A Method for Quickly Bounding the Optimal Objective Value of an OPF Problem using a Semidefinite Relaxation and a Local Solution

Alireza Barzegar,* *Student Member, IEEE*, Daniel K. Molzahn,† *Member, IEEE*, and Rong Su,* *Senior Member, IEEE*

***Abstract*—Optimal power flow (OPF) is an important problem in the operation of electric power systems. Due to the OPF problem's non-convexity, there may exist multiple local optima. Certifiably obtaining the global solution is important for certain applications of OPF problems. Many global optimization techniques compute an optimality gap that compares the achievable objective value corresponding to the feasible point from a local solution algorithm with the objective value bound from a convex relaxation technique. Rather than the traditional practice of completely separating the local solution and convex relaxation computations, this paper proposes a method that exploits information from a local solution to speed the computation of an objective value bound using a semidefinite programming (SDP) relaxation. The improvement in computational tractability comes with the trade-off of reduced tightness for the resulting objective value bound. Numerical experiments illustrate this trade-off, with the proposed method being faster but weaker than the SDP relaxation and slower but tighter than second-order cone programming (SOCP) and quadratic convex (QC) relaxations for many large test cases.**

***Index Terms*—Optimal power flow, Convex optimization, Global solution, Lower bound**

I. Introduction

FIRST formulated by Carpentier [1] in 1962, the optimal power flow (OPF) problem determines a steady-state operating point for an electric power system by optimizing an objective function, such as generation cost, while satisfying constraints that model the nonlinear power flow equations as well as limits on line-flows, voltage magnitudes, and generator outputs. The OPF problem is non-convex, NP-Hard [2], [3], and may have multiple *local optima* (i.e., feasible points that are superior to all nearby feasible points but possibly inferior to the *global optimum*) [4].

Research conducted for over fifty years has developed many mature local solution algorithms for OPF problems, including Newton-based techniques, sequential quadratic programming algorithms, interior point methods, etc. [5]–[7]. These algorithms often quickly find local optima for large OPF problems.

Certifiably finding global solutions to OPF problems is important for many applications. For instance, local optima in generation cost minimization problems may be significantly more expensive than the global solutions [4]. Additionally, some bi-level problems, e.g., those used in robust OPF algorithms [8], benefit from certificates of global optimality (or, at least, strong objective value bounds) for OPF subproblems.

*: School of Electrical and Electronic Engineering, Nanayang Technological University, Singapore, alireza001@e.ntu.edu.sg, rsu@ntu.edu.sg.
†: Energy Systems Div., Argonne National Laboratory, dmolzahn@anl.gov.

To compute global optimality certificates, recent research has focused on convex relaxations of OPF problems [9]. Convex relaxations bound the objective value of the original OPF problem, providing lower (upper) bounds for minimization (maximization) problems.

Comparing the bound provided by a relaxation and the objective value obtained from a local solver yields an *optimality gap*. A sufficiently small optimality gap certifies global optimality of the corresponding local solution. Many global solution algorithms use branch-and-bound methods that iteratively apply local solvers and convex relaxation techniques in order to shrink the optimality gap until global optimality is certifiably achieved [10]–[12].

Accordingly, the tightness of the bound provided by a convex relaxation is of primary importance for many applications. Various convex relaxation techniques provide a range of trade-offs between bound tightness and computational tractability. Simple relaxations, such as copper plate and network flow formulations [13], quickly provide crude bounds. More sophisticated approaches, such as those based on second-order cone programming (SOCP) [14]–[17] and semidefinite programming (SDP) [18], [19], provide tighter bounds with associated trade-offs in computational speed. The literature details a variety of approaches for refining trade-offs in computational speed and tightness of the relaxations (e.g., adding valid cuts [16] and exploiting problem structure [19]).

Building on the convex relaxation literature, this paper proposes a new method for balancing trade-offs between computational tractability and tightness. We specifically consider a method for obtaining an objective value bound using the SDP relaxation in [18] and the dual variables from a local solution.

The SDP relaxation in [18] is derived from an OPF formulation where all non-convexity is contained in a rank condition. Following Shor's approach [20], an SDP relaxation is formed by neglecting this rank condition. A relaxation whose solution satisfies the rank condition is "exact", thus providing a tight bound on the objective value and the globally optimal decision variables. While the SDP relaxation in [18] is exact for some OPF test cases, the rank condition is not always satisfied [21]. Nevertheless, the objective value from the SDP relaxation always bounds the objective of the original OPF problem.

Solving the SDP relaxation can be computationally challenging due to the presence of a positive semidefinite constraint on a large matrix. A matrix completion technique mitigates this computational challenge by decomposing the

single positive semidefinite constraint into constraints on many submatrices [22], [23]. However, the resulting formulation is still less tractable than many local solution algorithms. This motivates the development of techniques for exploiting information in a local solution to further speed computations of SDP relaxations.

Related work [24] proposes a quickly checkable sufficient condition for global optimality of a candidate OPF solution from a local solver. Satisfaction of the Karush-Kuhn-Tucker (KKT) optimality conditions for the SDP relaxation certifies that a local solution is, in fact, globally optimal. Since the KKT conditions for the SDP relaxation can be quickly evaluated for a candidate OPF solution without explicitly solving an SDP, the condition in [24] provides a tractable method for certifying global optimality.

