = \binom{n_l+d-d_j}{d-d_j}$ for all l, j. Then, with $k \ge 1$, the moment hierarchy based on correlative-term sparsity for POP (1.1) is defined as:

$$(\mathrm{Q}^{\text{cs-ts}}_{d,k}): \quad \begin{cases} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & B_{G_{d,l,0}^{(k)}} \circ M_d(\mathbf{y}, I_l) \in \Pi_{G_{d,l,0}^{(k)}}(\mathbf{S}_+^{r_{l,0}}), \quad l = 1, \dots, p, \\ & B_{G_{d,l,j}^{(k)}} \circ M_{d-d_j}(g_j\mathbf{y}, I_l) \in \Pi_{G_{d,l,j}^{(k)}}(\mathbf{S}_+^{r_{l,j}}), \quad j \in J_l, l = 1, \dots, p, \\ & y_{\mathbf{0}} = 1, \end{cases} \tag{3.5}$$

with optimal value denoted by $\rho_d^{(k)}$. Recall that the optimal value of the CSSOS hierarchy $(\mathrm{Q}^{\text{cs}}_d)$ (2.6) for POP (1.1) is denoted by ρ_d.

Proposition 3.2. *Fixing a relaxation order $d \ge d_{\min}$, the sequence $(\rho_d^{(k)})_{k\ge1}$ is monotonically non-decreasing and $\rho_d^{(k)} \le \rho_d$ for all k.*

Proof. By construction, we have $G_{d,l,j}^{(k)} \subseteq G_{d,l,j}^{(k+1)}$ for all d, l, j and all k. It follows that each maximal clique of $G_{d,l,j}^{(k)}$ is a subset of some maximal clique of $G_{d,l,j}^{(k+1)}$. Hence, by Theorem 2.4, $(\mathrm{Q}^{\text{cs-ts}}_{d,k})$ is a relaxation of $(\mathrm{Q}^{\text{cs-ts}}_{d,k+1})$ and is clearly also a relaxation of $(\mathrm{Q}^{\text{cs}}_d)$. Therefore, $(\rho_d^{(k)})_{k\ge1}$ is monotonically non-decreasing and $\rho_d^{(k)} \le \rho_d$ for all k. □

Proposition 3.3. *Fixing a sparse order $k \ge 1$, the sequence $(\rho_d^{(k)})_{d\ge d_{\min}}$ is monotonically non-decreasing.*

Proof. The conclusion follows if we can show that $G_{d,l,j}^{(k)} \subseteq G_{d+1,l,j}^{(k)}$ for all d, l, j, k since by Theorem 2.4 this implies that $(\mathrm{Q}^{\text{cs-ts}}_{d,k})$ is a relaxation of $(\mathrm{Q}^{\text{cs-ts}}_{d+1,k})$. Let us prove $G_{d,l,j}^{(k)} \subseteq G_{d+1,l,j}^{(k)}$ by induction on k. For $k = 1$, from (2.7), we have $G^{(0)}_{d,l,0} = G^{\text{tsp}}_{d,l} \subseteq G^{\text{tsp}}_{d+1,l} = G^{(0)}_{d+1,l,0}$, which together with (3.2) and (3.3) implies that $F^{(1)}_{d,l,j} \subseteq F^{(1)}_{d+1,l,j}$ for $j \in \{0\} \cup J_l, l = 1, \ldots, p$. It then follows that $G^{(1)}_{d,l,j} = (F^{(1)}_{d,l,j})' \subseteq (F^{(1)}_{d+1,l,j})' = G^{(1)}_{d+1,l,j}$. Now assume that $G^{(k)}_{d,l,j} \subseteq G^{(k)}_{d+1,l,j}$, $j \in \{0\} \cup J_l, l = 1, \ldots, p$, holds for some $k \geq 1$. Then, by (3.2) and (3.3) and by the induction hypothesis, we have $F^{(k+1)}_{d,l,j} \subseteq F^{(k+1)}_{d+1,l,j}$ for $j \in \{0\} \cup J_l, l = 1, \ldots, p$. Thus, $G^{(k+1)}_{d,l,j} = (F^{(k+1)}_{d,l,j})' \subseteq (F^{(k+1)}_{d+1,l,j})' = G^{(k+1)}_{d+1,l,j}$ which completes the induction. □

From Propositions 3.2 and 3.3, we deduce the following two-level hierarchy of lower bounds for the optimum ρ^* of (Q) (1.1):

$$\begin{array}{ccccccc}
\rho^{(1)}_{d_{\min}} & \leq & \rho^{(2)}_{d_{\min}} & \leq & \cdots & \leq & \rho_{d_{\min}} \\
\rotatebox[origin=c]{-90}{$\leq$} & & \rotatebox[origin=c]{-90}{$\leq$} & & & & \rotatebox[origin=c]{-90}{$\leq$} \\
\rho^{(1)}_{d_{\min}+1} & \leq & \rho^{(2)}_{d_{\min}+1} & \leq & \cdots & \leq & \rho_{d_{\min}+1} \\
\rotatebox[origin=c]{-90}{$\leq$} & & \rotatebox[origin=c]{-90}{$\leq$} & & & & \rotatebox[origin=c]{-90}{$\leq$} \\
\vdots & & \vdots & & \vdots & & \vdots \\
\rotatebox[origin=c]{-90}{$\leq$} & & \rotatebox[origin=c]{-90}{$\leq$} & & & & \rotatebox[origin=c]{-90}{$\leq$} \\
\rho^{(1)}_{d} & \leq & \rho^{(2)}_{d} & \leq & \cdots & \leq & \rho_{d} \\
\rotatebox[origin=c]{-90}{$\leq$} & & \rotatebox[origin=c]{-90}{$\leq$} & & & & \rotatebox[origin=c]{-90}{$\leq$} \\
\vdots & & \vdots & & \vdots & & \vdots
\end{array} \tag{3.6}$$

The array of lower bounds (3.6) (and its associated SDP relaxations (3.5)) is what we call the *CS-TSSOS* hierarchy associated with (Q) (1.1).

Example 3.4. Let $f = 1 + \sum_{i=1}^{6} x_i^4 + x_1x_2x_3 + x_3x_4x_5 + x_3x_4x_6 + x_3x_5x_6 + x_4x_5x_6$, and consider the unconstrained POP: $\min\{f(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^n\}$. We have $n = 6, m = 0$. Let us apply the CS-TSSOS hierarchy (using the maximal chordal extension in (3.4)) to this problem by taking the relaxation order $d = d_{\min} = 2$ and the sparse order $k = 1$. First, according to the csp graph (see Figure 4), we decompose the variables into two cliques: $\{x_1, x_2, x_3\}$ and $\{x_3, x_4, x_5, x_6\}$. Figures 5 and 6 illustrate the tsp graphs for the first clique and the second clique, respectively. For the first clique, one obtains four blocks of SDP matrices with respective sizes $4, 2, 2, 2$. For the second clique, one obtains two blocks of SDP matrices with respective sizes $5, 10$. As a result, the original SDP matrix of size 28 has been reduced to six blocks of maximal size 10.

If one applies the TSSOS hierarchy (using the maximal chordal extension in (2.9)) directly to the problem by taking $d = d_{\min} = 2, k = 1$ (i.e., without decomposing variables), then the tsp graph is illustrated in Figure 7. One obtains 11 SDP blocks with respective sizes $7, 2, 2, 2, 1, 1, 1, 1, 1, 1, 10$. Compared to the CS-TSSOS case, there are six additional blocks of size one and the two blocks with respective sizes $4, 5$ are replaced by a single block of size 7.

The CS-TSSOS hierarchy entails a tradeoff. Indeed, one has the freedom to choose a specific chordal extension for any graph involved in (3.5). This choice affects the resulting size of blocks of SDP matrices and the quality of optimal values of corresponding relaxations. Intuitively, chordal extensions with small clique numbers lead to blocks of small size and optimal values of (possibly) low quality while chordal extensions with large clique numbers lead to blocks of large size and optimal values of (possibly) high quality.

For all l, j, write $M_{d-d_j}(g_j\mathbf{y}, I_l) = \sum_{\boldsymbol{\alpha}} D^{l,j}_{\boldsymbol{\alpha}} y_{\boldsymbol{\alpha}}$ for appropriate symmetry matrices $\{D^{l,j}_{\boldsymbol{\alpha}}\}$. Then, for each $k \geq 1$, the dual of $(\text{Q}^{\text{cs-ts}}_{d,k})$ reads as:

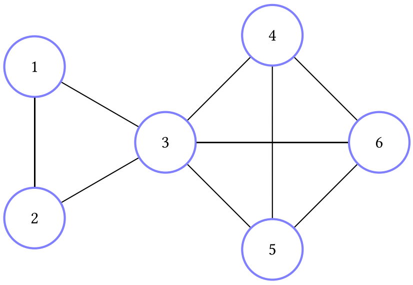


Fig. 4. The csp graph of Example 3.4.

