Certifying Global Optimality of AC-OPF Solutions via the CS-TSSOS Hierarchy

Jie Wang[a,*], Victor Magron[b], Jean B. Lasserre[b]

[a] *Academy of Mathematics and Systems Science, CAS, Beijing, China*
[b] *Laboratory for Analysis and Architecture of Systems, CNRS, Toulouse, France*

Abstract

We report the experimental results on certifying 1% global optimality of solutions of AC-OPF instances from PGLiB via the CS-TSSOS hierarchy – a moment-SOS based hierarchy that exploits both correlative and term sparsity, which can provide tighter SDP relaxations than Shor's relaxation. Our numerical experiments demonstrate that the CS-TSSOS hierarchy scales well with the problem size and is indeed useful in certifying global optimality of solutions for large-scale real world problems, e.g., the AC-OPF problem. In particular, we are able to certify 1% global optimality for a challenging AC-OPF instance with 6515 buses involving 14398 real variables and 63577 constraints.

Keywords: sparse moment-SOS hierarchy, CS-TSSOS hierarchy, global optimality, Lasserre's hierarchy, large-scale polynomial optimization, optimal power flow
2020 MSC: 90C23, 14P10, 90C22, 90C26, 12D15

1. Introduction

Background on polynomial optimization. Polynomial optimization considers optimization problems where both the cost function and constraints are defined by polynomials, which widely arises in numerous fields, such as optimal power flow [1], numerical analysis [2], computer vision [3], deep learning [4], discrete optimization [5], etc. Even though it is usually not difficult

*Corresponding author.
Email address: wangjie212@amss.ac.cn (Jie Wang)
URL: https://wangjie212.github.io/jiewang (Jie Wang)

to find a locally optimal solution via a local solver (e.g., Ipopt [6]), the task of solving a polynomial optimization problem (POP) to global optimality is NP-hard in general. Over the last decades, the moment-sums of squares (moment-SOS) hierarchy consisting of a sequence of increasingly tight SDP relaxations, initially established by Lasserre [7], has become a popular tool to handle polynomial optimization. The moment-SOS hierarchy features its global convergence and finite convergence under mild conditions [8]. However, the main concern on the moment-SOS hierarchy comes from its scalability as the d-th step of the moment-SOS hierarchy involves a semidefinite program (SDP) of size $\binom{n+d}{d}$ where n is the number of decision variables of the POP. Except the first (relaxation) step of the moment-SOS hierarchy (also known as Shor's relaxation for quadratically constrained quadratic programs (QCQP) [9]), solving higher steps of the moment-SOS hierarchy is typically limited to small-scale POPs, at least when relying on interior-point solvers. To overcome this scalability issue, one practicable way is to exploit the structure of the POP to reduce the size of SDPs arising from the moment-SOS hierarchy. Such structures include symmetry [10], correlative sparsity [11, 12], term sparsity [13, 14, 15]. The purpose of this paper is to demonstrate that the scalability of the moment-SOS hierarchy can be significantly improved when appropriate sparsity patterns are accessible via a thorough numerical experiment on the AC optimal power flow (AC-OPF) problem.

Background on the AC-OPF problem. The AC-OPF is a fundamental problem in power systems, which has been extensively studied in recent years; for a detailed introduction and recent developments, the reader is referred to the survey [16] and references therein. One can formulate the AC-OPF problem as a POP either with real variables [16, 1] or with complex variables [17]. Nonlinear programming tools can mostly produce a locally optimal solution whose global optimality is however unknown. Since 2006, several convex relaxation schemes (e.g., second order cone relaxations (SOCR) [18], quadratic convex relaxations (QCR) [19] and semidefinite (Shor's) relaxations (SDR) [20]) have been proposed to provide lower bounds for the AC-OPF which can be then used to certify global optimality of locally optimal solutions. While these relaxations (SOCR, QCR, SDR) could be scalable to problems of large size and prove to be tight for quite a few cases [21, 19, 22], they yield significant optimality gaps for a large number of other cases[1]. To tackle these more challenging cases, it is then mandatory to go to higher steps of the moment-SOS hierarchy which can provide tighter lower bounds. Along with this line recently in [23], the authors certified global optimality for all AC-OPF instances with up to 300 buses from the AC-OPF library PGLiB using the moment-SOS hierarchy (combined with other techniques to improve scalability). Exact global optimality is obtained for certain 2000 bus cases in [17] using a multi-order moment-SOS hierarchy (after some case modifications and dropping some constraints). To tackle AC-OPF instances with more buses, the CS-TSSOS hierarchy then comes into play.

The CS-TSSOS hierarchy for large-scale POPs. The CS-TSSOS hierarchy [15] is a sparsity-adapted version of the moment-SOS hierarchy targeted at large-scale POPs by simultaneously exploiting correlative sparsity (CS) and term sparsity (TS). The underlying idea is the following:
(1) partitioning the system into subsystems by exploiting correlative sparsity, i.e., the fact that only a few variable products occur;
(2) exploiting term sparsity, i.e., the fact that the input data only contain a few terms (by comparison with the maximal possible amount), to each subsystem to further reduce the size of SDPs.
By virtue of this two-step reduction procedure, one may obtain SDP relaxations of significantly smaller size compared to the original SDP relaxations. Next the main concern on the CS-TSSOS hierarchy might be how it performs when applying to real-word large-scale POPs in terms of scalability and accuracy.

Certifying global optimality for AC-OPF instances from PGLiB. As the main contribution of this paper, we benchmark the CS-TSSOS hierarchy through a comprehensive numerical experiment on AC-OPF instances from the AC-OPF library PGLiB v20.07 [21] with up to tens of thousands of variables and constraints. The experimental results (see Section 6) demonstrate that the CS-TSSOS hierarchy scales well with the problem size and is able to certify global optimality (in the sense of reducing optimality gap

[1]The reader may find related results on benchmarking SOCR and QR at https://github.com/power-grid-lib/pglib-opf/blob/master/BASELINE.md.

within 1%) for many of the challenging test cases. In particular, the largest instance whose global optimality is certified beyond Shor's relaxation involves 14398 real variables and 63577 constraints. Besides, the largest instance for which the CS-TSSOS hierarchy is able to provide a smaller optimality gap than Shor's relaxation involves 24032 real variables and 96805 constraints. To the best of our knowledge, this is the first time in literature that one can solve higher steps of the moment-SOS hierarchy other than Shor's relaxation for POPs of such large sizes.