While the approach in [24] provides significant computational advantages for certain problems, the condition is sufficient but not necessary for global optimality and thus does not provide any information regarding global optimality when the condition is not satisfied. This paper proposes another method for leveraging information from a local solution to speed the computation of an objective value bound. Specifically, we seek a dual feasible point for the SDP relaxation. The objective value of any dual feasible point bounds the objective value of the primal SDP relaxation [25], which itself bounds the objective value of the original OPF problem. Rather than attempt the computationally difficult task of computing the best achievable bounds, we instead leverage information from local solutions to more tractably obtain (possibly weaker) bounds. In particular, we fix certain dual variables in the SDP relaxation to their values from a local solution and solve the resulting simpler SDP problem.

Application to a variety of large test cases empirically demonstrates that the proposed method is faster than the SDP relaxation of [18], often without excessive degradation in the quality of the objective value bound. Conversely, empirical results indicate that the proposed method is slower than the SOCP relaxation in [14] and the QC relaxation in [17] but can provide significantly tighter bounds. Thus, the proposed method can be viewed as a "middle ground" between the SDP relaxation and the SOCP and QC relaxations in terms of the trade-off between computational speed and tightness.

This paper is organized as follows. Section II formulates the OPF problem and the SDP relaxation as well as the matrix completion decomposition. Section III proposes our method for quickly computing objective value bounds using the SDP relaxation in combination with a local solution, which is the main contribution of this paper. Section IV numerically demonstrates this method using a variety of large test cases. Section V concludes the paper.

II. The OPF Problem and an SDP Relaxation

This section first formulates the OPF problem as well as its SDP relaxation [18] and then discusses the matrix completion decomposition used to exploit network sparsity [22].

A. Optimal Power Flow Formulation

Consider an n-bus system with the set of buses $\mathcal{N} = \{1, \ldots, n\}$. Each bus $k \in \mathcal{N}$ has an associated complex voltage phasor $V_k = V_{dk} + jV_{qk}$, power generation $P_{Gk} + jQ_{Gk}$, and specified power demand $P_{Dk} + jQ_{Dk}$, where $j = \sqrt{-1}$. The bus which sets the reference angle is denoted with the subscript "ref". The set of lines is denoted as $\mathcal{L}$, where $(l, m) \in \mathcal{L}$ indicates the line connecting buses l and m. Each line $(l, m) \in \mathcal{L}$ is modeled as a Π circuit with mutual admittance y_{lm} and shunt susceptance $b_{sh,lm}$.[1] The complex power flow and angle difference on each line $(l, m) \in \mathcal{L}$ are denoted by $S_{lm} = P_{lm} + jQ_{lm}$ and $\theta_{lm} = \theta_l - \theta_m$, respectively. The network admittance matrix is denoted as $\mathbf{Y} = \mathbf{G} + j\mathbf{B}$. Superscripts "max" and "min" indicate specified upper and lower limits. Buses without generators have the corresponding generation limits set to zero.

Denote e_k as the k^{th} standard basis vector in $\mathbb{R}^n$. Define the matrix $Y_k = e_k e_k^\mathsf{T} \mathbf{Y}$, where $(\,\cdot\,)^\mathsf{T}$ denotes the matrix transpose. Following the notation in [18], define the matrices $\mathbf{Y}_k$, $\overline{\mathbf{Y}}_k$, $\mathbf{M}_k$, and $\mathbf{N}_k$ used to construct expressions for the active and reactive power injections, the squared voltage magnitudes, and the angle reference, respectively, at bus $k \in \mathcal{N}$:

$$\mathbf{Y}_k = \frac{1}{2}\begin{bmatrix} \mathrm{Re}(Y_k + Y_k^\mathsf{T}) & \mathrm{Im}(Y_k^\mathsf{T} - Y_k) \\ \mathrm{Im}(Y_k - Y_k^\mathsf{T}) & \mathrm{Re}(Y_k + Y_k^\mathsf{T}) \end{bmatrix}, \tag{1a}$$

$$\overline{\mathbf{Y}}_k = -\frac{1}{2}\begin{bmatrix} \mathrm{Im}(Y_k + Y_k^\mathsf{T}) & \mathrm{Re}(Y_k - Y_k^\mathsf{T}) \\ \mathrm{Re}(Y_k^\mathsf{T} - Y_k) & \mathrm{Im}(Y_k + Y_k^\mathsf{T}) \end{bmatrix}, \tag{1b}$$

$$\mathbf{M}_k = \begin{bmatrix} e_k e_k^\mathsf{T} & 0 \\ 0 & e_k e_k^\mathsf{T} \end{bmatrix}, \tag{1c}$$

$$\mathbf{N}_k = \begin{bmatrix} 0 & 0 \\ 0 & e_k e_k^\mathsf{T} \end{bmatrix}. \tag{1d}$$

For each line $(l, m) \in \mathcal{L}$, let $Y_{lm} = \left(\frac{b_{sh,lm}}{2} + y_{lm}\right) e_l e_l^\mathsf{T} - y_{lm} e_l e_m^\mathsf{T}$. To formulate the expressions for active and reactive power flows and phase angle differences associated with the line $(l, m) \in \mathcal{L}$, define the matrices

$$\mathbf{Y}_{lm} = \frac{1}{2}\begin{bmatrix} \mathrm{Re}(Y_{lm} + Y_{lm}^\mathsf{T}) & \mathrm{Im}(Y_{lm} - Y_{lm}^\mathsf{T}) \\ \mathrm{Im}(Y_{lm}^\mathsf{T} - Y_{lm}) & \mathrm{Re}(Y_{lm} + Y_{lm}^\mathsf{T}) \end{bmatrix}, \tag{2a}$$

$$\overline{\mathbf{Y}}_{lm} = -\frac{1}{2}\begin{bmatrix} \mathrm{Im}(Y_{lm} + Y_{lm}^\mathsf{T}) & \mathrm{Re}(Y_{lm} - Y_{lm}^\mathsf{T}) \\ \mathrm{Re}(Y_{lm}^\mathsf{T} - Y_{lm}) & \mathrm{Im}(Y_{lm} + Y_{lm}^\mathsf{T}) \end{bmatrix}, \tag{2b}$$