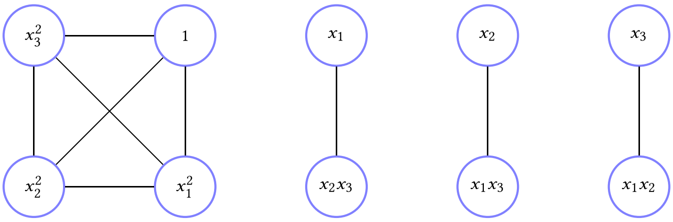


Fig. 5. The tsp graph for the first clique of Example 3.4.

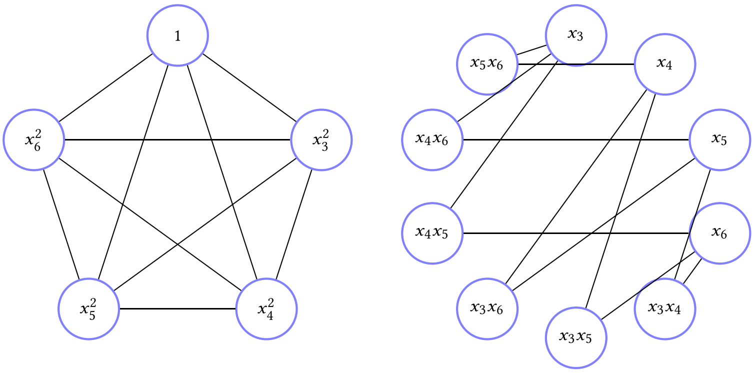


Fig. 6. The tsp graph for the second clique of Example 3.4.

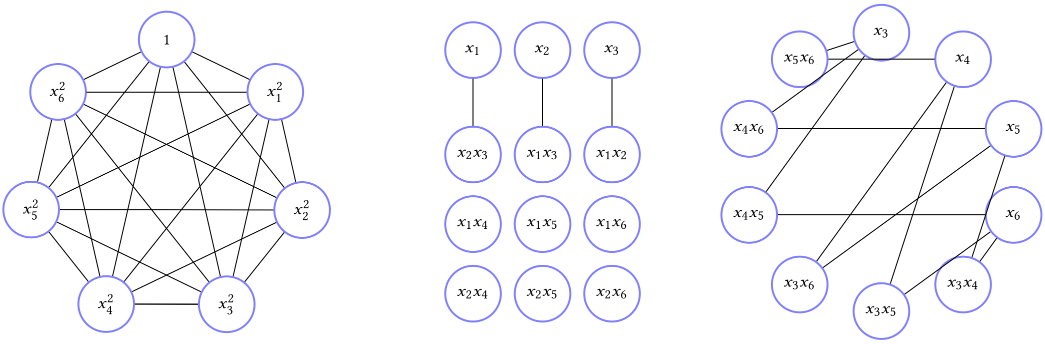


Fig. 7. The tsp graph without decomposing variables of Example 3.4.

$$(\mathrm{Q}_{d,k}^{\text{cs-ts}})^*: \quad \begin{cases} \sup & \rho \\ \text{s.t.} & \sum_{l=1}^{p}\sum_{j\in\{0\}\cup J_l}\langle Q_{l,j}, D_{\boldsymbol{\alpha}}^{l,j}\rangle + \rho\delta_{\mathbf{0}\boldsymbol{\alpha}} = f_{\boldsymbol{\alpha}}, \quad \forall\boldsymbol{\alpha}\in\mathscr{C}_d^{(k)}, \\ & Q_{l,j}\in\mathbf{S}_+^{r_{l,j}}\cap\mathbf{S}_{G_{d,l,j}^{(k)}}, \quad j\in\{0\}\cup J_l, l=1,\dots,p, \end{cases} \tag{3.7}$$

where $\mathscr{C}_d^{(k)}$ is defined in (3.3).

Proposition 3.5. *Let $f\in\mathbb{R}[\mathbf{x}]$ and $\mathbf{K}$ be as in (1.2). Assume that $\mathbf{K}$ has a nonempty interior. Then, there is no duality gap between $(Q_{d,k}^{cs\text{-}ts})$ and $(Q_{d,k}^{cs\text{-}ts})^*$ for any $d\geq d_{\min}$ and $k\geq 1$.*

Proof. By the duality theory of convex programming, this easily follows from Theorem 3.6 of [21] and Theorem 2.4. □

Note that the number of equality constraints in (3.7) is equal to the cardinality of $\mathscr{C}_d^{(k)}$. We next give a description of the elements in $\mathscr{C}_d^{(k)}$ in terms of sign symmetries.

3.2 Sign Symmetries

Definition 3.6. Given a finite set $\mathscr{A}\subseteq\mathbb{N}^n$, the *sign symmetries* of $\mathscr{A}$ are defined by all vectors $\mathbf{r}\in\mathbb{Z}_2^n:=\{0,1\}^n$ such that $\mathbf{r}^T\boldsymbol{\alpha}\equiv 0 \pmod 2$ 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 *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 3.7. By definition, the set of sign symmetries of $\mathscr{A}$ is exactly the orthogonal complement space $(\mathscr{A})_2^\perp$ in $\mathbb{Z}_2^n$, which therefore can be essentially represented by a basis of the subspace $(\mathscr{A})_2^\perp$ in $\mathbb{Z}_2^n$.

For a subset $S\subseteq\mathbb{Z}_2^n$, we say that S is *closed* under addition modulo 2 if $\mathbf{s}_1,\mathbf{s}_2\in S$ implies $(\mathbf{s}_1+\mathbf{s}_2)_2\in S$. The minimal set containing S with elements which are closed under addition modulo 2 is denoted by $\langle S\rangle_{\mathbb{Z}_2}$. It is easy to prove $\langle S\rangle_{\mathbb{Z}_2}=\{(\sum_i\mathbf{s}_i)_2 \mid \mathbf{s}_i\in S\}$ which is the subspace spanned by S in $\mathbb{Z}_2^n$.

Lemma 3.8. *Let $S\subseteq\mathbb{Z}_2^n$. Then, $(S^\perp)^\perp=\langle S\rangle_{\mathbb{Z}_2}$.*

Proof. It is immediate from the definitions. □

Lemma 3.9. *Suppose G is a graph with $V(G)\subseteq\mathbb{N}^n$. Then it holds* $(\text{supp}(G'))_2\subseteq\langle(\text{supp}(G))_2\rangle_{\mathbb{Z}_2}$.

Proof. By definition, we need to show $(\boldsymbol{\beta}+\boldsymbol{\gamma})_2\in\langle(\text{supp}(G))_2\rangle_{\mathbb{Z}_2}$ for any $\{\boldsymbol{\beta},\boldsymbol{\gamma}\}\in E(G')$. Since in the process of chordal extensions, edges are added only if two nodes belong to the same connected component, for any $\{\boldsymbol{\beta},\boldsymbol{\gamma}\}\in E(G')$ there is a path connecting $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ in G: $\{\boldsymbol{\beta},\boldsymbol{v}_1,\dots,\boldsymbol{v}_r,\boldsymbol{\gamma}\}$ with $\{\boldsymbol{\beta},\boldsymbol{v}_1\},\{\boldsymbol{v}_r,\boldsymbol{\gamma}\}\in E(G)$ and $\{\boldsymbol{v}_i,\boldsymbol{v}_{i+1}\}\in E(G), i=1,\dots,r-1$. From $(\boldsymbol{\beta}+\boldsymbol{v}_1)_2,(\boldsymbol{v}_1+\boldsymbol{v}_2)_2\in(\text{supp}(G))_2$, we deduce that $(\boldsymbol{\beta}+\boldsymbol{v}_2)_2\in\langle(\text{supp}(G))_2\rangle_{\mathbb{Z}_2}$ because $\langle(\text{supp}(G))_2\rangle_{\mathbb{Z}_2}$ is closed under addition modulo 2. Likewise, we can prove $(\boldsymbol{\beta}+\boldsymbol{v}_i)_2\in\langle(\text{supp}(G))_2\rangle_{\mathbb{Z}_2}$ for $i=3,\dots,r+1$ with $\boldsymbol{v}_{r+1}:=\boldsymbol{\gamma}$. Hence, $(\boldsymbol{\beta}+\boldsymbol{\gamma})_2\in\langle(\text{supp}(G))_2\rangle_{\mathbb{Z}_2}$ as desired. □