2. Notation and preliminaries

Let $\mathbf{x} = (x_1, \ldots, x_n)$ be a tuple of variables and $\mathbb{R}[\mathbf{x}] = \mathbb{R}[x_1, \ldots, x_n]$ be the ring of real n-variate polynomials. 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 $\mathscr{A} \subseteq \mathbb{N}^n$, $f_{\boldsymbol{\alpha}} \in \mathbb{R}$, and $\mathbf{x}^{\boldsymbol{\alpha}} := x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. The support of f is defined by $\text{supp}(f) := \{\boldsymbol{\alpha} \in \mathscr{A} \mid f_{\boldsymbol{\alpha}} \neq 0\}$. A positive semidefinite (PSD) matrix A is written as $A \succeq 0$. For a positive integer r, the set of $r \times r$ symmetric matrices is denoted by $\mathbf{S}^r$ and the set of $r \times r$ PSD matrices is denoted by $\mathbf{S}^r_+$. For matrices $A, B \in \mathbf{S}^r$, let $A \circ B \in \mathbf{S}^r$ denote the Hadamard product, defined by $[A \circ B]_{ij} = A_{ij}B_{ij}$. We use $|\cdot|$ to denote the cardinality of a set. For $d \in \mathbb{N}$, let $\mathbb{N}^n_d := \{\boldsymbol{\alpha} = (\alpha_i)_{i=1}^n \in \mathbb{N}^n \mid \sum_{i=1}^n \alpha_i \leq d\}$. The set $\mathbf{x}^{\mathbb{N}^n_d} := \{\mathbf{x}^{\boldsymbol{\alpha}} \mid \boldsymbol{\alpha} \in \mathbb{N}^n_d\}$ (fixing any ordering on it) is called the *standard monomial basis* (up to degree d). For convenience we abuse notation in the sequel, and denote by $\mathbb{N}^n_d$ instead of $\mathbf{x}^{\mathbb{N}^n_d}$ the standard monomial basis and use the exponent $\boldsymbol{\alpha}$ to represent a monomial $\mathbf{x}^{\boldsymbol{\alpha}}$. With $\mathbf{y} = (y_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathbb{N}^n} \subseteq \mathbb{R}$ being a sequence indexed by $\mathbb{N}^n$, 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 $\boldsymbol{\alpha} \in \mathbb{N}^n, \mathscr{A}, \mathscr{B} \subseteq \mathbb{N}^n$, 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 $m, l \in \mathbb{N}\backslash\{0\}$ with $l > m$, let $[m] := \{1, 2, \ldots, m\}$ and $[m : l] = \{m, m + 1, \ldots, l\}$. For $\boldsymbol{\beta} = (\beta_i)_i \in \mathbb{N}^n$, let $\text{supp}(\boldsymbol{\beta}) := \{i \in [n] \mid \beta_i \neq 0\}$.

An *(undirected) graph* $G(V, E)$ or simply G consists of a set of nodes V and a set of edges $E \subseteq \{\{u, v\} \mid u \neq v, (u, v) \in V \times V\}$. 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. 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. By definition, a *chordal graph* is a graph in

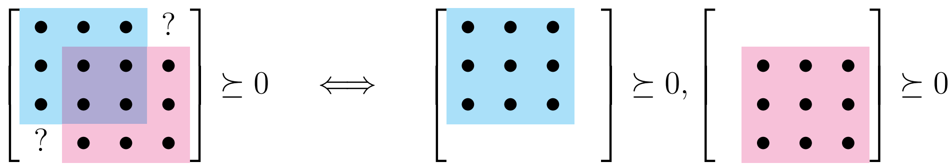

Figure 1: Illustration for Theorem 2.1

which any cycle of length at least four has a chord[2]. 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 $G(V,E)$. The chordal extension of G is usually not unique and the symbol G' is used to represent any specific chordal extension of G throughout the paper.

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 $Q_{uv} = Q_{vu} = 0$ whenever $u \neq v$ and $\{u,v\} \notin E$. Let $\mathbf{S}_G$ be the set of symmetric matrices with sparsity pattern G and let Π_G be the projection from $\mathbf{S}^{|V|}$ to the subspace $\mathbf{S}_G$, i.e., for $Q \in \mathbf{S}^{|V|}$,

$$[\Pi_G(Q)]_{uv} = \begin{cases} Q_{uv}, & \text{if } u = v \text{ or } \{u,v\} \in E, \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

The set $\Pi_G(\mathbf{S}_+^{|V|})$ denotes matrices in $\mathbf{S}_G$ that have a PSD completion in the sense that diagonal entries, and off-diagonal entries corresponding to edges of G are fixed; other off-diagonal entries are free. More precisely, $\Pi_G(\mathbf{S}_+^{|V|}) = \{\Pi_G(Q) \mid Q \in \mathbf{S}_+^{|V|}\}$. For a chordal graph G, the following theorem due to Grone et al. gives a characterization of matrices in the PSD completable cone $\Pi_G(\mathbf{S}_+^{|V|})$, which plays a crucial role in sparse semidefinite programming.

Theorem 2.1 ([24], Theorem 7). *Let $G(V,E)$ be a chordal graph and assume that $C_1,\dots,C_t$ 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 $i = 1,\dots,t$, where $Q[C_i]$ denotes the principal submatrix of Q indexed by the clique C_i.*

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

3. The CS-TSSOS hierarchy

The moment-SOS hierarchy [7] provides a sequence of increasingly tighter SDP relaxations for the following polynomial optimization problem:

$$
\text{(POP)}: \quad \begin{cases} \inf_{\mathbf{x}\in\mathbb{R}^n} & f(\mathbf{x}) \\ \text{s.t.} & g_j(\mathbf{x}) \geq 0, \quad j \in [m], \\ & g_j(\mathbf{x}) = 0, \quad j \in [m+1 : m+l], \end{cases} \tag{3.1}
$$

where $f, g_1, \ldots, g_{m+l} \in \mathbb{R}[\mathbf{x}]$ are all polynomials.

To state the moment hierarchy[3], recall that for a given $d \in \mathbb{N}$, the d-th order *moment matrix* $\mathbf{M}_d(\mathbf{y})$ associated with $\mathbf{y} = (y_{\boldsymbol{\alpha}})_{\boldsymbol{\alpha}\in\mathbb{N}^n}$ is defined by $[\mathbf{M}_d(\mathbf{y})]_{\boldsymbol{\beta}\boldsymbol{\gamma}} := L_{\mathbf{y}}(\mathbf{x}^{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\gamma}}) = y_{\boldsymbol{\beta}+\boldsymbol{\gamma}}, \forall \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{N}^n_d$ and the d-th order *localizing matrix* $\mathbf{M}_d(g\mathbf{y})$ associated with $\mathbf{y}$ and $g = \sum_{\boldsymbol{\alpha}} g_{\boldsymbol{\alpha}}\mathbf{x}^{\boldsymbol{\alpha}} \in \mathbb{R}[\mathbf{x}]$ is defined by $[\mathbf{M}_d(g\,\mathbf{y})]_{\boldsymbol{\beta}\boldsymbol{\gamma}} := L_{\mathbf{y}}(g\mathbf{x}^{\boldsymbol{\beta}}\mathbf{x}^{\boldsymbol{\gamma}}) = \sum_{\boldsymbol{\alpha}} g_{\boldsymbol{\alpha}} y_{\boldsymbol{\alpha}+\boldsymbol{\beta}+\boldsymbol{\gamma}}, \forall \boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{N}^n_d$. Let $d_j := \lceil \deg(g_j)/2 \rceil$ for $j = 1, \ldots, m+l$ and $d_{\min} := \max\{\lceil \deg(f)/2\rceil, d_1, \ldots, d_{m+l}\}$. Then for an integer $d \geq d_{\min}$, the d-th order moment relaxation for POP (3.1) is given by

$$
\begin{cases} \inf_{\mathbf{y}\in\mathbb{R}^{\binom{n+2d}{n}}} & L_{\mathbf{y}}(f) \\ \text{s.t.} & \mathbf{M}_d(\mathbf{y}) \succeq 0, \\ & \mathbf{M}_{d-d_j}(g_j\mathbf{y}) \succeq 0, \quad j \in [m], \\ & \mathbf{M}_{d-d_j}(g_j\mathbf{y}) = 0, \quad j \in [m+1 : m+l], \\ & y_{\mathbf{0}} = 1. \end{cases} \tag{3.2}
$$

We call (3.2) the *dense* moment hierarchy for POP (3.1), whose optima converge to the global optimum of (3.1) under mild conditions (slightly stronger than compactness of the feasible set) [7]. Unfortunately, when the relaxation order d is greater than 2, the dense moment hierarchy encounters a severe scalability issue as the maximal size of PSD constraints is a combinatorial number in terms of n and d. Therefore in the following subsections, we briefly revisit the framework of exploiting sparsity to derive a *sparse* moment hierarchy of remarkably smaller size for POP (3.1) in the presence of appropriate sparsity patterns. For details, the interested reader may refer to

[3] We mainly focus on the moment hierarchy. The SOS hierarchy consists of the dual SDPs.

the early work on correlative sparsity by Waki et al. [11, 12] and the recent work on term sparsity by the authors [25, 13, 14, 15].