$$\mathbf{M}_{lm} = \frac{1}{2}\begin{bmatrix} e_l e_m^\mathsf{T} + e_m e_l^\mathsf{T} & 0 \\ 0 & e_l e_m^\mathsf{T} + e_m e_l^\mathsf{T} \end{bmatrix}, \tag{2c}$$

$$\overline{\mathbf{M}}_{lm} = \frac{1}{2}\begin{bmatrix} 0 & -e_l e_m^\mathsf{T} + e_m e_l^\mathsf{T} \\ e_l e_m^\mathsf{T} - e_m e_l^\mathsf{T} & 0 \end{bmatrix}. \tag{2d}$$

By denoting the vector of voltage coordinates x as

$$x = [V_{d1} \quad V_{d2} \quad \ldots \quad V_{dn} \quad V_{q1} \quad V_{q2} \quad \ldots \quad V_{qn}] \tag{3}$$

and defining the rank-one matrix

$$\mathbf{W} = xx^\mathsf{T}, \tag{4}$$

all quantities of interest can be written in terms of $\mathbf{W}$. Let $\mathrm{tr}\,(\cdot)$ denote the trace of a matrix. The active and reactive power injections at each bus k are $\mathrm{tr}\,(\mathbf{Y}_k \mathbf{W})$ and $\mathrm{tr}\,(\overline{\mathbf{Y}}_k \mathbf{W})$,

[1] Extending our method to more general line models is straightforward. The numerical results in Section IV use MATPOWER's line model [7].

respectively. The squared voltage magnitude at bus k is $\mathrm{tr}\,(\mathbf{M}_k\mathbf{W})$. The active and reactive power flows into terminal l of line $(l,m)\in\mathcal{L}$ are $\mathrm{tr}\,(\mathbf{Y}_{lm}\mathbf{W})$ and $\mathrm{tr}\,\big(\overline{\mathbf{Y}}_{lm}\mathbf{W}\big)$, and the phase angle difference is $\arctan\big(\mathrm{tr}\,\big(\overline{\mathbf{M}}_{lm}\mathbf{W}\big)/\mathrm{tr}\,(\mathbf{M}_{lm}\mathbf{W})\big)$.

We consider a convex quadratic cost function of active power generation with specified quadratic, linear, and constant coefficients $c_{k2}\geq 0$, c_{k1}, and c_{k0} for each generator $k\in\mathcal{N}$. Buses without generators have $c_{k2}=c_{k1}=c_{k0}=0$.

Using these definitions, the OPF problem is

$$\min \quad \sum_{k\in\mathcal{N}} \alpha_k \tag{5a}$$

$$\text{subject to} \quad (\,\forall k\in\mathcal{N},\ (l,m)\in\mathcal{L}\,)$$

$$P_{Gk}^{min}-P_{Dk}\leq \mathrm{tr}\,(\mathbf{Y}_k\mathbf{W})\leq P_{Gk}^{max}-P_{Dk}, \tag{5b}$$

$$Q_{Gk}^{min}-Q_{Dk}\leq \mathrm{tr}\,\big(\overline{\mathbf{Y}}_k\mathbf{W}\big)\leq Q_{Gk}^{max}-Q_{Dk}, \tag{5c}$$

$$(V_k^{min})^2\leq \mathrm{tr}\,(\mathbf{M}_k\mathbf{W})\leq (V_k^{max})^2, \tag{5d}$$

$$\mathrm{tr}\,(\mathbf{N}_{ref}\mathbf{W})=0, \tag{5e}$$

$$\begin{bmatrix} -(S_{lm}^{max})^2 & \mathrm{tr}\,(\mathbf{Y}_{lm}\mathbf{W}) & \mathrm{tr}\,\big(\overline{\mathbf{Y}}_{lm}\mathbf{W}\big)\\ \mathrm{tr}\,(\mathbf{Y}_{lm}\mathbf{W}) & -1 & 0\\ \mathrm{tr}\,\big(\overline{\mathbf{Y}}_{lm}\mathbf{W}\big) & 0 & -1\end{bmatrix}\preceq 0, \tag{5f}$$

$$\tan(\theta_{lm}^{min})\cdot\mathrm{tr}\,(\mathbf{M}_{lm}\mathbf{W})\leq \mathrm{tr}\,\big(\overline{\mathbf{M}}_{lm}\mathbf{W}\big) \leq \tan(\theta_{lm}^{max})\cdot\mathrm{tr}\,(\mathbf{M}_{lm}\mathbf{W}), \tag{5g}$$

$$\begin{bmatrix} c_{k1}\mathrm{tr}\,(\mathbf{Y}_k\mathbf{W})+a_k-\alpha_k & \sqrt{c_{k2}}\mathrm{tr}\,(\mathbf{Y}_k\mathbf{W})+b_k\\ \sqrt{c_{k2}}\mathrm{tr}\,(\mathbf{Y}_k\mathbf{W})+b_k & -1\end{bmatrix}\preceq 0, \tag{5h}$$

$$\mathbf{W}=xx^{\mathsf{T}}, \tag{5i}$$

where $a_k=c_{k1}P_{Dk}+c_{k0}$, $b_k=\sqrt{c_{k2}}P_{Dk}$, and $(\,\cdot\,)\succeq 0$ indicates positive semidefinteness. Constraints (5b)–(5d) limit the power injections and squared voltage magnitudes, (5e) sets the reference angle, and (5f) and (5g) limit the apparent power flows and phase angle differences. The objective (5a) and constraint (5h) represent the generation cost using a Schur complement formulation with auxiliary variables α_k. All non-convexity is contained in the rank-one condition (5i).