Proposition 3.10. *Let $\mathscr{A}$ be defined as in (3.1), $\mathscr{C}_d^{(k)}$ be defined as in (3.3) and assume that the sign symmetries of $\mathscr{A}$ are represented by the column vectors of a binary matrix, denoted by R. Then, for any $k\geq 1$ and any $\boldsymbol{\alpha}\in\mathscr{C}_d^{(k)}$, it holds $R^T\boldsymbol{\alpha}\equiv 0\,(\text{mod } 2)$. In other words, $(\mathscr{C}_d^{(k)})_2\subseteq R^\perp$, where we regard R as a set of its column vectors.*

Proof. By Lemma 3.8, we only need to prove $(\mathscr{C}_d^{(k)})_2 \subseteq \langle(\mathscr{A})_2\rangle_{\mathbb{Z}_2}$. Let us do induction on $k \geq 0$. For $k = 0$, by (3.3), $\mathscr{C}_d^{(0)} = \bigcup_{l=1}^p \operatorname{supp}(G_{d,l,0}^{(0)}) = \bigcup_{l=1}^p \operatorname{supp}(G_{d,l}^{\mathrm{tsp}}) \subseteq \bigcup_{l=1}^p (\mathscr{A}_l \cup (2\mathbb{N})^{n_l}) \subseteq \mathscr{A} \cup (2\mathbb{N})^n$. Hence, $(\mathscr{C}_d^{(0)})_2 \subseteq \langle(\mathscr{A})_2\rangle_{\mathbb{Z}_2}$. Now assume that $(\mathscr{C}_d^{(k)})_2 \subseteq \langle(\mathscr{A})_2\rangle_{\mathbb{Z}_2}$ holds for some $k \geq 0$. By (3.2), for any l, j and any $\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \in E(F_{d,l,j}^{(k+1)})$, we have $(\operatorname{supp}(g_j) + \boldsymbol{\beta} + \boldsymbol{\gamma}) \cap \mathscr{C}_d^{(k)} \neq \emptyset$, i.e., there exists $\boldsymbol{\alpha} \in \operatorname{supp}(g_j)$ such that $\boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\gamma} \in \mathscr{C}_d^{(k)}$, which implies $(\boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\gamma})_2 \in (\mathscr{C}_d^{(k)})_2$. Hence, by the induction hypothesis, $(\boldsymbol{\alpha} + \boldsymbol{\beta} + \boldsymbol{\gamma})_2 \in \langle(\mathscr{A})_2\rangle_{\mathbb{Z}_2}$. Since $\langle(\mathscr{A})_2\rangle_{\mathbb{Z}_2}$ is closed under addition modulo 2 and $(\boldsymbol{\alpha})_2 \in (\mathscr{A})_2$, we have $(\boldsymbol{\beta} + \boldsymbol{\gamma})_2 \in \langle(\mathscr{A})_2\rangle_{\mathbb{Z}_2}$. It follows $(\operatorname{supp}(F_{d,l,j}^{(k+1)}))_2 \subseteq \langle(\mathscr{A})_2\rangle_{\mathbb{Z}_2}$. Because $G_{d,l,j}^{(k+1)} = (F_{d,l,j}^{(k+1)})'$, by Lemma 3.9, we have $(\operatorname{supp}(G_{d,l,j}^{(k+1)}))_2 \subseteq \langle(\operatorname{supp}(F_{d,l,j}^{(k+1)}))_2\rangle_{\mathbb{Z}_2} \subseteq \langle(\mathscr{A})_2\rangle_{\mathbb{Z}_2}$. From this, (3.3) and the fact that $\langle(\mathscr{A})_2\rangle_{\mathbb{Z}_2}$ is closed under addition modulo 2, we then deduce the inclusion $(\mathscr{C}_d^{(k+1)})_2 \subseteq \langle(\mathscr{A})_2\rangle_{\mathbb{Z}_2}$ which completes the induction. □

Remark 3.11. Proposition 3.10 actually indicates that the block structure produced by the CS-TSSOS hierarchy is consistent with the sign symmetries of the POP.

4 CONVERGENCE ANALYSIS

4.1 Global Convergence

We next prove that if for any graph involved in (3.5), the chordal extension is chosen to be *maximal*, then for any relaxation order $d \geq d_{\min}$ the sequence of optimal values $(\rho_d^{(k)})_{k\geq 1}$ of the CS-TSSOS hierarchy converges to the optimal value ρ_d of the corresponding CSSOS hierarchy (2.6). In turn, as the relaxation order d increases, the latter sequence converges to the global optimum ρ^* of the original POP (1.1) (after adding some redundant quadratic constraints) as shown in [21].

Obviously, the sequences of graphs $(G_{d,l,j}^{(k)}(V_{d,l,j}, E_{d,l,j}^{(k)}))_{k\geq 1}$ stabilize for all $j \in \{0\} \cup J_l, l = 1, \ldots, p$ after finitely many steps. We denote the resulting stabilized graphs by $G_{d,l,j}^{(*)}, j \in \{0\} \cup J_l, l = 1, \ldots, p$ and the corresponding SDP (3.5) by $(\mathrm{Q}_{d,*}^{\text{cs-ts}})$.

Theorem 4.1. *Assume that the chordal extension in* (3.4) *is the maximal chordal extension. Then for any $d \geq d_{\min}$, the sequence $(\rho_d^{(k)})_{k\geq 1}$ converges to ρ_d in finitely many steps.*

Proof. Let $\mathbf{y} = (y_{\boldsymbol{\alpha}})$ be an arbitrary feasible solution of $(\mathrm{Q}_{d,*}^{\text{cs-ts}})$ and ρ_d^* be the optimal value of $(\mathrm{Q}_{d,*}^{\text{cs-ts}})$. Note that $\{y_{\boldsymbol{\alpha}} \mid \boldsymbol{\alpha} \in \bigcup_{l=1}^p (\cup_{j\in\{0\}\cup J_l} (\operatorname{supp}(g_j) + \operatorname{supp}(G_{d,l,j}^{(*)})))\}$ is the set of decision variables involved in $(\mathrm{Q}_{d,*}^{\text{cs-ts}})$. Let $\mathscr{R}$ be the set of decision variables involved in $(\mathrm{Q}_d^{\mathrm{cs}})$ (2.6). We then define a vector $\overline{\mathbf{y}} = (\overline{y}_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathscr{R}}$ as follows:

$$\overline{y}_{\boldsymbol{\alpha}} = \begin{cases} y_{\boldsymbol{\alpha}}, & \text{if } \boldsymbol{\alpha} \in \bigcup_{l=1}^p \left(\cup_{j\in\{0\}\cup J_l} \left(\operatorname{supp}(g_j) + \operatorname{supp}\left(G_{d,l,j}^{(*)}\right)\right)\right), \\ 0, & \text{otherwise.} \end{cases}$$

By construction and since $G_{d,l,j}^{(*)}$ stabilizes under support extension for all l, j, we have $M_{d-d_j}(g_j\overline{\mathbf{y}}, I_l) = B_{G_{d,l,j}^{(*)}} \circ M_{d-d_j}(g_j\mathbf{y}, I_l)$. As we use the maximal chordal extension in (3.4), the matrix $B_{G_{d,l,j}^{(*)}} \circ M_{d-d_j}(g_j\mathbf{y}, I_l)$ is block diagonal up to permutation (see Remark 2.2). So from $B_{G_{d,l,j}^{(*)}} \circ M_{d-d_j}(g_j\mathbf{y}, I_l) \in \Pi_{G_{d,l,j}^{(*)}}(\mathbf{S}_+^{r_{l,j}})$ it follows $M_{d-d_j}(g_j\overline{\mathbf{y}}, I_l) \succeq 0$ for $j \in \{0\} \cup J_l, l = 1, \ldots, p$. Therefore, $\overline{\mathbf{y}}$ is a feasible solution of $(\mathrm{Q}_d^{\mathrm{cs}})$ and so $L_{\mathbf{y}}(f) = L_{\overline{\mathbf{y}}}(f) \geq \rho_d$. Hence, $\rho_d^* \geq \rho_d$ since $\mathbf{y}$ is an arbitrary feasible solution of $(\mathrm{Q}_{d,*}^{\text{cs-ts}})$. By Proposition 3.2, we already have $\rho_d^* \leq \rho_d$. Therefore, $\rho_d^* = \rho_d$. □

To guarantee the global optimality, we need the following compactness assumption on the feasible set $\mathbf{K}$.

Assumption 1. *Let* $\mathbf{K}$ *be as in* (1.2). *There exists an* $M > 0$ *such that* $||\mathbf{x}||_\infty < M$ *for all* $\mathbf{x} \in \mathbf{K}$.