3.1. Correlative sparsity (CS)

Let us from now on fix a relaxation order d. By exploiting correlative sparsity, we partition the set of variables into a tuple of subsets and then the initial system splits into a tuple of subsystems. To this end, we define the *correlative sparsity pattern (csp) graph*[4] associated with POP (3.1) to be the graph $G^{\rm csp}$ with nodes $V = [n]$ and edges E satisfying $\{i, j\} \in E$ if one of the following holds:

(i) there exists $\boldsymbol{\alpha} \in \operatorname{supp}(f) \cup \bigcup_{k \in J' \cup K'} \operatorname{supp}(g_k)$ such that $\{i, j\} \subseteq \operatorname{supp}(\boldsymbol{\alpha})$;

(ii) there exists $k \in [m+l] \backslash (J' \cup K')$ such that $\{i, j\} \subseteq \bigcup_{\boldsymbol{\alpha} \in \operatorname{supp}(g_k)} \operatorname{supp}(\boldsymbol{\alpha})$,

where $J' := \{k \in [m] \mid d_k = d\}$ and $K' := \{k \in [m+1 : m+l] \mid d_k = d\}$.

Let $(G^{\rm csp})'$ be a chordal extension of $G^{\rm csp}$ and $\{I_k\}_{k \in [p]}$ be the list of maximal cliques of $(G^{\rm csp})'$ with $n_k := |I_k|$. We then partition the polynomials $g_j, j \in [m] \backslash J'$ into groups $\{g_j \mid j \in J_k\}, k \in [p]$ which satisfy

(i) $J_1, \ldots, J_p \subseteq [m] \backslash J'$ are pairwise disjoint and $\cup_{k=1}^{p} J_k = [m] \backslash J'$;

(ii) for any $j \in J_k$, $\bigcup_{\boldsymbol{\alpha} \in \operatorname{supp}(g_j)} \operatorname{supp}(\boldsymbol{\alpha}) \subseteq I_k$, $k \in [p]$.

Similarly, we also partition the polynomials $g_j, j \in [m+1 : m+l] \backslash K'$ into groups $\{g_j \mid j \in K_k\}, k \in [p]$.

For any $k \in [p]$, let $\mathbf{M}_d(\mathbf{y}, I_k)$ (resp. $\mathbf{M}_d(g\mathbf{y}, I_k)$) be the moment (resp. localizing) submatrix obtained from $\mathbf{M}_d(\mathbf{y})$ (resp. $\mathbf{M}_d(g\mathbf{y})$) by retaining only those rows and columns indexed by $\boldsymbol{\beta} \in \mathbb{N}^n_d$ of $\mathbf{M}_d(\mathbf{y})$ (resp. $\mathbf{M}_d(g\mathbf{y})$) with $\operatorname{supp}(\boldsymbol{\beta}) \subseteq I_k$.

Example 3.1. *Consider the POP:*

$$\begin{cases} \inf_{\mathbf{x} \in \mathbb{R}^3} & x_1^2 x_2 + x_2 x_3^2 \\ \text{s.t.} & 1 - x_1^2 - x_2^2 \geq 0, \\ & 1 - x_2^2 - x_3^2 \geq 0, \\ & x_1^4 + x_2 x_3 = 1. \end{cases}$$

[4] We adopt the idea of “monomial sparsity” introduced in [17] for the definition of csp graphs, which thus is slightly different from the original definition given in [12].

Let us take the relaxation order $d = 2$. Then the csp graph is shown in Figure 2, which contains two maximal cliques: $\{x_1, x_2\}$ and $\{x_2, x_3\}$.

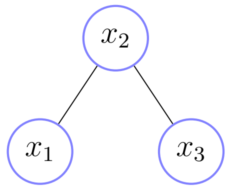

Figure 2: Illustration for correlative sparsity

3.2. Term sparsity (TS)

We next apply an iterative procedure to exploit term sparsity for each subsystem involving variables $\mathbf{x}(I_k) := \{x_i \mid i \in I_k\}$ for $k \in [p]$. The intuition behind this procedure is the following: starting with a minimal initial set of moments, we expand the set of moments that is taken into account in the moment relaxation by iteratively performing chordal extension to the related graphs inspired by Theorem 2.1. More concretely, let $\mathscr{A} := \operatorname{supp}(f) \cup \bigcup_{j=1}^{m+l} \operatorname{supp}(g_j)$ and $\mathscr{A}_k := \{\boldsymbol{\alpha} \in \mathscr{A} \mid \operatorname{supp}(\boldsymbol{\alpha}) \subseteq I_k\}$ for $k \in [p]$. We define $G^{(0)}_{d,k,0}$ to be the graph with nodes $V_{d,k,0} := \mathbb{N}^{n_k}_d$ and edges

$$E(G^{(0)}_{d,k,0}) := \{\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \mid \boldsymbol{\beta}, \boldsymbol{\gamma} \in V_{d,k,0}, \boldsymbol{\beta} + \boldsymbol{\gamma} \in \mathscr{A}_k \cup (2\mathbb{N})^n\}. \tag{3.3}$$

Note that here we embed $\mathbb{N}^{n_k}$ into $\mathbb{N}^n$ by specifying the i-th coordinate to be zero when $i \in [n] \backslash I_k$.

Example 3.2. *Consider again the POP in Example 3.1 with the relaxation order $d = 2$. Since there are two variable cliques derived from the correlative sparsity pattern, we have $p = 2$. Figure 3 illustrates the term sparsity pattern of this POP.*

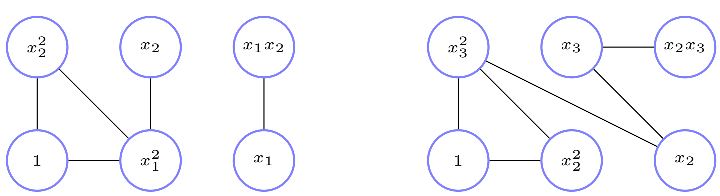

Figure 3: Illustration for term sparsity: $G^{(0)}_{d,1,0}$ (left) and $G^{(0)}_{d,2,0}$ (right)