B. Semidefinite Programming Relaxation of the OPF Problem

The approach in [18] forms a Shor relaxation [20] of (5) by replacing the rank-one condition (5i) with a less stringent positive semidefinite matrix constraint,

$$\mathbf{W}\succeq 0. \tag{6}$$

This yields the primal formulation of the SDP relaxation:

$$\min \quad \sum_{k\in\mathcal{N}}\alpha_k \quad \text{subject to} \quad \text{(5b)–(5h), (6).} \tag{7}$$

If the solution to the SDP relaxation, $\mathbf{W}^\star$, satisfies the rank condition $\mathrm{rank}\,(\mathbf{W}^\star)=1$, then the relaxation is *exact* and thus provides both the globally optimal objective value and decision variables $V^\star=\sqrt{\lambda}\,(\eta_{1:n}+j\eta_{n+1:2n})$, where λ denotes the non-zero eigenvalue of $\mathbf{W}^\star$ with corresponding unit-length eigenvector η. While solutions for which $\mathrm{rank}\,(\mathbf{W}^\star)>1$ do not provide globally optimal decision variables, the optimal objective value from the solution to (7) still lower bounds the objective value of the original OPF problem (5).

Our method for speeding computations is based on the dual form of the SDP relaxation. To formulate the dual of (7), we define (for each bus $k\in\mathcal{N}$ and each line $(l,m)\in\mathcal{L}$) the dual scalar variables $\underline{\lambda}_k$, $\underline{\gamma}_k$, $\underline{\mu}_k$, and $\underline{\beta}_{lm}$ associated with the lower bounds of (5b), (5c), (5d) and (5g); the dual scalar variables $\overline{\lambda}_k$, $\overline{\gamma}_k$, $\overline{\mu}_k$, and $\overline{\beta}_{lm}$ associated with the upper bounds of (5b), (5c), (5d) and (5g); the dual 3×3 matrix variables $\mathbf{H}_{lm}$ associated with line-flow constraints (5f); and the dual 2×2 matrix variables $\mathbf{R}_k$ associated with the Schur complement formulation of the quadratic cost function (5h).

For notational convenience, denote

$$\lambda_k=\overline{\lambda}_k-\underline{\lambda}_k+c_{k1}+2\sqrt{c_{k2}}R_k^{12},\quad \gamma_k=\overline{\gamma}_k-\underline{\gamma}_k,$$
$$\mu_k=\overline{\mu}_k-\underline{\mu}_k,\quad \beta_{lm}=\overline{\beta}_{lm}-\underline{\beta}_{lm}.$$

The dual formulation of the SDP relaxation (7) is

$$\max \quad \rho \tag{8a}$$

$$\text{subject to} \quad (\,\forall k\in\mathcal{N},\ (l,m)\in\mathcal{L}\,)$$

$$\mathbf{A}\succeq 0, \tag{8b}$$

$$\mathbf{H}_{lm}\succeq 0, \tag{8c}$$

$$\mathbf{R}_k\succeq 0,\quad \mathbf{R}_k^{11}=1, \tag{8d}$$

$$\underline{\lambda}_k\geq 0,\ \overline{\lambda}_k\geq 0,\ \underline{\gamma}_k\geq 0,\ \overline{\gamma}_k\geq 0,\ \underline{\mu}_k\geq 0,\ \overline{\mu}_k\geq 0, \tag{8e}$$

$$\underline{\beta}_{lm}\geq 0,\ \overline{\beta}_{lm}\geq 0, \tag{8f}$$

where the scalar- and matrix-valued functions ρ and $\mathbf{A}$ are

$$\rho=\sum_{k\in\mathcal{N}}\Big\{\lambda_kP_{Dk}+\underline{\lambda}_kP_{Gk}^{min}-\overline{\lambda}_kP_{Gk}^{max}+\gamma_kQ_{Dk}+\underline{\gamma}_kQ_{Gk}^{min} -\overline{\gamma}_kQ_{Gk}^{max}+\underline{\mu}_k(V_k^{min})^2-\overline{\mu}_k(V_k^{max})^2\Big\}+\sum_{k\in\mathcal{N}}\Big(c_{k0}-\mathbf{R}_k^{22}\Big) -\sum_{(l,m)\in\mathcal{L}}\Big\{(S_{lm}^{max})^2\mathbf{H}_{lm}^{11}+\mathbf{H}_{lm}^{22}+\mathbf{H}_{lm}^{33}\Big\}, \tag{9}$$

$$\mathbf{A}=\sum_{k\in\mathcal{N}}\Big\{\lambda_k\mathbf{Y}_k+\gamma_k\overline{\mathbf{Y}}_k+\mu_k\mathbf{M}_k\Big\} +\sum_{(l,m)\in\mathcal{L}}\Big\{2\mathbf{H}_{lm}^{12}\mathbf{Y}_{lm}+2\mathbf{H}_{lm}^{13}\overline{\mathbf{Y}}_{lm}^{13}-\overline{\mathbf{M}}_{lm}\beta_{lm} +\tan\Big(\theta_{lm}^{max}\Big)\mathbf{M}_{lm}\overline{\beta}_{lm}-\tan\Big(\theta_{lm}^{min}\Big)\mathbf{M}_{lm}\underline{\beta}_{lm}\Big\}. \tag{10}$$

Note that $\mathbf{R}_k^{cd}$ and $\mathbf{H}_{lm}^{cd}$ represent the (c,d) elements of matrices $\mathbf{R}_k$ and $\mathbf{H}_{lm}$, respectively.