Because of Assumption 1, one has $||\mathbf{x}(I_l)||_2^2 \leq n_l M^2$, $l = 1, \dots, p$. Therefore, we can add the p redundant quadratic constraints

$$g_{m+l}(\mathbf{x}) := n_l M^2 - ||\mathbf{x}(I_l)||_2^2 \geq 0, \quad l = 1, \dots, p \tag{4.1}$$

in the definition (1.2) of $\mathbf{K}$ and set $m' = m + p$, so that $\mathbf{K}$ is now defined by

$$\mathbf{K} := \{\mathbf{x} \in \mathbb{R}^n \mid g_j(\mathbf{x}) \geq 0, \quad j = 1, \dots, m'\}. \tag{4.2}$$

Note that $g_{m+l} \in \mathbb{R}[\mathbf{x}(I_l)]$ for $l = 1, \dots, p$.

Then, by Theorem 3.6 in [21], the sequence $(\rho_d)_{d \geq d_{\min}}$ converges to the globally optimal value ρ^* of (Q) (1.1). So this together with Theorem 4.1 gives the global convergence of the CS-TSSOS hierarchy.

4.2 A Sparse Representation Theorem

Proceeding along Theorem 4.1, we are able to provide a *sparse representation* theorem for a polynomial positive on a compact basic semialgebraic set.

Theorem 4.2 (Sparse Representation). *Let* $f \in \mathbb{R}[\mathbf{x}]$ *and* $\mathbf{K}$ *be as in* (4.2) *with the additional quadratic constraints* (4.1). *Let* I_l, J_l *be defined as in Section* 3.1 *and* $\mathscr{A} = \operatorname{supp}(f) \cup \bigcup_{j=1}^{m'} \operatorname{supp}(g_j)$. *Assume that the sign symmetries of* $\mathscr{A}$ *are represented by the column vectors of the binary matrix* R. *If* f *is positive on* $\mathbf{K}$, *then*

$$f = \sum_{l=1}^{p} \left(\sigma_{l,0} + \sum_{j \in J_l} \sigma_{l,j} g_j \right), \tag{4.3}$$

for some polynomials $\sigma_{l,j} \in \Sigma[\mathbf{x}(I_l)], j \in \{0\} \cup J_l, l = 1, \dots, p$, *satisfying* $R^T \boldsymbol{\alpha} \equiv 0 \ (mod\ 2)$ *for any* $\boldsymbol{\alpha} \in \operatorname{supp}(\sigma_{l,j})$, *i.e.,* $(\operatorname{supp}(\sigma_{l,j}))_2 \subseteq R^\perp$, *where we regard* R *as a set of its column vectors.*

That is, (4.3) *provides a certificate of positivity of* f *on* $\mathbf{K}$.

Proof. By Corollary 3.9 of [21] (or Theorem 5 of [13]), there exist polynomials $\sigma'_{l,j} \in \Sigma[\mathbf{x}(I_l)], j \in \{0\} \cup J_l, l = 1, \dots, p$ such that

$$f = \sum_{l=1}^{p} \left(\sigma'_{l,0} + \sum_{j \in J_l} \sigma'_{l,j} g_j \right). \tag{4.4}$$

Let $d = \max\{\lceil \deg(\sigma'_{l,j} g_j)/2 \rceil : j \in \{0\} \cup J_l, l = 1, \dots, p\}$. Let $Q'_{l,j}$ be a PSD Gram matrix associated with $\sigma'_{l,j}$ and indexed by the monomial basis $\mathbb{N}^{n_l}_{d-d_j}$. Then, for all l, j, we define $Q_{l,j} \in \mathbf{S}^{r_{l,j}}$ with $r_{l,j} = \binom{n_l + d - d_j}{d - d_j}$ (indexed by $\mathbb{N}^{n_l}_{d-d_j}$) by

$$[Q_{l,j}]_{\boldsymbol{\beta}\boldsymbol{\gamma}} := \begin{cases} [Q'_{l,j}]_{\boldsymbol{\beta}\boldsymbol{\gamma}}, & \text{if } R^T(\boldsymbol{\beta} + \boldsymbol{\gamma}) \equiv 0 \ (\text{mod } 2), \\ 0, & \text{otherwise}, \end{cases}$$

and let $\sigma_{l,j} = (\mathbf{x}^{\mathbb{N}^{n_l}_{d-d_j}})^T Q_{l,j} \mathbf{x}^{\mathbb{N}^{n_l}_{d-d_j}}$. One can easily verify that $Q_{l,j}$ is block diagonal up to permutation (see also [47]) and each block is a principal submatrix of $Q'_{l,j}$. Then, the positive semidefiniteness of $Q'_{l,j}$ implies that $Q_{l,j}$ is also positive semidefinite. Thus, $\sigma_{l,j} \in \Sigma[\mathbf{x}(I_l)]$.

By construction, substituting $\sigma'_{l,j}$ with $\sigma_{l,j}$ in (4.4) boils down to removing the terms with exponents $\boldsymbol{\alpha}$ that do not satisfy $R^T \boldsymbol{\alpha} \equiv 0 \ (\text{mod } 2)$ from the right hand side of (4.4). Since any

$\boldsymbol{\alpha} \in \operatorname{supp}(f)$ satisfies $R^T\boldsymbol{\alpha} \equiv 0 \pmod 2$, this does not change the match of coefficients on both sides of the equality. Thus, we obtain

$$f = \sum_{l=1}^{p}\left(\sigma_{l,0} + \sum_{j\in J_l}\sigma_{l,j}g_j\right)$$

with the desired property. □

4.3 Extracting a Solution

In the case of dense moment-SOS relaxations, there is a standard procedure described in [15] to extract globally optimal solutions when the so-called flatness condition for the moment matrix is satisfied. This procedure was partially generalized to the correlative sparsity setting in [21, section 3.3]. However, in the combined sparsity setting, the corresponding procedure cannot be applied because we do not have complete information on the moment matrix associated with each clique. In order to extract a solution in this case, we may add a dense moment matrix of order one for each clique in (3.5):

$$(\mathrm{Q}^{\text{cs-ts}}_{d,k})' : \quad \begin{cases} \inf & L_{\mathbf{y}}(f) \\ \text{s.t.} & B_{G^{(k)}_{d,l,0}} \circ M_d(\mathbf{y}, I_l) \in \Pi_{G^{(k)}_{d,l,0}}(\mathbf{S}_+^{r_{l,0}}), \quad l = 1,\ldots,p, \\ & M_1(\mathbf{y}, I_l) \succeq 0, \quad l = 1,\ldots,p, \\ & B_{G^{(k)}_{d,l,j}} \circ M_{d-d_j}(g_j\mathbf{y}, I_l) \in \Pi_{G^{(k)}_{d,l,j}}(\mathbf{S}_+^{r_{l,j}}), \quad j \in J_l, l = 1,\ldots,p, \\ & y_{\mathbf{0}} = 1. \end{cases} \tag{4.5}$$

Let $\mathbf{y}^*$ be an optimal solution of $(\mathrm{Q}^{\text{cs-ts}}_{d,k})'$. Typically, $M_1(\mathbf{y}^*, I_l)$ (after identifying sufficiently small entries with zeros) is a block diagonal matrix (up to permutation). If, for all l, every block of $M_1(\mathbf{y}^*, I_l))$ is of rank one, then a globally optimal solution $\mathbf{x}^*$ to (Q) (1.1) which is unique up to sign symmetries can be extracted, and the global optimality is certified (see [21, Theorem 3.2]). Otherwise, the relaxation might be not exact or yield multiple global solutions.

Remark 4.3. Note that $(\mathrm{Q}^{\text{cs-ts}}_{d,k})'$ is a tighter relaxation of (Q) than $(\mathrm{Q}^{\text{cs-ts}}_{d,k})$ and so might provide a better lower bound for (Q).

5 APPLICATIONS AND NUMERICAL EXPERIMENTS

In this section, we conduct numerical experiments for the proposed CS-TSSOS hierarchy and apply it to two important classes of POPs: Max-Cut problems and **AC optimal power flow (AC-OPF)** problems. Depending on specific problems, we consider two types of chordal extensions for the term sparsity pattern: maximal chordal extensions and approximately smallest chordal extensions.[4] The tool TSSOS which executes the CS-TSSOS hierarchy (as well as the CSSOS hierarchy and the TSSOS hierarchy) is implemented in Julia. For an introduction to TSSOS, one could refer to [27]. TSSOS is available on the website:

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

In the following subsections, we compare the performances of the CSSOS approach, the TSSOS approach, the CS-TSSOS approach, and the SDSOS approach [2] (implemented in SPOT [29]). Mosek [4] is used as an SDP (in the CSSOS, TSSOS, CS-TSSOS cases) or SOCP (in the SDSOS case) solver.