For the sake of convenience, we set $g_0 := 1$ and $d_0 := 0$ hereafter and for a graph $G(V, E)$ with $V \subseteq \mathbb{N}^n$, let $\mathrm{supp}(G) := \{\boldsymbol{\beta} + \boldsymbol{\gamma} \mid \{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \in E\}$. Assume that $G^{(0)}_{d,k,j}, j \in J_k \cup K_k, k \in [p]$ are all empty graphs with $V_{d,k,j} := \mathbb{N}^{n_k}_{d-d_j}$. Now for each $j \in \{0\} \cup J_k \cup K_k, k \in [p]$, we iteratively define an ascending chain of graphs $(G^{(s)}_{d,k,j})_{s\geq 1}$ by

$$G^{(s)}_{d,k,j} := (F^{(s)}_{d,k,j})', \tag{3.4}$$

where $F^{(s)}_{d,k,j}$ is the graph with nodes $V_{d,k,j}$ and edges

$$E(F^{(s)}_{d,k,j}) = \{\{\boldsymbol{\beta}, \boldsymbol{\gamma}\} \mid \boldsymbol{\beta}, \boldsymbol{\gamma} \in V_{d,k,j}, (\boldsymbol{\beta} + \boldsymbol{\gamma} + \mathrm{supp}(g_j)) \cap \mathscr{C}^{(s-1)}_d \neq \emptyset\}, \tag{3.5}$$

with

$$\mathscr{C}^{(s-1)}_d := \bigcup_{k=1}^{p} \bigcup_{j\in\{0\}\cup J_k\cup K_k} (\mathrm{supp}(g_j) + \mathrm{supp}(G^{(s-1)}_{d,k,j}))). \tag{3.6}$$

Let $r_{d,k,j} := |\mathbb{N}^{n_k}_{d-d_j}| = \binom{n_k+d-d_j}{d-d_j}$ for all k, j. Then for each $s \geq 1$, the moment relaxation based on correlative-term sparsity for POP (3.1) is given by

$$\begin{cases} \inf_{\mathbf{y}} & L_{\mathbf{y}}(f) \\ \text{s.t.} & B_{G^{(s)}_{d,k,0}} \circ \mathbf{M}_d(\mathbf{y}, I_k) \in \Pi_{G^{(s)}_{d,k,0}}(\mathbf{S}^{r_{d,k,0}}_+), \quad k \in [p], \\ & B_{G^{(s)}_{d,k,j}} \circ \mathbf{M}_{d-d_j}(g_j\mathbf{y}, I_k) \in \Pi_{G^{(k)}_{d,k,j}}(\mathbf{S}^{r_{d,k,j}}_+), \quad j \in J_k, k \in [p], \\ & B_{G^{(s)}_{d,k,j}} \circ \mathbf{M}_{d-d_j}(g_j\mathbf{y}, I_k) = 0, \quad j \in K_k, k \in [p], \\ & L_{\mathbf{y}}(g_j) \geq 0, \quad j \in J', \\ & L_{\mathbf{y}}(g_j) = 0, \quad j \in K', \\ & y_{\mathbf{0}} = 1. \end{cases} \tag{3.7}$$

The above hierarchy is called the *CS-TSSOS* hierarchy, which is indexed by two parameters: the relaxation order d and the *sparse order* s.

3.3. The minimal initial relaxation step

For POP (3.1), suppose that f is not a homogeneous polynomial or the polynomials $g_j, j \in [m + l]$ are of different degrees as in the case of the AC-OPF problem. Then instead of using the uniform minimum relaxation order $d_{\min}$, it might be more beneficial, from the computational point of view, to assign different relaxation orders to different subsystems obtained from the correlative sparsity pattern for the initial relaxation step of the CS-TSSOS hierarchy. To this end, we redefine the csp graph $G^{\text{icsp}}(V,E)$ as follows: $V = [n]$ and $\{i,j\} \in E$ whenever there exists $\boldsymbol{\alpha} \in \text{supp}(f) \cup \bigcup_{j\in[m+l]} \text{supp}(g_j)$ such that $\{i,j\} \subseteq \text{supp}(\boldsymbol{\alpha})$. This is clearly a subgraph of G^{csp} defined in Section 3.1 and hence typically admits a smaller chordal extension. Let $(G^{\text{icsp}})'$ be a chordal extension of G^{icsp} and $\{I_k\}_{k\in[p]}$ be the list of maximal cliques of $(G^{\text{icsp}})'$ with $n_k := |I_k|$. Now we partition the polynomials $g_j, j \in [m]$ into groups $\{g_j \mid j \in J_k\}_{k\in[p]}$ and $\{g_j \mid j \in J'\}$ which satisfy

(i) $J_1, \ldots, J_p, J' \subseteq [m]$ are pairwise disjoint and $\bigcup_{k=1}^{p} J_k \cup J' = [m]$;

(ii) for any $j \in J_k$, $\bigcup_{\boldsymbol{\alpha}\in\text{supp}(g_j)} \text{supp}(\boldsymbol{\alpha}) \subseteq I_k$, $k \in [p]$;

(iii) for any $j \in J'$, $\bigcup_{\boldsymbol{\alpha}\in\text{supp}(g_j)} \text{supp}(\boldsymbol{\alpha}) \nsubseteq I_k$ for all $k \in [p]$.

Similarly, we also partition the polynomials $g_j, j \in [m+1 : m+l]$ into groups $\{g_j \mid j \in K_k\}_{k\in[p]}$ and $\{g_j \mid j \in K'\}$.

Assume that f decomposes as $f = \sum_{k\in[p]} f_k$ such that $\bigcup_{\boldsymbol{\alpha}\in\text{supp}(f_k)} \text{supp}(\boldsymbol{\alpha}) \subseteq I_k$ for $k \in [p]$. We define the vector of minimum relaxation orders $\mathbf{o} = (o_k)_k \in \mathbb{N}^p$ with $o_k := \max(\{d_j : j \in J_k \cup K_k\} \cup \{\lceil \deg(f_k)/2 \rceil\})$. Then with $s \geq 1$, we define the following initial relaxation step of the CS-TSSOS hierarchy:

$$\begin{cases} \inf_{\mathbf{y}} & L_{\mathbf{y}}(f) \\ \text{s.t.} & B_{G^{(s)}_{o_k,k,0}} \circ \mathbf{M}_{o_k}(\mathbf{y}, I_k) \in \Pi_{G^{(s)}_{o_k,k,0}}(\mathbf{S}_+^{t_{k,0}}), \quad k \in [p], \\ & \mathbf{M}_1(\mathbf{y}, I_k) \succeq 0, \quad k \in [p], \\ & B_{G^{(s)}_{o_k,k,j}} \circ \mathbf{M}_{o_k-d_j}(g_j\mathbf{y}, I_k) \in \Pi_{G^{(s)}_{o_k,k,j}}(\mathbf{S}_+^{t_{k,j}}), \quad j \in J_k, k \in [p], \\ & L_{\mathbf{y}}(g_j) \geq 0, \quad j \in J', \\ & B_{G^{(s)}_{o_k,k,j}} \circ \mathbf{M}_{o_k-d_j}(g_j\mathbf{y}, I_k) = 0, \quad j \in K_k, k \in [p], \\ & L_{\mathbf{y}}(g_j) = 0, \quad j \in K', \\ & y_{\mathbf{0}} = 1, \end{cases} \tag{3.8}$$

where $G^{(s)}_{o_k,k,j}, j \in J_k \cup K_k, k \in [p]$ are defined as in Section 3.2 and $t_{k,j} := \binom{n_k+o_k-d_j}{o_k-d_j}$ for all k, j. Note that in (3.8) we add the PSD constraint on each first-order moment matrix $\mathbf{M}_1(\mathbf{y}, I_k)$ to strengthen the relaxation.