Any feasible point for the dual SDP relaxation (8) provides a lower bound for the objective value of the primal SDP relaxation (7) [25], and thus a lower bound for the original OPF problem (5). The solution to (8) provides the best achievable lower bound among all dual feasible points.

C. Matrix Completion Decomposition

The positive semidefinite constraint on a $2n\times 2n$ matrix ($\mathbf{W}\succeq 0$ in the primal form, $\mathbf{A}\succeq 0$ in the dual form) challenges the computational tractability of the SDP relaxation. To improve computational speed, a matrix completion theorem [26] enables the decomposition of this single positive semidefinite constraint into constraints on many smaller submatrices [22], [27]. Our method leverages this decomposition.

This section summarizes the matrix completion decomposition, beginning with several definitions from graph theory. A

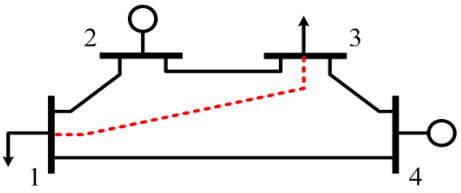


Fig. 1. Example of the one-line diagram of a four-bus power network (black lines) and a chordal extension (black lines plus the dashed red line).

"clique" is a set of nodes which are adjacent to all other nodes in the clique. A clique that is not contained in a larger clique is a "maximal clique". An edge that connects two non-adjacent vertices in a cycle is called a "chord". A graph is "chordal" if every cycle with the length of four or more nodes has a chord. We use $\mathcal{S}$ to denote the set of all maximal cliques and m to indicate the number of maximal cliques.

As a simple example, consider the one-line diagram of a four-bus system shown in Fig. 1. The network graph is not chordal since the cycle $\{1, 2, 3, 4\}$ does not have a chord. The graph formed by adding the dashed red line is chordal with two maximal cliques $\mathcal{C}_1 = \{1, 2, 3\}$ and $\mathcal{C}_2 = \{1, 3, 4\}$. Thus, the graph including the dashed red line is a "chordal extension". See [22] and [28] for further details on these definitions.

The matrix completion theorem in [26] considers a partially specified symmetric matrix with an associated undirected chordal graph. For our purposes, this graph is (a chordal extension of) the power system network and the unspecified matrix entries correspond to non-adjacent pairs of buses. The theorem states that the partially specified matrix can be "completed" to be positive semidefinite (i.e., there exist values for the unspecified entries such that the matrix is positive semidefinite) if and only if all the submatrices corresponding to the graph's maximal cliques are positive semidefinite.

For non-chordal power system networks, the matrix completion decomposition is applied by forming a chordal extension of the network graph. An appropriate chordal extension is constructed using the sparsity pattern of a Cholesky factorization of the network's adjacency matrix [22], [27]. A minimum degree ordering reduces the number of additional edges required in the chordal extension [29]. Further computational improvements are possible via heuristics for constructing alternative chordal extensions [23].

Using the matrix completion theorem, the positive semidefinite constraint is equivalently replaced by constraints on smaller submatrices associated with the maximal cliques of the chordal extension of the network graph. For the system in Fig. 1, this implies that the constraint $\mathbf{W} \succeq 0$ in (6) can be replaced by the constraints on the two submatrices whose rows and columns are associated with buses in the maximal cliques $\mathcal{C}_1$ and $\mathcal{C}_2$. Due to the possibility of non-empty intersections among maximal cliques, note that different submatrices may contain elements which refer to the same element in the original $2n \times 2n$ matrix $\mathbf{W}$. "Linking constraints" ensure consistency among variables contained in the submatrices associated with multiple cliques [22], [27].

As an illustrative example, again consider the chordal extension of the four-bus system in Fig. 1. The matrix completion theorem enables the decomposition of the 8×8 positive semidefinite matrix constraint (6) into two constraints on the 6×6 submatrices, denoted $\mathbf{W}^{(1)}$ and $\mathbf{W}^{(2)}$, that are associated with the cliques $\mathcal{C}_1$ and $\mathcal{C}_2$. For notational simplicity, we next show the upper-left 3×3 diagonal blocks and corresponding linking constraints for these submatrices. The remaining entries are treated analogously.

$$\mathbf{W}^{(1)} = \begin{bmatrix} w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} & \cdots \\ w_{12}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} & \cdots \\ w_{13}^{(1)} & w_{23}^{(1)} & w_{33}^{(1)} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \succeq 0, \qquad \mathbf{W}^{(2)} = \begin{bmatrix} w_{11}^{(2)} & w_{12}^{(2)} & w_{13}^{(2)} & \cdots \\ w_{12}^{(2)} & w_{22}^{(2)} & w_{23}^{(2)} & \cdots \\ w_{13}^{(2)} & w_{23}^{(2)} & w_{33}^{(2)} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \succeq 0$$

Since buses 1 and 3 are the first and third elements in $\mathcal{C}_1$ and the first and second elements in $\mathcal{C}_2$, the linking constraints are

$$w_{11}^{(1)} = w_{11}^{(2)}, \quad w_{13}^{(1)} = w_{12}^{(2)}, \quad w_{33}^{(1)} = w_{22}^{(2)}, \ \ldots$$

Applying this decomposition to the matrix $\mathbf{W}$ in (6) significantly improves the computational tractability of the SDP relaxation [22]. Taking the dual of the resulting SDP yields a chordal-sparsity-exploiting version of (8), which is the formulation used by our proposed method. We hereafter denote the submatrices in the sparsity-exploiting version of (8) as $\mathbf{A}_i$, $i = 1, \ldots, m$, each corresponding to a maximal clique contained in the set $\mathcal{S} = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$. Note that the sparsity-exploiting version of (8) includes dual variables in $\mathbf{A}_i$ that correspond to the linking constraints in the primal version. For further details, see [9], [22], [23].