[4] A smallest chordal extension is a chordal extension with the smallest clique number. Computing a smallest chordal extension is generally NP-complete. So in practice we compute approximately smallest chordal extensions instead with efficient heuristic algorithms.

Table 1. Notation

var	number of variables
cons	number of constraints
mc	maximal size of variable cliques
mb	maximal size of SDP blocks
opt	optimal value
time	running time in seconds
gap	optimality gap
CE	type of chordal extensions used in (3.4)
min	approximately smallest chordal extension
max	maximal chordal extension
0	a number whose absolute value is less than 1e-5
-	an out of memory error

All numerical examples were computed on an Intel Core i5-8265U@1.60 GHz CPU with 8 GB RAM memory. The timing includes the time required to generate the SDP/SOCP and the time spent to solve it. The notations used in this section are listed in Table 1.

5.1 Benchmarks for Unconstrained POPs

The Broyden banded function is defined as

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

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

The task is to minimize the Broyden banded function over $\mathbb{R}^n$ which is formulated as an unconstrained POP. Using the relaxation order $d = 3$, we solve the CSSOS hierarchy ($\mathrm{Q}_d^{\mathrm{cs}}$) (2.6), the TSSOS hierarchy ($\mathrm{Q}_{d,k}^{\mathrm{ts}}$) (2.10) with $k = 1$ and the CS-TSSOS hierarchy ($\mathrm{Q}_{d,k}^{\mathrm{cs\text{-}ts}}$) (3.5) with $k = 1$. In the latter two cases, approximately smallest chordal extensions are used. We also solve the POP with the SDSOS approach. The results are displayed in Table 2.

It can be seen from the table that CS-TSSOS significantly reduces the maximal size of SDP blocks and is the most efficient approach. CSSOS, TSSOS, and CS-TSSOS all give the exact minimum 0 while SDSOS only gives a very loose lower bound -13731 when $n = 20$. Due to the limitation of memory, CSSOS scales up to 180 variables; TSSOS scales up to 40 variables; SDSOS scales up to 20 variables. On the other hand, CS-TSSOS can easily handle instances with up to 500 variables.

5.2 Benchmarks for Constrained POPs

- The generalized Rosenbrock function

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

- The Broyden tridiagonal function

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

Table 2. The Result for Broyden Banded Functions ($d = 3$)

var	CSSOS			TSSOS			CS-TSSOS			SDSOS	
	mb	opt	time	mb	opt	time	mb	opt	time	opt	time
20	120	0	21.7	33	0	4.39	19	0	2.24	−13731	374
40	120	0	44.6	52	0	231	19	0	6.95	-	-
60	120	0	81.8	-	-	-	19	0	13.0	-	-
80	120	0	116	-	-	-	19	0	19.6	-	-
100	120	0	151	-	-	-	19	0	27.0	-	-
120	120	0	195	-	-	-	19	0	34.4	-	-
140	120	0	249	-	-	-	19	0	43.1	-	-
160	120	0	298	-	-	-	19	0	50.2	-	-
180	120	0	338	-	-	-	19	0	63.8	-	-
200	120	-	-	-	-	-	19	0	72.9	-	-
250	120	-	-	-	-	-	19	0	106	-	-
300	120	-	-	-	-	-	19	0	132	-	-
400	120	-	-	-	-	-	19	0	220	-	-
500	120	-	-	-	-	-	19	0	313	-	-

- The chained Wood function

$$\begin{aligned} f_{\mathrm{cW}}(\mathbf{x}) = 1 + \sum_{i\in J}&(100(x_{i+1}-x_i^2)^2 + (1-x_i)^2 + 90(x_{i+3}-x_{i+2}^2)^2 \\ &+ (1-x_{i+2})^2 + 10(x_{i+1}+x_{i+3}-2)^2 + 0.1(x_{i+1}-x_{i+3})^2), \end{aligned}$$

where $J = \{1, 3, 5, \ldots, n-3\}$ and $4|n$.

With the generalized Rosenbrock (respectively, Broyden tridiagonal or chained Wood) function as the objective function, we consider the following constrained POP:

$$\begin{cases} \inf & f_{\mathrm{gR}} \quad (\text{resp. } f_{\mathrm{Bt}} \text{ or } f_{\mathrm{cW}}) \\ \text{s.t.} & 1 - (\sum_{i=20j-19}^{20j} x_i^2) \geq 0, \quad j = 1, 2, \ldots, n/20, \end{cases} \tag{5.1}$$

where $20|n$. The generalized Rosenbrock function, the Broyden tridiagonal function and the chained Wood function involve cliques with 2 or 3 variables, which can be efficiently handled by the CSSOS hierarchy; see [40]. For them, the CS-TSSOS hierarchy gives almost the same results with the CSSOS hierarchy. Hence, we add the sphere constraints in (5.1) to increase the clique size and to show the difference.

For these problems, the minimum relaxation order $d = 2$ is used. As in the unconstrained case, we solve the CSSOS hierarchy $(\mathrm{Q}_d^{\mathrm{cs}})$ (2.6), the TSSOS hierarchy $(\mathrm{Q}_{d,k}^{\mathrm{ts}})$ (2.10) with $k = 1$ and the CS-TSSOS hierarchy $(\mathrm{Q}_{d,k}^{\mathrm{cs\text{-}ts}})$ (3.5) with $k = 1$, and use approximately smallest chordal extensions. We also solve these POPs with the SDSOS approach. The results are displayed in Tables 3–5.

From these tables, one can see that CS-TSSOS significantly reduces the maximal size of SDP blocks and is again the most efficient approach. For the generalized Rosenbrock function, CSSOS, TSSOS, and CS-TSSOS give almost the same optimum while SDSOS gives a slightly loose lower bound (only for $n = 40$); for the Broyden tridiagonal function, CSSOS, TSSOS, and CS-TSSOS all give the same optimum while SDSOS gives a very loose lower bound (only for $n = 40$); for the chained Wood function, CSSOS, TSSOS, and CS-TSSOS all give the same optimum while SDSOS gives a slightly loose lower bound (only for $n = 40$). Due to the limitation of memory, CSSOS scales

Table 3. The Result for the Generalized Rosenbrock Function ($d = 2$)

var	CSSOS			TSSOS			CS-TSSOS			SDSOS	
	mb	opt	time	mb	opt	time	mb	opt	time	opt	time
40	231	38.051	126	41	38.049	0.61	21	38.049	0.23	37.625	115
60	231	57.849	232	61	57.845	3.31	21	57.845	0.32	-	-
80	231	77.647	306	81	77.641	11.7	21	77.641	0.41	-	-
100	231	97.445	377	101	97.436	31.3	21	97.436	0.54	-	-
120	231	117.24	408	121	117.23	75.4	21	117.23	0.60	-	-
140	231	137.04	495	141	137.03	190	21	137.03	0.75	-	-
160	231	156.84	570	161	156.82	367	21	156.82	0.90	-	-
180	231	176.64	730	181	176.62	628	21	176.62	1.09	-	-
200	231	-	-	201	196.41	1327	21	196.41	1.27	-	-
300	231	-	-	-	-	-	21	295.39	2.26	-	-
400	231	-	-	-	-	-	21	394.37	3.36	-	-
500	231	-	-	-	-	-	21	493.35	4.65	-	-
1000	231	-	-	-	-	-	21	988.24	15.8	-	-

Table 4. The Result for the Broyden Tridiagonal Function ($d = 2$)

var	CSSOS			TSSOS			CS-TSSOS			SDSOS	
	mb	opt	time	mb	opt	time	mb	opt	time	opt	time
40	231	31.234	168	43	31.234	1.95	23	31.234	0.64	−5.8110	138
60	231	47.434	273	63	47.434	8.33	23	47.434	1.14	-	-
80	231	63.634	413	83	63.634	33.9	23	63.634	1.50	-	-
100	231	79.834	519	103	79.834	104	23	79.834	1.96	-	-
120	231	96.034	671	123	96.034	199	23	96.034	2.30	-	-
140	231	112.23	872	143	112.23	490	23	112.23	2.94	-	-
160	231	128.43	1002	163	128.43	783	23	128.43	3.67	-	-
180	231	144.63	1066	183	144.63	1329	23	144.63	4.46	-	-
200	231	-	-	-	-	-	23	160.83	4.88	-	-
300	231	-	-	-	-	-	23	241.83	8.67	-	-
400	231	-	-	-	-	-	23	322.83	13.3	-	-
500	231	-	-	-	-	-	23	403.83	19.9	-	-
1000	231	-	-	-	-	-	23	808.83	57.5	-	-

up to 180 variables; TSSOS scales up to 180 or 200 variables; SDSOS scales up to 40 variables. On the other hand, CS-TSSOS can easily handle these instances with up to 1000 variables.