The CS-TSSOS hierarchy is implemented in the Julia package `TSSOS`[5]. In `TSSOS`, the minimal initial relaxation step is accessible via the commands `cs_tssos_first` and `cs_tssos_higher!` by setting the relaxation order to be `"min"`. For an introduction to `TSSOS`, the reader is referred to [26].

4. Problem formulation of AC-OPF

The AC-OPF problem aims to minimize the generation cost of an alternating current transmission network under the physical constraints (Kirchhoff's laws, Ohm's law) as well as operational constraints, which can be formulated as the following POP in complex variables:

$$
\begin{cases}
\inf\limits_{V_i, S_k^g \in \mathbb{C}} & \sum_{k\in G}(\mathbf{c}_{2k}(\Re(S_k^g))^2 + \mathbf{c}_{1k}\Re(S_k^g) + \mathbf{c}_{0k}) \\
\text{s.t.} & \angle V_r = 0, \\
& \mathbf{S}_k^{gl} \le S_k^g \le \mathbf{S}_k^{gu}, \quad \forall k \in G, \\
& \boldsymbol{\upsilon}_i^l \le |V_i| \le \boldsymbol{\upsilon}_i^u, \quad \forall i \in N, \\
& \sum_{k\in G_i} S_k^g - \mathbf{S}_i^d - \mathbf{Y}_i^s|V_i|^2 = \sum_{(i,j)\in E_i\cup E_i^R} S_{ij}, \quad \forall i \in N, \\
& S_{ij} = (\mathbf{Y}_{ij}^* - \mathbf{i}\frac{\mathbf{b}_{ij}^c}{2})\frac{|V_i|^2}{|\mathbf{T}_{ij}|^2} - \mathbf{Y}_{ij}^*\frac{V_iV_j^*}{\mathbf{T}_{ij}}, \quad \forall (i,j) \in E, \\
& S_{ji} = (\mathbf{Y}_{ij}^* - \mathbf{i}\frac{\mathbf{b}_{ij}^c}{2})|V_j|^2 - \mathbf{Y}_{ij}^*\frac{V_i^*V_j}{\mathbf{T}_{ij}^*}, \quad \forall (i,j) \in E, \\
& |S_{ij}| \le \mathbf{s}_{ij}^u, \quad \forall (i,j) \in E \cup E^R, \\
& \boldsymbol{\theta}_{ij}^{\Delta l} \le \angle(V_iV_j^*) \le \boldsymbol{\theta}_{ij}^{\Delta u}, \quad \forall (i,j) \in E.
\end{cases} \tag{4.1}
$$

The meaning of the symbols in (4.1) 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_k^g - the power generation at generator k, 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 ($\mathbf{c}_{0k}, \mathbf{c}_{1k}, \mathbf{c}_{2k}, \boldsymbol{\upsilon}_i^l, \boldsymbol{\upsilon}_i^u, \mathbf{s}_{ij}^u, \boldsymbol{\theta}_{ij}^{\Delta l}, \boldsymbol{\theta}_{ij}^{\Delta u} \in \mathbb{R}$, $\mathbf{S}_k^{gl}, \mathbf{S}_k^{gu}, \mathbf{S}_i^d, \mathbf{Y}_i^s, \mathbf{Y}_{ij}, \mathbf{b}_{ij}^c, \mathbf{T}_{ij} \in \mathbb{C}$). For a full description on the AC-OPF problem, the reader may refer to [21]. By introducing real variables for both real and imaginary parts of each complex

[5] `TSSOS` is freely available at https://github.com/wangjie212/TSSOS.

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 (e.g., `Ipopt` [6]) and then rely on lower bounds obtained from certain relaxation schemes (SOCR/QR/SDR/CS-TSSOS) to certify 1% global optimality. Suppose that the optimum reported by the local solver is AC and the lower bound given by a certain convex relaxation is opt. Then the *optimality gap* is defined by

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

As in [23], if the optimality gap is less than 1%, then we accept the locally optimal solution to be globally optimal.

In our experiments, we eliminate the power flow variables S_{ij} from (4.1) so that it only involves the voltage variable V_i and the power generation variables S_k^g. Because of the inequality $S_{ij}S_{ij}^* \le (\mathbf{s}_{ij}^u)^2$, the resulting optimization problem contains a quartic constraint. To implement the first order relaxation (Shor's relaxation) for QCQPs, we then relax this quartic constraint to a quadratic constraint using the trick described in [16, Sec. 5.3]. The minimal initial relaxation step of the CS-TSSOS hierarchy for (4.1) is able to provide a tighter lower bound than the first order relaxation and is less expensive than the second order relaxation. Therefore, we hereafter refer to it as the 1.5th order relaxation.

5. Experimental settings

Challenging test cases. Our test cases are selected from the AC-OPF library PGLiB v20.07 which provides various AC-OPF instances for benchmarking AC-OPF algorithms. For an introduction to this library, the reader is referred to [21]. We observe that for a number of instances in PGLiB, the SOCR approach is able to close the gap to below 1% and these instances are not particularly interesting as our purpose is to certify 1% global optimality for more challenging cases. To that end, we select test cases from PGLiB (with no more than 25000 buses) for which SOCR yields an optimality gap greater than 1%. There are 115 such instances in total. For each instance,

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

with TSSOS we initially solve the first order relaxation and if this relaxation fails to certify 1% global optimality, we further solve the 1.5th order relaxation with $s = 1$. Here Mosek 9.0 [27] is employed as an SDP solver with the default settings.

Chordal extension. To achieve a good balance between the computational cost and the approximation quality of lower bounds, two types of chordal extensions are used in the computation. For correlative sparsity, we use approximately smallest chordal extensions which give rise to small clique numbers. For term sparsity, instead we use maximal chordal extensions which make every connected component to be a complete subgraph by setting TS = "block" in TSSOS.

Figure 4: Illustration for smallest chordal extension (left) and maximal chordal extension (right): the dashed edges are added via chordal extension

Scaling of polynomial coefficients. To improve the numerical conditioning of the SDP relaxations, we scale the coefficients of f and g_j so that they lie in the interval $[-1, 1]$ before building the SDP relaxations.

Computational resources. Instances with no more than 3500 buses (except 2853_sdet and 2869_pegase) were computed on a laptop with an Intel Core i5-8265U@1.60GHz CPU and 8GB RAM memory; instances with more than 3500 buses (including 2853_sdet and 2869_pegase) were computed on a server with an Intel Xeon E5-2695v4@2.10GHz CPU and 128GB RAM memory.

6. Computational results and discussion

The computational results are summarized in Table 2–4 corresponding to three operational conditions, denoted by "typical", "congested" and "small angle differences", respectively, where the timing includes the time for pre-processing (to obtain the block structure), the time for building SDP and the

Table 1: Notation

AC	local optimum (available from PGLiB)
mc	maximal size of variable cliques
mb	maximal size of SDP blocks
opt	optimum of SDP relaxations
time	running time in seconds
gap	optimality gap (%)
$*$	encountering a numerical error
-	out of memory

time for solving SDP. Note that the maximal size of variable cliques varies from 6 to 218 among the tested cases. According to the tables, we can draw the following conclusions.