III. Proposed Method

Despite the computational improvements from the matrix completion decomposition, sparsity-exploiting SDP relaxations are slower than many local solution algorithms for typical OPF problems. This section describes our proposed method for leveraging the speed of local solution algorithms to more quickly compute an objective value bound using the SDP relaxation. We present a high-level summary, detail each step, and discuss several features of the proposed method.

A. Description of the Proposed Method

Our method is based on the fact that the objective value associated with any dual feasible point for the SDP relaxation (8) bounds the objective of the primal SDP relaxation (7) and therefore also bounds the objective of the original OPF problem (5). Rather than perform the computationally expensive calculation of the best achievable bound via application of an SDP solver to (8), we instead use certain information from a local solution to form a simplified SDP problem. Applying an SDP solver to this simplified problem can quickly provide a dual feasible point whose corresponding objective value is a bound on the objective of the original OPF problem (5). The trade-off inherent to this computational improvement is potential suboptimality in the obtained bound.

To form this simplified problem, we fix certain dual variables in the SDP relaxation (8) to their corresponding values from a specified local solution, leaving only a subset of the dual variables to vary. The fixed dual variables are identified

Algorithm 1: Quickly Bounding the Objective Value

Input: Dual variables from a local OPF solution.
Output: A bound for the OPF's optimal objective value.

1 Obtain the matrix $\hat{\mathbf{A}}$ matrix by substituting the dual variable values from the local solution into (10).
2 Compute the matrix completion decomposition, i.e., the submatrices $\mathbf{A}_i$ for all $i \in \mathcal{S}$ using the approach in [22], [23]. Substitute in the values of the dual variables from the local solution to form $\hat{\mathbf{A}}_i$.
3 Identify the set of problematic submatrices, $\mathcal{S}_p$.
4 Create the sets of all buses, $\mathcal{N}_p$, and lines, $\mathcal{L}_p$, included in $\mathcal{S}_p$.
5 Formulate and solve the simplified SDP problem (12). Return the objective value as a bound on the original OPF problem (5).

using a heuristic that computes "problematic" submatrices of the matrix $\mathbf{A}$ in (8) evaluated using the dual variable values from the local solution. Consider the submatrices $\mathbf{A}_i$, $i = 1, \ldots, m$, resulting from the matrix completion decomposition. A submatrix is deemed "problematic" if it has any eigenvalues that are either negative or small positive numbers. Dual variables associated with "problematic" submatrices are allowed to vary, while the other dual variables in (8) are fixed to their values from the local solution.

Algorithm 1 summarizes our method. Each step is described in further detail below.

Input: Our method takes as input the values of the dual variables from a local solution to the OPF problem (5). These values can be obtained using any primal/dual local OPF solver.

Note that solutions to equivalent but differently formulated OPF problems can have different values for the dual variables. For instance, the OPF formulation in MATPOWER [7] limits the voltage magnitudes rather than the squared voltage magnitudes as in (5d). If the local solution is computed for a different OPF formulation than (5), use conversions such as those in [24] to obtain dual variables that are consistent with (5).

1) Evaluate the dual matrix: Substitute the values of the dual variables into the $\mathbf{A}$ matrix defined in (10). Denote the resulting matrix as $\hat{\mathbf{A}}$.

2) Construct the matrix completion decomposition: As summarized in Section II-C, use the approach in [22], [23] to compute the maximal cliques of a chordal extension of the network graph and form the corresponding submatrices of $\hat{\mathbf{A}}$, i.e., $\hat{\mathbf{A}}_1, \ldots, \hat{\mathbf{A}}_m$. When constructing $\hat{\mathbf{A}}_i$, the dual variables corresponding to the linking constraints are set to zero.

3) Identify problematic submatrices: Use the following heuristic to identify a set of "problematic" cliques, which are denoted $\mathcal{S}_p$. Let $|\cdot|$ indicate the cardinality of a set. Let $\mathrm{ceil}(\cdot)$ denote the ceiling function which returns the smallest integer greater than or equal to the argument. Let $\mathrm{eig}(\cdot)$ return the eigenvalues of a matrix. Let σ be a specified scalar parameter that dictates the minimum percentage of submatrices that are identified as "problematic".

First, identify as problematic all submatrices that are not positive semidefinite, i.e., if $\min(\mathrm{eig}(\hat{\mathbf{A}}_i)) < 0$, include i in $\mathcal{S}_p$. Then, if $|\mathcal{S}_p| < \sigma \cdot |\mathcal{S}|$, list all the submatrices $\mathbf{A}_i$ in increasing order of their smallest eigenvalues. Select the first $\mathrm{ceil}(\sigma \cdot |\mathcal{S}|)$ submatrices in this ordering and add them to $\mathcal{S}_p$. Thus, at least σ percent of the submatrices are identified as problematic. Note that $\sigma = 0$ does not necessarily imply that $\mathcal{S}_p$ is empty. Rather, $\sigma = 0$ indicates that we only identify the submatrices that are not positive semidefinite as problematic.