5.3 The Max-Cut Problem

The Max-Cut problem is one of the basic combinatorial optimization problems, which is known to be NP-hard. Let $G(V, E)$ be an undirected graph with $V = \{1, \ldots, n\}$ and with edge weights w_{ij} for $\{i, j\} \in E$. Then, the Max-Cut problem for G can be formulated as a QCQP in binary variables:

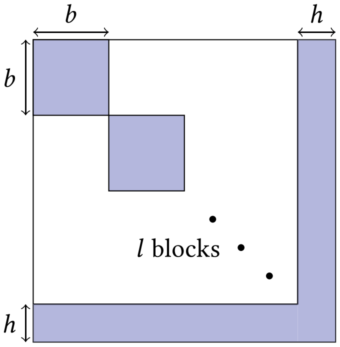
$$\begin{cases} \max & \frac{1}{2} \sum_{\{i,j\} \in E} w_{ij}(1 - x_i x_j) \\ \text{s.t.} & 1 - x_i^2 = 0, \quad i = 1, \ldots, n. \end{cases} \tag{5.2}$$

The property of binary variables in (5.2) can be also exploited to reduce the size of SDPs arising from the moment-SOS hierarchy, which has been implemented in TSSOS.

For the numerical experiments, we construct random instances of Max-Cut problems with a block-band sparsity pattern (illustrated in Figure 8) which consists of l blocks of size b and two

Table 5. The Result for the Chained Wood Function ($d = 2$)

var	CSSOS			TSSOS			CS-TSSOS			SDSOS	
	mb	opt	time	mb	opt	time	mb	opt	time	opt	time
40	231	574.51	164	41	574.51	0.81	21	574.51	0.26	518.11	110
60	231	878.26	254	61	878.26	3.61	21	878.26	0.40	-	-
80	231	1182.0	393	81	1182.0	15.3	21	1182.0	0.57	-	-
100	231	1485.8	505	101	1485.8	43.2	21	1485.8	0.73	-	-
120	231	1789.5	516	121	1789.5	88.4	21	1789.5	0.93	-	-
140	231	2093.3	606	141	2093.3	195	21	2093.3	1.16	-	-
160	231	2397.0	700	161	2397.0	403	21	2397.0	1.39	-	-
180	231	2700.8	797	181	2700.8	867	21	2700.8	1.54	-	-
200	231	-	-	201	3004.5	1238	21	3004.5	1.91	-	-
300	231	-	-	-	-	-	21	4523.6	3.39	-	-
400	231	-	-	-	-	-	21	6042.0	5.72	-	-
500	231	-	-	-	-	-	21	7560.7	7.88	-	-
1000	231	-	-	-	-	-	21	15155	23.0	-	-



l: the number of blocks; b: the size of blocks; h: the width of bands.

Fig. 8. The block-band sparsity pattern.

bands of width h. Here, we select $b = 25$ and $h = 5$. For a given l, we generate a random sparse binary matrix $A \in \mathbf{S}^{lb+h}$ according to the block-arrow sparsity pattern: the entries out of the blue area take zero; the entries in the block area take one with probability 0.16; the entries in the band area take one with probability $2/\sqrt{l}$. Then we construct the graph G with A as its adjacency matrix. For each edge $\{i,j\} \in E(G)$, the weight w_{ij} randomly takes values 1 or -1 with equal probability. Doing so, we build 10 Max-Cut instances with $l = 20, 40, 60, 80, 100, 120, 140, 160, 180, 200$, respectively.[5] The largest number of nodes is 5005.

For each instance, we solve the first-order moment-SOS relaxation (Shor's relaxation), the CSSOS hierarchy with $d = 2$, and the CS-TSSOS hierarchy with $d = 2, k = 1$ for which the maximal chordal extension is used. The results are displayed in Table 6. From the table, we can see that for each instance, both CSSOS and CS-TSSOS significantly improve the bound obtained by Shor's relaxation. Meanwhile, CS-TSSOS is several times faster than CSSOS at the cost of possibly providing a sightly weaker bound. In addition, CS-TSSOS yields smaller block sizes than CSSOS.

[5] The instances are available at https://wangjie212.github.io/jiewang/code.html.

Table 6. The Result for Max-Cut Instances

instance	nodes	edges	mc	Shor	CSSOS			CS-TSSOS		
				opt	mb	opt	time	mb	opt	time
g20	505	2045	14	570	120	488	51.2	92	488	19.6
g40	1005	3441	14	1032	120	885	134	92	893	41.1
g60	1505	4874	14	1439	120	1227	183	92	1247	71.3
g80	2005	6035	15	1899	136	1638	167	106	1669	84.8
g100	2505	7320	14	2398	120	2073	262	92	2128	112
g120	3005	8431	14	2731	120	2358	221	79	2443	127
g140	3505	9658	13	3115	105	2701	250	79	2812	153
g160	4005	10677	14	3670	120	3202	294	79	3404	166
g180	4505	12081	13	4054	105	3525	354	79	3666	246
g200	5005	13240	13	4584	105	4003	374	79	4218	262

In this table, only the integer part of optimal values is preserved.

5.4 The AC-OPF Problem

The AC optimal power flow (AC-OPF) is a central problem in power systems. It can be formulated as the following POP in complex variables V_i, S_q^g, S_{ij}:

$$\left\{\begin{array}{ll}\inf\limits_{V_i,S_q^g,S_{ij}} & \sum_{q\in G}(\mathbf{c}_{2q}(\Re(S_q^g))^2+\mathbf{c}_{1q}\Re(S_q^g)+\mathbf{c}_{0q})\\ \text{s.t.} & \angle V_r=0,\\ & \mathbf{S}_q^{gl}\le S_q^g\le \mathbf{S}_q^{gu},\quad \forall q\in G,\\ & \boldsymbol{v}_i^l\le |V_i|\le \boldsymbol{v}_i^u,\quad \forall i\in N,\\ & \sum_{q\in G_i}S_q^g-\mathbf{S}_i^d-\mathbf{Y}_i^s|V_i|^2=\sum_{(i,j)\in E_i\cup E_i^R}S_{ij},\quad \forall i\in N,\\ & S_{ij}=(\mathbf{Y}_{ij}^*-\mathbf{i}\frac{\mathbf{b}_{ij}^c}{2})\frac{|V_i|^2}{|\mathbf{T}_{ij}|^2}-\mathbf{Y}_{ij}^*\frac{V_iV_j^*}{\mathbf{T}_{ij}},\quad \forall (i,j)\in E,\\ & S_{ji}=(\mathbf{Y}_{ij}^*-\mathbf{i}\frac{\mathbf{b}_{ij}^c}{2})|V_j|^2-\mathbf{Y}_{ij}^*\frac{V_i^*V_j}{\mathbf{T}_{ij}^*},\quad \forall (i,j)\in E,\\ & |S_{ij}|\le \mathbf{s}_{ij}^u,\quad \forall (i,j)\in E\cup E^R,\\ & \boldsymbol{\theta}_{ij}^{\Delta l}\le \angle(V_iV_j^*)\le \boldsymbol{\theta}_{ij}^{\Delta u},\quad \forall (i,j)\in E.\end{array}\right. \tag{5.3}$$

The meaning of the symbols in (5.3) is as follows: N - the set of buses, G - the set of generators, G_i - the set of generators connected to bus i, E - the set of *from* branches, E^R - the set of *to* branches, E_i and E_i^R - the subsets of branches that are incident to bus i, $\mathbf{i}$ - imaginary unit, V_i - the voltage at bus i, S_q^g - the power generation at generator q, S_{ij} - the power flow from bus i to bus j, $\Re(\cdot)$ - real part of a complex number, $\angle(\cdot)$ - angle of a complex number, $|\cdot|$ - magnitude of a complex number, $(\cdot)^*$ - conjugate of a complex number, r - the voltage angle reference bus. All symbols in boldface are constants ($\mathbf{c}_{0q},\mathbf{c}_{1q},\mathbf{c}_{2q},\boldsymbol{v}_i^l,\boldsymbol{v}_i^u,\mathbf{s}_{ij}^u,\boldsymbol{\theta}_{ij}^{\Delta l},\boldsymbol{\theta}_{ij}^{\Delta u}\in\mathbb{R},\mathbf{S}_q^{gl},\mathbf{S}_q^{gu},\mathbf{S}_i^d,\mathbf{Y}_i^s,\mathbf{Y}_{ij},\mathbf{b}_{ij}^c,\mathbf{T}_{ij}\in\mathbb{C}$). For a full description on the AC-OPF problem, the reader may refer to [6]. By introducing real variables for both real and imaginary parts of each complex variable, we can convert the AC-OPF problem to a POP involving only real variables.[6]

To tackle an AC-OPF instance, we first compute a locally optimal solution with a local solver and then rely on an SDP relaxation to certify the global optimality. Suppose that the optimal value

[6] The expressions involving angles of complex variables can be converted to polynomials by using $\tan(\angle z)=y/x$ for $z=x+\mathbf{i}y\in\mathbb{C}$.