Reducing the optimality gap. As we would expect, the 1.5th order relaxation provides tighter lower bounds than the first order relaxation. Indeed, when it is solvable, the 1.5th order relaxation always reduces the optimality gap (unless the lower bounds given by the first order relaxation are already globally optimal). The largest instance for which the 1.5th order relaxation is solvable is 10000_goc and its corresponding POP involves 24032 real variables and 96805 constraints. The improvement of optimality gaps with the 1.5th order relaxation is significant on quite a few cases. For instance, the first order relaxation yields a optimality gap of 42.96% on 30_as under congested operating conditions while the 1.5th order relaxation yields a optimality gap of merely 0.01%. On the other hand, as the cost of these improvements solving the 1.5th order relaxation typically spends significantly more time than solving the first order relaxation.

Certifying 1% global optimality. The first order relaxation is able to certify 1% global optimality for 29 out of all 115 instances. The 1.5th order relaxation is able to certify 1% global optimality for 29 out of the remaining 86 instances. The largest instance for which we are able to certify 1% global optimality with the 1.5th order relaxation is 6515_rte and its corresponding POP involves 14398 real variables and 63577 constraints. One would expect that solving the second order relaxation could certify global optimality for more instances. However this is too expensive to implement for large-scale instances in practice.

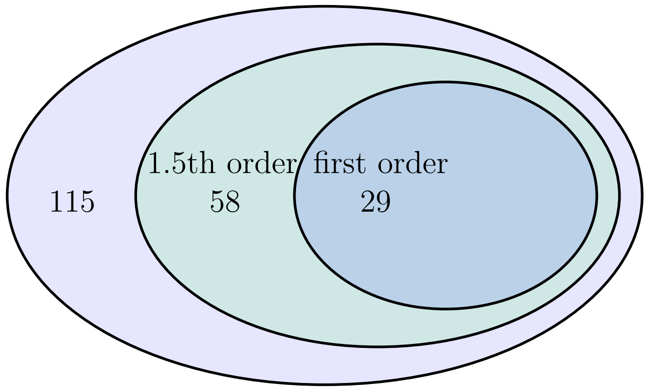

Figure 5: Certifying 1% global optimality for the test cases: the first order relaxation solves 29 cases; the 1.5th order relaxation solves extra 29 cases

Computational burden. The computational burden of the CS-TSSOS relaxations heavily relies on the maximal size of variable cliques. This is because large variable cliques usually lead to SDP matrices of large size in the resulting CS-TSSOS relaxations. It takes 63785 seconds to solve the 1.5th order relaxation for the case 4020_goc under congested operating conditions as it involves a variable clique of size 120. For similar reasons, `Mosek` runs out of memory with the 1.5th order relaxation for the cases 9241_pegase, 9591_goc, 10480_goc, 13659_pegase, 19402_goc, 24464_goc.

Numerical issues. Even though we have scaled polynomial coefficients to improve numerical conditioning of the resulting SDPs, we observe that in numerous cases (especially when solving the 1.5th order relaxation), the termination status of `Mosek` is "slow_progress", which means that `Mosek` does not converge to the default tolerance although the solver usually still gives a fairly good near-optimal solution in this case. Moreover, there are 12 even more challenging instances for which `Mosek` encounters severe numerical issues with the 1.5th order relaxation and fails in converging to the optimum. This indicates that there is still room for improvement in order to tackle these challenging SDPs.

7. Conclusions

We have benchmarked the CS-TSSOS hierarchy on a number of challenging AC-OPF cases and demonstrated that the 1.5th order relaxation is indeed useful in reducing the optimality gap and certifying global optimality

Table 2: The results for AC-OPF problems: typical operating conditions

case name	AC	first order				1.5th order				
		opt	time	mb	gap	mc	opt	time	mb	gap
3_lmbd	5.8126e3	5.7455e3	0.10	5	1.15	6	5.8126e3	0.12	22	**0.00**
5_pjm	1.7552e4	1.4997e4	0.15	6	14.56	6	1.7534e4	0.58	22	**0.10**
30_ieee	8.2085e3	7.5472e3	0.22	8	8.06	8	8.2085e3	0.99	22	**0.00**
162_ieee_dtc	1.0808e5	1.0164e5	2.15	28	5.96	28	1.0645e5	99.1	74	1.51
240_pserc	3.3297e6	3.2512e6	2.39	16	2.36	16	3.3084e6	28.6	44	**0.64**
300_ieee	5.6522e5	5.5423e5	2.72	16	1.94	14	5.6522e5	25.2	40	**0.00**
588_sdet	3.1314e5	3.0886e5	4.37	18	1.37	18	3.1196e5	50.6	32	**0.38**
793_goc	2.6020e5	2.5636e5	5.35	18	1.47	18	2.5932e5	66.1	33	**0.34**
1888_rte	1.4025e6	1.3666e6	30.0	26	2.56	26	1.3756e6	458	56	1.92
2312_goc	4.4133e5	4.3435e5	87.8	68	1.58	68	4.3858e5	997	81	**0.62**
2383wp_k	1.8682e6	1.8584e6	63.0	50	**0.52**	48	1.8646e6	945	77	**0.19**
2742_goc	2.7571e5	2.7561e5	703	92	**0.04**					
2869_pegase	2.4624e6	2.4384e6	85.0	26	**0.97**	26	2.4571e6	3641	191	**0.22**
3012wp_k	2.6008e6	2.5828e6	123	52	**0.69**	52	2.5948e6	1969	81	**0.23**
3022_goc	6.0138e5	5.9277e5	115	48	1.43	50	5.9858e5	1886	76	**0.47**
4020_goc	8.2225e5	8.2208e5	2356	112	**0.02**					
4661_sdet	2.2513e6	2.2246e6	25746	204	1.18	218	*	*	285	*
4917_goc	1.3878e6	1.3658e6	267	64	1.59	68	1.3793e6	29562	110	**0.61**
6468_rte	2.0697e6	2.0546e6	415	54	**0.73**					
6470_rte	2.2376e6	2.2060e6	478	54	1.41	58	*	*	98	*
6495_rte	3.0678e6	2.6327e6	426	56	14.18	54	*	*	108	*
6515_rte	2.8255e6	2.6563e6	460	56	5.99	54	*	*	108	*
9241_pegase	6.2431e6	6.1330e6	982	64	1.76	64	-	-	1268	-
10000_goc	1.3540e6	1.3460e6	1714	84	**0.59**					
10480_goc	2.3146e6	2.3051e6	8559	136	**0.41**					
13659_pegase	8.9480e6	8.8707e6	1808	64	**0.86**					
19402_goc	1.9778e6	1.9752e6	37157	180	**0.13**					