4) Identify the fixed and non-fixed dual variables: Identify the set $\mathcal{N}_p$ that contains all buses which are in problematic cliques $\mathcal{S}_p$. Similarly, let $\mathcal{L}_p$ denote the set of all lines for which either terminal bus is in a problematic clique $\mathcal{S}_p$. The dual variables associated with buses in $\mathcal{N}_p$ and lines in $\mathcal{L}_p$ are decision variables in the simplified SDP problem. Dual variables associated with the remaining buses and lines, i.e., $\mathcal{N} \setminus \mathcal{N}_p$ and $\mathcal{L} \setminus \mathcal{L}_p$, are fixed to their corresponding values from the local solution.

5) Formulate and solve the simplified SDP relaxation: Fixing the dual variables for buses in $\mathcal{N} \setminus \mathcal{N}_p$ and lines in $\mathcal{L} \setminus \mathcal{L}_p$ yields the following simplified version of (8):

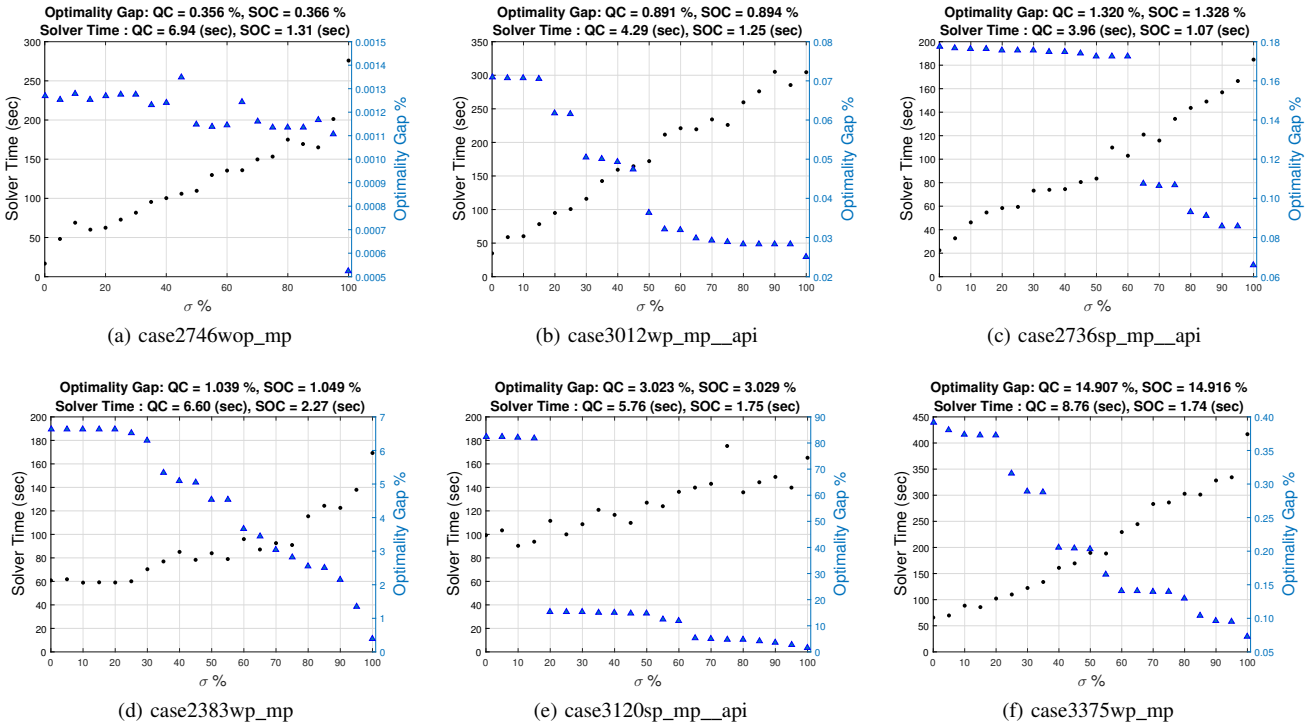
$$\max \quad \rho \tag{12a}$$
$$\text{subject to} \quad (\ \forall k \in \mathcal{N}_p,\ \forall (l,m) \in \mathcal{L}_p)$$
$$\tilde{\mathbf{A}}_i \succeq 0, \qquad \forall i = 1, \ldots, m \tag{12b}$$
$$\mathbf{H}_{lm} \succeq 0, \tag{12c}$$
$$\mathbf{R}_k \succeq 0, \quad \mathbf{R}_k^{11} = 1, \tag{12d}$$
$$\underline{\lambda}_k \geq 0,\ \overline{\lambda}_k \geq 0,\ \underline{\gamma}_k \geq 0,\ \overline{\gamma}_k \geq 0,\ \underline{\mu}_k \geq 0,\ \overline{\mu}_k \geq 0, \tag{12e}$$
$$\underline{\beta}_{lm} \geq 0,\ \overline{\beta}_{lm} \geq 0, \tag{12f}$$

where the submatrices in the sparsity-exploiting dual formulation are denoted as $\tilde{\mathbf{A}}_i$. These submatrices include constant terms which correspond to the values of the dual variables associated with the buses in $\mathcal{N} \setminus \mathcal{N}_p$ and lines in $\mathcal{L} \setminus \mathcal{L}_p$, i.e., the fixed dual variable values from the local solution. Furthermore, all matrices $\tilde{\mathbf{A}}_i$, $i = 1, \ldots, m$, are functions of the dual variables corresponding to the linking constraints, as discussed in Section II-C and [9], [22], [23]. The dual variables corresponding to the linking constraints are decision variables in (12) whose values are not provided by the local solution. Thus, (12b) considers the submatrices $\tilde{\mathbf{A}}_i$ corresponding to *all* maximal cliques, $\mathcal{S}$, rather than just those corresponding to the problematic cliques $\mathcal{S}_p$.