Table 7. The Data for AC-OPF Instances

case name	var	cons	mc	AC	Shor	
					opt	gap
3_lmbd_api	12	28	6	1.1242e4	1.0417e4	7.34%
5_pjm	20	55	6	1.7552e4	1.6634e4	5.22%
24_ieee_rts_api	114	315	10	1.3495e5	1.3216e5	2.06%
24_ieee_rts_sad	114	315	14	7.6943e4	7.3592e4	4.36%
30_as_api	72	297	8	4.9962e3	4.9256e3	1.41%
73_ieee_rts_api	344	971	16	4.2263e5	4.1041e5	2.89%
73_ieee_rts_sad	344	971	16	2.2775e5	2.2148e5	2.75%
162_ieee_dtc	348	1809	21	1.0808e5	1.0616e5	1.78%
162_ieee_dtc_api	348	1809	21	1.2100e5	1.1928e5	1.42%
240_pserc	766	3322	16	3.3297e6	3.2818e6	1.44%
500_tamu_api	1112	4613	20	4.2776e4	4.2286e4	1.14%
500_tamu	1112	4613	30	7.2578e4	7.1034e4	2.12%
793_goc	1780	7019	18	2.6020e5	2.5636e5	1.47%
1888_rte	4356	18257	26	1.4025e6	1.3748e6	1.97%
3022_goc	6698	29283	50	6.0143e5	5.9278e5	1.44%

reported by the local solver is AC and the optimal value of the SDP relaxation is opt. The *optimality gap* between the locally optimal solution and the SDP relaxation is defined by

$$\text{gap} := \frac{\text{AC} - \text{opt}}{\text{AC}} \times 100\%.$$

If the optimality gap is less than 1.00%, then we accept the locally optimal solution as globally optimal. For many AC-OPF instances, the first-order moment-SOS relaxation (Shor's relaxation) is already able to certify the global optimality (with an optimality gap less than 1.00%). Therefore, we focus on the more challenging AC-OPF instances for which the optimality gap given by Shor's relaxation is greater than 1.00%. We select such instances from the AC-OPF library *PGLiB* [6]. Since we shall go to the second-order moment-SOS relaxation, we can replace the variables S_{ij} and S_{ji} by their right-hand side expressions in (5.3) and then convert the resulting problem to a POP involving real variables. The data for these selected AC-OPF instances are displayed in Table 7, where the AC values are taken from *PGLiB*.

We solve Shor's relaxation, the CSSOS hierarchy with $d = 2$ and the CS-TSSOS hierarchy with $d = 2, k = 1$ for these AC-OPF instances and the results are displayed in Tables 7 and 8. For instances 162_ieee_dtc, 162_ieee_dtc_api, 500_tamu, 1888_rte, with maximal chordal extensions Mosek ran out of memory and so we use approximately smallest chordal extensions. As the tables show, CS-TSSOS is more efficient and scales much better with the problem size than CSSOS. In particular, CS-TSSOS succeeds in reducing the optimality gap to less than 1.00% for all instances.

6 DISCUSSION AND CONCLUSIONS

This article introduces the CS-TSSOS hierarchy, a sparse variant of the moment-SOS hierarchy, which can be used to solve large-scale real-world nonlinear optimization problems, assuming that the input data are sparse polynomials. In addition to its theoretical convergence guarantees, CS-TSSOS allows one to make a trade-off between the quality of optimal values and the computational efficiency by controlling the types of chordal extensions and the sparse order k.

Table 8. The Result for AC-OPF Instances

case name	CSSOS				CS-TSSOS				
	mb	opt	time	gap	mb	opt	time	gap	CE
3_lmbd_api	28	1.1242e4	0.21	0.00%	22	1.1242e4	0.09	0.00%	max
5_pjm	28	1.7543e4	0.56	0.05%	22	1.7543e4	0.30	0.05%	max
24_ieee_rts_api	66	1.3442e5	5.59	0.39%	31	1.3396e5	2.01	0.73%	max
24_ieee_rts_sad	120	7.6943e4	94.9	0.00%	39	7.6942e4	14.8	0.00%	max
30_as_api	45	4.9927e3	4.43	0.07%	22	4.9920e3	2.69	0.08%	max
73_ieee_rts_api	153	4.2246e5	758	0.04%	44	4.2072e5	96.0	0.45%	max
73_ieee_rts_sad	153	2.2775e5	504	0.00%	44	2.2766e5	71.5	0.04%	max
162_ieee_dtc	253	–	–	–	34	1.0802e5	278	0.05%	min
162_ieee_dtc_api	253	–	–	–	34	1.2096e5	201	0.03%	min
240_pserc	153	3.3072e6	585	0.68%	44	3.3042e6	33.9	0.77%	max
500_tamu_api	231	4.2413e4	3114	0.85%	39	4.2408e4	46.6	0.86%	max
500_tamu	496	–	–	–	31	7.2396e4	410	0.25%	min
793_goc	190	2.5938e5	563	0.31%	33	2.5932e5	66.1	0.34%	max
1888_rte	378	–	–	–	27	1.3953e6	934	0.51%	min
3022_goc	1326	–	–	–	76	5.9858e5	1886	0.47%	max

By fully exploiting sparsity, CS-TSSOS allows one to go beyond Shor's relaxation and solve the second-order moment-SOS relaxation associated with large-scale POPs to obtain more accurate bounds. Indeed CS-TSSOS can handle second-order relaxations of POP instances with thousands of variables and constraints on a standard laptop in tens of minutes. Such instances include the optimal power flow (OPF) problem, an important challenge in the management of electricity networks. In particular, our plan is to perform advanced numerical experiments on HPC cluster, for OPF instances with larger numbers of buses [10].

This work suggests additional investigation tracks for further research:

(1) The standard procedure of extracting optimal solutions for the dense moment-SOS hierarchy does not apply to the CS-TSSOS hierarchy. It would be interesting to develop a procedure for extracting (approximate) solutions from partial information of moment matrices.

(2) Recall that chordal extension plays an important role for both correlative and term sparsity patterns. It turns out that the size of the resulting maximal cliques is crucial for the overall computational efficiency of the CS-TSSOS hierarchy. So far, we have only considered *maximal* chordal extensions (for convergence guarantee) and approximately *smallest* chordal extensions. It would be worth investigating more general choices of chordal extensions.

(3) The CS-TSSOS strategy could be adapted to other applications involving sparse polynomial problems, including deep learning [9] or noncommutative optimization problems [18] arising in quantum information.

(4) Last but not least, a challenging research issue is to establish serious computationally cheaper alternatives to interior-point methods for solving SDP relaxations of POPs. The recent work [49] which reports spectacular results for standard SDPs (and Max-Cut problems in particular) is a positive sign in this direction.

ACKNOWLEDGMENTS

We would like to thank Tillmann Weisser for helpful discussions on OPF problems.

REFERENCES

[1] Jim Agler, William Helton, Scott McCullough, and Leiba Rodman. 1988. Positive semidefinite matrices with a given sparsity pattern. *Linear Algebra and Its Applications* 107 (1988), 101–149.

[2] Amir A. Ahmadi and Anirudha Majumdar. 2019. DSOS and SDSOS optimization: More tractable alternatives to sum of squares and semidefinite optimization. *SIAM Journal on Applied Algebra and Geometry* 3, 2 (2019), 193–230.

[3] Erling D. Andersen and Knud D. Andersen. 2000. The MOSEK interior point optimizer for linear programming: An implementation of the homogeneous algorithm. In *High Performance Optimization*. Springer, 197–232.

[4] Mosek ApS. 2019. The MOSEK Optimization Software. https://www.mosek.com/.

[5] Gennadiy Averkov. 2019. Optimal size of linear matrix inequalities in semidefinite approaches to polynomial optimization. *SIAM Journal on Applied Algebra and Geometry* 3, 1 (2019), 128–151.

[6] Sogol Babaeinejadsarookolaee, Adam Birchfield, Richard D. Christie, Carleton Coffrin, Christopher DeMarco, Ruisheng Diao, Michael Ferris, Stephane Fliscounakis, Scott Greene, Renke Huang, Cedric Josz, Roman Korab, Bernard Lesieutre, Jean Maeght, Terrence W. K. Mak, Daniel K. Molzahn, Thomas J. Overbye, Patrick Panciatici, Byungkwon Park, Jonathan Snodgrass, Ahmad Tbaileh, Pascal Van Hentenryck, and Ray Zimmerman. 2019. The power grid library for benchmarking AC optimal power flow algorithms. *arXiv preprint arXiv:1908.02788* (2019).