Table 3: The results for AC-OPF problems: congested operating conditions

case name	AC	first order				1.5th order				
		opt	time	mb	gap	mc	opt	time	mb	gap
3_lmbd	1.1236e4	1.0685e4	0.11	5	4.90	6	1.1236e4	0.23	22	**0.00**
5_pjm	7.6377e4	7.3253e4	0.14	6	4.09	6	7.6377e4	0.55	22	**0.00**
14_ieee	5.9994e3	5.6886e3	0.17	6	5.18	6	5.9994e3	0.54	22	**0.00**
24_ieee_rts	1.3494e5	1.2630e5	0.37	10	6.40	10	1.3392e5	1.52	31	**0.76**
30_as	4.9962e3	2.8499e3	0.36	8	42.96	8	4.9959e3	2.41	22	**0.01**
30_ieee	1.8044e4	1.7253e4	0.25	8	4.38	8	1.8044e4	1.24	22	**0.00**
39_epri	2.4967e5	2.4522e5	0.28	8	1.78	8	2.4966e5	2.72	25	**0.00**
73_ieee_rts	4.2263e5	3.9912e5	0.76	12	5.56	12	4.1495e5	6.77	36	1.82
89_pegase	1.2781e5	1.0052e5	1.13	24	21.35	24	1.0188e5	1404	184	20.29
118_ieee	2.4224e5	1.9375e5	1.80	10	20.02	10	2.2151e5	11.5	37	8.56
162_ieee_dtc	1.2099e5	1.1206e5	1.97	28	7.38	28	1.1955e5	84.1	74	1.19
179_goc	1.9320e6	1.7224e6	1.24	10	10.85	10	1.9226e6	9.69	37	**0.48**
500_goc	6.9241e5	6.6004e5	4.31	18	4.67	18	6.7825e5	78.0	50	2.05
588_sdet	3.9476e5	3.9026e5	6.42	18	1.14	18	3.9414e5	57.0	32	**0.15**
793_goc	3.1885e5	2.9796e5	6.18	18	6.55	18	3.1386e5	79.2	33	1.56
2000_goc	1.4686e6	1.4147e6	54.0	42	3.67	42	1.4610e6	1094	62	**0.51**
2312_goc	5.7152e5	4.7872e5	93.4	68	16.24	68	5.2710e5	972	81	7.77
2736sp_k	6.5394e5	5.8042e5	89.6	50	11.24	48	*	*	79	*
2737sop_k	3.6531e5	3.4557e5	71.9	48	5.40	48	3.4557e5	1653	77	5.40
2742_goc	6.4219e5	5.0824e5	772	92	20.86	90	6.0719e5	4644	108	5.45
2853_sdet	2.4578e6	2.3869e6	118	40	2.88	40	2.4445e6	10292	293	**0.54**
2869_pegase	2.9858e6	2.9604e6	90.2	26	**0.85**	26	2.9753e6	5409	191	**0.35**
3022_goc	6.5185e5	6.2343e5	102	48	4.36	50	6.4070e5	1519	76	1.71
3120sp_k	9.3599e5	7.6012e5	138	52	18.79	58	8.5245e5	1627	70	8.93
3375wp_k	5.8460e6	5.5378e6	222	58	5.27	54	5.7148e6	2619	90	2.25
3970_goc	1.4241e6	1.0087e6	2469	104	29.17	98	1.0719e6	15482	135	24.73
4020_goc	1.2979e6	1.0836e6	3523	112	16.51	120	1.1218e6	63785	174	13.57
4601_goc	7.9253e5	6.7523e5	2143	108	14.80	98	7.3914e5	17249	125	6.74
4619_goc	1.0299e6	9.6351e5	1782	82	6.45	84	9.9766e5	18348	132	3.13
4661_sdet	2.6953e6	2.6112e6	15822	204	3.12	218	*	*	285	*
4837_goc	1.1578e6	1.0769e6	500	80	6.98	84	1.0947e6	8723	132	5.45
4917_goc	1.5479e6	1.4670e6	259	64	5.23	68	1.5180e6	4688	110	1.93
6470_rte	2.6065e6	2.5795e6	427	54	1.04	58	*	*	98	*
6495_rte	2.9750e6	2.9092e6	453	56	2.21	54	*	*	108	*
6515_rte	3.0617e6	2.9996e6	421	56	2.02	54	3.0434e6	8456	108	**0.60**
9241_pegase	7.0112e6	6.8784e6	865	64	1.89	64	-	-	1268	-
9591_goc	1.4259e6	1.2425e6	7674	148	12.86	134	-	-	201	-
10000_goc	2.3728e6	2.1977e6	2564	84	7.38	84	2.3206e6	27179	97	2.20
10480_goc	2.7627e6	2.6580e6	8791	136	3.79	132	-	-	208	-
13659_pegase	9.2842e6	9.1360e6	1599	64	1.60	64	-	-	1268	-
19402_goc	2.3987e6	2.3290e6	32465	180	2.91	172	-	-	242	-
24464_goc	2.4723e6	2.4177e6	11760	116	2.21	118	-	-	172	-

Table 4: The results for AC-OPF problems: small angle difference conditions

case name	AC	first order				1.5th order				
		opt	time	mb	gap	mc	opt	time	mb	gap
3_lmbd	5.9593e3	5.7463e3	0.11	5	3.57	6	5.9593e3	0.13	22	**0.00**
5_pjm	2.6109e4	2.6109e4	0.12	6	**0.00**					
14_ieee	2.7768e3	2.7743e3	0.14	6	**0.09**					
24_ieee_rts	7.6918e4	7.3555e4	0.21	10	4.37	10	7.4852e4	1.64	31	2.69
30_as	8.9735e2	8.9527e2	0.19	8	**0.23**					
30_ieee	8.2085e3	7.5472e3	0.30	8	8.06	8	8.2085e3	1.03	22	**0.00**
73_ieee_rts	2.2760e5	2.2136e5	0.55	12	2.74	12	2.2447e5	5.01	36	1.38
118_ieee	1.0516e5	1.0191e5	0.79	10	3.10	10	1.0313e5	10.8	37	1.93
162_ieee_dtc	1.0869e5	1.0282e5	2.46	28	5.40	28	1.0740e5	105	74	1.19
179_goc	7.6186e5	7.5261e5	1.39	10	1.21	10	7.5573e5	11.8	37	**0.80**
240_pserc	3.4054e6	3.2772e6	3.00	16	3.76	16	3.3128e6	33.8	44	2.72
300_ieee	5.6570e5	5.6162e5	2.73	16	**0.72**	14	5.6570e5	25.2	40	**0.00**
500_goc	4.8740e5	4.6043e5	6.52	18	5.53	18	4.6098e5	67.7	50	5.42
588_sdet	3.2936e5	3.1233e5	5.12	18	5.17	18	3.1898e5	56.6	32	3.15
793_goc	2.8580e5	2.7133e5	5.61	18	5.06	18	2.7727e5	76.0	33	2.98
1354_pegase	1.2588e6	1.2172e6	19.8	26	3.31	26	1.2582e6	387	49	**0.05**
1888_rte	1.4139e6	1.3666e6	31.2	26	3.34	26	1.3756e6	497	56	2.71
2000_goc	9.9288e5	9.8400e5	50.9	42	**0.89**	42	9.8435e5	1052	62	**0.86**
2312_goc	4.6235e5	4.4719e5	121	68	3.28	68	4.5676e5	1009	81	1.21
2383wp_k	1.9112e6	1.9041e6	65.6	50	**0.37**	48	1.9060e6	937	77	**0.27**
2736sp_k	1.3266e6	1.3229e6	89.5	50	**0.28**					
2737sop_k	7.9095e5	7.8672e5	76.3	48	**0.53**					
2742_goc	2.7571e5	2.7561e5	686	92	**0.04**					
2746wop_k	1.2337e6	1.2248e6	79.1	48	**0.72**					
2746wp_k	1.6669e6	1.6601e6	83.1	50	**0.41**					
2853_sdet	2.0692e6	2.0303e6	106	40	1.88	40	2.0537e6	40671	293	**0.75**
2869_pegase	2.4687e6	2.4477e6	85.4	26	**0.85**					
3012wp_k	2.6192e6	2.5994e6	97.1	52	**0.76**					
3022_goc	6.0143e5	5.9278e5	93.4	48	1.44	50	5.9859e5	1340	76	**0.47**
3120sp_k	2.1749e6	2.1611e6	117	52	**0.64**					
4020_goc	8.8969e5	8.4238e5	2746	112	5.32	120	8.7038e5	43180	174	2.17
4601_goc	8.7803e5	8.3370e5	1763	108	5.05	98	8.3447e5	15585	125	4.96
4619_goc	4.8435e5	4.8106e5	1387	82	**0.68**					
4661_sdet	2.2610e6	2.2337e6	16144	204	1.21	218	$*$	$*$	285	$*$
4917_goc	1.3890e6	1.3665e6	260	64	1.62	68	1.3800e6	4914	110	**0.65**
6468_rte	2.0697e6	2.0546e6	399	54	**0.73**					
6470_rte	2.2416e6	2.2100e6	451	54	1.41	58	$*$	$*$	98	$*$
6495_rte	3.0678e6	2.6323e6	404	56	14.19	54	$*$	$*$	108	$*$
6515_rte	2.8698e6	2.6565e6	399	56	7.43	54	$*$	$*$	108	$*$
9241_pegase	6.3185e6	6.1696e6	912	64	2.36	64	-	-	1268	-
9591_goc	1.1674e6	1.0712e6	7835	148	8.24	134	-	-	201	-
10000_goc	1.4902e6	1.4204e6	2508	84	4.68	84	1.4212e6	25340	97	4.63
10480_goc	2.3147e6	2.3051e6	6522	136	**0.42**					
13659_pegase	9.0422e6	8.9142e6	1653	64	1.42	64	-	-	1268	-
19402_goc	1.9838e6	1.9783e6	30122	180	**0.28**					
24464_goc	2.6540e6	2.6268e6	12101	116	1.03	118	-	-	172	-

of AC-OPF solutions. We hope that the computational results presented in this paper would convince people to think of CS-TSSOS as an alternative tool for certifying global optimality of solutions of large-scale POPs. One line of future research is to improve the efficiency of the CS-TSSOS relaxations by relying on more advanced chordal extension algorithms. We also plan to design suitable branch and bound algorithms to reach better accuracy results such as 0.1% or 0.01% global optimality for the AC-OPF problem.

Acknowledgements. We would like to thank Jean Maeght, Patrick Panciatici and Manual Ruiz for interesting discussions and insights regarding AC-OPF. Both authors were supported by the Tremplin ERC Stg Grant ANR-18-ERC2-0004-01 (T-COPS project). The second author was supported by the FMJH Program PGMO (EPICS project), as well as the PEPS2 Program (FastOPF project) funded by AMIES and RTE. This work has benefited from the European Union's Horizon 2020 research and innovation program 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.

References

[1] B. Ghaddar, J. Marecek, M. Mevissen, Optimal power flow as a polynomial optimization problem, IEEE Transactions on Power Systems 31 (1) (2015) 539–546.

[2] V. Magron, G. Constantinides, A. Donaldson, Certified Roundoff Error Bounds Using Semidefinite Programming, ACM Trans. Math. Softw. 43 (4) (2017) 1–34.

[3] H. Yang, L. Carlone, One ring to rule them all: Certifiably robust geometric perception with outliers, arXiv preprint arXiv:2006.06769 (2020).

[4] T. Chen, J. B. Lasserre, V. Magron, E. Pauwels, Semialgebraic Optimization for Lipschitz Constants of ReLU Networks, in: H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, Vol. 33, Curran Associates, Inc., 2020, pp. 19189–19200.

[5] L. Slot, M. Laurent, Sum-of-squares hierarchies for binary polynomial optimization, arXiv preprint arXiv:2011.04027 (2020).

[6] A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Mathematical programming 106 (1) (2006) 25–57.

[7] J.-B. Lasserre, Global Optimization with Polynomials and the Problem of Moments, SIAM Journal on Optimization 11 (3) (2001) 796–817.

[8] J. Nie, Optimality conditions and finite convergence of Lasserre's hierarchy, Mathematical Programming 146 (1) (2014) 97–121.

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

[10] C. Riener, T. Theobald, L. J. Andrén, J. B. Lasserre, Exploiting symmetries in SDP-relaxations for polynomial optimization, Mathematics of Operations Research 38 (1) (2013) 122–141.

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

[12] H. Waki, S. Kim, M. Kojima, M. Muramatsu, Sums of Squares and Semidefinite Programming Relaxations for Polynomial Optimization Problems with Structured Sparsity, SIAM Journal on Optimization 17 (1) (2006) 218–242.

[13] J. Wang, V. Magron, J.-B. Lasserre, Chordal-TSSOS: a moment-SOS hierarchy that exploits term sparsity with chordal extension, SIAM Journal on OptimizationAccepted for publication (2020).

[14] J. Wang, V. Magron, J.-B. Lasserre, TSSOS: A moment-SOS hierarchy that exploits term sparsity, SIAM Journal on Optimization 31 (1) (2021) 30–58.

[15] J. Wang, V. Magron, J.-B. Lasserre, N. H. A. Mai, CS-TSSOS: Correlative and term sparsity for large-scale polynomial optimization, arXiv:2005.02828 (2020).

[16] D. Bienstock, M. Escobar, C. Gentile, L. Liberti, Mathematical programming formulations for the alternating current optimal power flow problem, 4OR 18 (3) (2020) 249–292.

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

[18] R. A. Jabr, Radial distribution load flow using conic programming, IEEE transactions on power systems 21 (3) (2006) 1458–1459.

[19] C. Coffrin, H. L. Hijazi, P. Van Hentenryck, The QC relaxation: A theoretical and computational study on optimal power flow, IEEE Transactions on Power Systems 31 (4) (2015) 3008–3018.

[20] X. Bai, H. Wei, K. Fujisawa, Y. Wang, Semidefinite programming for optimal power flow problems, International Journal of Electrical Power & Energy Systems 30 (6-7) (2008) 383–392.

[21] S. Babaeinejadsarookolaee, A. Birchfield, R. D. Christie, C. Coffrin, C. DeMarco, R. Diao, M. Ferris, S. Fliscounakis, S. Greene, R. Huang, et al., The power grid library for benchmarking AC optimal power flow algorithms, arXiv preprint arXiv:1908.02788 (2019).

[22] A. Eltved, J. Dahl, M. S. Andersen, On the robustness and scalability of semidefinite relaxation for optimal power flow problems, Optimization and Engineering (2019) 1–18.

[23] S. Gopinath, H. L. Hijazi, T. Weisser, H. Nagarajan, M. Yetkin, K. Sundar, R. W. Bent, Proving global optimality of ACOPF solutions, Electric Power Systems Research 189 (2020).

[24] R. Grone, C. R. Johnson, E. M. Sá, H. Wolkowicz, Positive definite completions of partial hermitian matrices, Linear algebra and its applications 58 (1984) 109–124.

[25] J. Wang, V. Magron, Exploiting sparsity in complex polynomial optimization, arXiv preprint arXiv:2103.12444 (2021).

[26] V. Magron, J. Wang, TSSOS: a Julia library to exploit sparsity for large-scale polynomial optimization, arXiv preprint arXiv:2103.00915 (2021).

[27] M. ApS, The MOSEK optimization software (2019).
URL https://www.mosek.com/