[7] Jean R. S. Blair and Barry Peyton. 1993. An introduction to chordal graphs and clique trees. In *Graph Theory and Sparse Matrix Computation*. Springer, 1–29.

[8] Venkat Chandrasekaran and Parikshit Shah. 2016. Relative entropy relaxations for signomial optimization. *SIAM Journal on Optimization* 26, 2 (2016), 1147–1173.

[9] Tong Chen, Jean B. Lasserre, Victor Magron, and Edouard Pauwels. 2020. Semialgebraic optimization for Lipschitz constants of RELU networks. *Advances in Neural Information Processing Systems* 33 (2020), 19189–19200.

[10] Anders Eltved, Joachim Dahl, and Martin S. Andersen. 2019. On the robustness and scalability of semidefinite relaxation for optimal power flow problems. *Optimization and Engineering* 21 (2019), 375–392.

[11] Delbert Fulkerson and Oliver Gross. 1965. Incidence matrices and interval graphs. *Pacific J. Math.* 15, 3 (1965), 835–855.

[12] Martin Charles Golumbic. 2004. *Algorithmic Graph Theory and Perfect Graphs*. Elsevier.

[13] David Grimm, Tim Netzer, and Markus Schweighofer. 2007. A note on the representation of positive polynomials with structured sparsity. *Archiv Der Mathematik* 89, 5 (2007), 399–403.

[14] Robert Grone, Charles R. Johnson, Eduardo M. Sá, and Henry Wolkowicz. 1984. Positive definite completions of partial hermitian matrices. *Linear Algebra Appl.* 58 (1984), 109–124.

[15] Didier Henrion and Jean B. Lasserre. 2005. Detecting global optimality and extracting solutions in gloptipoly. In *Positive Polynomials in Control*. Springer, 293–310.

[16] Sadik Iliman and Timo De Wolff. 2016. Amoebas, nonnegative polynomials and sums of squares supported on circuits. *Research in the Mathematical Sciences* 3, 1 (2016), 9.

[17] Cédric Josz and Daniel K. Molzahn. 2018. Lasserre hierarchy for large scale polynomial optimization in real and complex variables. *SIAM Journal on Optimization* 28, 2 (2018), 1017–1048.

[18] Igor Klep, Victor Magron, and Janez Povh. 2021. Sparse noncommutative polynomial optimization. *Mathematical Programming* (2021), 1–41.

[19] Jean-Louis Krivine. 1964. Anneaux préordonnés. *Journal d'Analyse Mathématique* 12, 1 (1964), 307–326.

[20] Jean B. Lasserre. 2001. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization* 11, 3 (2001), 796–817.

[21] Jean B. Lasserre. 2006. Convergent SDP-relaxations in polynomial optimization with sparsity. *SIAM Journal on Optimization* 17, 3 (2006), 822–843.

[22] Jean B. Lasserre. 2015. *An Introduction to Polynomial and Semi-Algbraic Optimization*. Cambridge University Press, Cambridge, UK.

[23] Jean B. Lasserre, Kim-Chuan Toh, and Shouguang Yang. 2017. A bounded degree SOS hierarchy for polynomial optimization. *EURO Journal on Computational Optimization* 5, 1–2 (2017), 87–117.

[24] Monique Laurent. 2003. A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0–1 programming. *Mathematics of Operations Research* 28, 3 (2003), 470–496.

[25] Victor Magron. 2018. Interval enclosures of upper bounds of roundoff errors using semidefinite programming. *ACM Trans. Math. Software* 44, 4 (2018), 1–18.

[26] Victor Magron, George Constantinides, and Alastair Donaldson. 2017. Certified roundoff error bounds using semidefinite programming. *ACM Trans. Math. Software* 43, 4, Article 34 (2017), 34 pages.

[27] Victor Magron and Jie Wang. 2021. TSSOS: A Julia library to exploit sparsity for large-scale polynomial optimization. *arXiv preprint arXiv:2103.00915* (2021).

[28] Ngoc Hoang Anh Mai, Victor Magron, and Jean Lasserre. 2022. A sparse version of Reznick's positivstellensatz. *Mathematics of Operations Research* (2022).

[29] Alexandre Megretski. 2010. Systems Polynomial Optimization Tools (SPOT). https://github.com/spot-toolbox/spotless.

[30] Jared Miller, Yang Zheng, Mario Sznaier, and Antonis Papachristodoulou. 2022. Decomposed structured subsets for semidefinite and sum-of-squares optimization. *Automatica* 137 (2022), 110125. https://www.sciencedirect.com/science/article/abs/pii/S0005109821006543.

[31] Jiawang Nie. 2014. Optimality conditions and finite convergence of lasserre's hierarchy. *Mathematical Programming* 146, 1 (1 Aug. 2014), 97–121.

[32] Bruce Reznick. 1978. Extremal PSD forms with few terms. *Duke Mathematical Journal* 45, 2 (1978), 363–374.

[33] Cordian Riener, Thorsten Theobald, Lina Jansson Andrén, and Jean B. Lasserre. 2013. Exploiting symmetries in SDP-relaxations for polynomial optimization. *Mathematics of Operations Research* 38, 1 (2013), 122–141.

[34] Naum Z. Shor. 1987. Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences* 25 (1987), 1–11.

[35] Gilbert Stengle. 1974. A nullstellensatz and a positivstellensatz in semialgebraic geometry. *Math. Ann.* 207, 2 (1974), 87–97.

[36] Matteo Tacchi, Carmen Cardozo, Didier Henrion, and Jean B. Lasserre. 2020. Approximating regions of attraction of a sparse polynomial differential system. *IFAC-PapersOnLine* 53, 2 (2020), 3266–3271.

[37] Matteo Tacchi, Tillmann Weisser, Jean B. Lasserre, and Didier Henrion. 2021. Exploiting sparsity for semi-algebraic set volume computation. *Foundations of Computational Mathematics* (2021), 1–49.

[38] Kim C. Toh. 2018. Some numerical issues in the development of SDP algorithms. *Informs OS Today* 8, 2 (2018), 7–20.

[39] Lieven Vandenberghe and Martin S. Andersen. 2015. Chordal graphs and semidefinite optimization. *Foundations and Trends in Optimization* 1, 4 (2015), 241–433.

[40] Hayato Waki, Sunyoung Kim, Masakazu Kojima, and Masakazu Muramatsu. 2006. Sums of squares and semidefinite programming relaxations for polynomial optimization problems with structured sparsity. *SIAM Journal on Optimization* 17, 1 (2006), 218–242.

[41] Hayato Waki, Sunyoung Kim, Masakazu Kojima, Masakazu Muramatsu, and Hiroshi Sugimoto. 2008. Algorithm 883: SparsePOP—a sparse semidefinite programming relaxation of polynomial optimization problems. *ACM Transactions on Mathematical Software (TOMS)* 35, 2 (2008), 1–13.

[42] Jie Wang, Haokun Li, and Bican Xia. 2019. A new sparse SOS decomposition algorithm based on term sparsity. In *Proceedings of the 2019 on International Symposium on Symbolic and Algebraic Computation.* 347–354.

[43] Jie Wang and Victor Magron. 2020. A second order cone characterization for sums of nonnegative circuits. In *Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation.* 450–457.

[44] Jie Wang and Victor Magron. 2021. Exploiting sparsity in complex polynomial optimization. *Journal of Optimization Theory and Applications* (2021), 1–25.

[45] Jie Wang and Victor Magron. 2021. Exploiting term sparsity in noncommutative polynomial optimization. *Computational Optimization and Applications* 80, 2 (2021), 483–521.

[46] Jie Wang, Victor Magron, and Jean B. Lasserre. 2021. Chordal-TSSOS: A moment-SOS hierarchy that exploits term sparsity with chordal extension. *SIAM Journal on Optimization* 31, 1 (2021), 114–141.

[47] Jie Wang, Victor Magron, and Jean B. Lasserre. 2021. TSSOS: A moment-SOS hierarchy that exploits term sparsity. *SIAM Journal on Optimization* 31, 1 (2021), 30–58.

[48] Henry Wolkowicz, Romesh Saigal, and Lieven Vandenberghe. 2012. *Handbook of Semidefinite Programming: Theory, Algorithms, and Applications.* Vol. 27. Springer Science & Business Media.

[49] Alp Yurtsever, Joel A. Tropp, Olivier Fercoq, Madeleine Udell, and Volkan Cevher. 2021. Scalable semidefinite programming. *SIAM Journal on Mathematics of Data Science* 3, 1 (2021), 171–200.

Received 6 May 2020; revised 19 April 2022; accepted 23 October 2022