A solution to (12) is a dual feasible point for the (sparsity-exploiting) SDP relaxation (8). Thus, the objective value for the solution to (12) is a lower bound on the objective value of the original OPF problem (5).

B. Discussion

With many variables fixed to their corresponding values from the local solution, the SDP problem (12) is easier to solve than the original SDP relaxation (8), which results in a computational speed advantage for the proposed method. This advantage comes at the cost of reduced tightness since the dual problem has fewer degrees of freedom with which to improve the objective ρ. The value of σ controls this trade-off between the solution speed and bound tightness. The tightest lower bound occurs for $\sigma = 100\%$, which yields the (sparsity-exploiting) complete dual SDP relaxation (8), i.e., all dual variables are allowed to vary. Conversely, $\sigma = 0\%$ fixes the



(a) case2746wop_mp (b) case3012wp_mp__api (c) case2736sp_mp__api

(d) case2383wp_mp (e) case3120sp_mp__api (f) case3375wp_mp

Fig. 2. Solver Times (seconds) (●) and Optimality Gaps (%) (▲) vs. $\sigma(\%)$.

largest number of dual variables in (12) and thus yields the fastest computational speed at the cost of a weaker bound.

The proposed method performs well with small values of σ when the local solution is close to the SDP solution. To numerically evaluate this claim, we compute the two-norm $||\cdot||_2$ of the ratio between the dual variable values for the local solution and the SDP solution:

$$\text{Dual Correspondence Ratio } (\%) = \left|\left| \frac{X^{Dual}_{local} - X^{Dual}_{SDP}}{X^{Dual}_{SDP}} \right|\right|_2 \times 100, \tag{13}$$

where X denotes the vector of the dual variables defined in Section II-B. When the local solution is far from the SDP solution, fixing variables to the values from the local solution overly constrains (12), resulting in an inferior bound and potentially even infeasibility. Such cases can be addressed by increasing the value of σ.

The dual correspondence ratio in (13) provides an a-posteriori explanation of our method's performance. We also consider a heuristic for estimating the quality of our method's bound without prior knowledge of the dual SDP solution. Specifically, we empirically find that our method performs well when more than 95% of the submatrices $\hat{\mathbf{A}}_i$ obtained using the local solution's dual variables are positive semidefinite.

IV. Numerical Experiments

A key advantage of the proposed method is its ability to control the trade-off between computational speed and tightness of the objective value bound. This trade-off is determined by how well the dual variables from the local solution match the dual variables for the SDP relaxation's solution. The closeness of these solutions is problem dependent. Thus, evaluating the effectiveness of the proposed method requires numerical experimentation. After providing our implementation details and evaluation methodology, this section applies the proposed method to a variety of large test cases. The results indicate that the proposed method is a valuable "middle ground" in that it is often tighter but slower than the SOCP and QC relaxations while weaker but faster than the SDP relaxation.

A. Implementation Details and Evaluation Methodology

Algorithm 1 is implemented using MATLAB R2016a and YALMIP [30] and solved using MOSEK 8 on a computer with a 64-bit Intel Quad Core 3.40 Ghz CPU with 8 GB of RAM. The local solutions are obtained using the interior point method in MATPOWER [7]. The numerical results include comparisons to the SOCP relaxation [15], the QC relaxation [17] augmented with the "Lifted Nonlinear Cuts" from [31], and the sparsity-exploiting SDP relaxation [22], [23]. For our experiments, we consider the large test cases (≥ 118 buses) in the NESTA 0.6.0 archive [32].

We evaluate bound tightness via the *optimality gap*, which compares the objective value from the local solution to the bound from a dual feasible point obtained using the proposed method or the bound from a relaxation:

$$\text{Optimality Gap } (\%) = \frac{\text{Local Sol. Obj. – Obj. Value Bound}}{\text{Local Sol. Obj.}} \times 100.$$

B. Test Case Results

To illustrate the trade-off in computational speed and bound tightness, Fig. 2 shows how the solver times and optimality gaps vary with σ for selected test cases. For some cases, such as case3012wp_mp__api, case2736sp_mp__api, and case3375wp_mp, the solver times decrease by approximately a factor of nine to ten as σ varies from 100% (equivalent to the sparse version of the SDP relaxation (8)) to 0%

TABLE I
OPTIMALITY GAPS AND SOLVER TIMES FOR $\sigma = 20\%$

System Model	MATPOWER Obj. Value ($/h)	Optimality Gap (%)				Solver Time (sec)			
		Our Method	SDP	QC	SOCP	Our Method	SDP	QC	SOCP
Typical Operating Condition (TOC)									
case118_ieee	3718.64	0.862	0.064	1.568	2.069	0.33	0.65	0.19	0.11
case300_ieee	16891.28	0.148	0.078	1.175	1.179	0.86	2.80	0.56	0.18
case1354_pegase	74069.35	0.107	0.010	0.077	0.081	4.46	13.58	2.31	0.52
case2383wp_mp	1868511.83	6.643	0.375	1.039	1.049	58.48	181.81	6.60	2.27
case2736sp_mp	1307883.13	0.000	0.000	0.288	0.297	50.00	277.30	6.29	1.29
case2737sop_mp	777629.30	0.001	0.000	0.246	0.249	53.10	230.97	3.49	1.43
case2746wop_mp	1208279.81	0.001	0.001	0.356	0.366	55.45	318.05	6.94	1.31
case2746wp_mp	1631775.10	0.000	0.000	0.318	0.324	51.55	239.08	6.21	1.34
case2869_pegase	133999.29	0.085	0.008	0.088	0.091	13.02	41.13	4.52	1.82
case3012wp_mp	2600842.77	0.850	0.143	1.002	1.021	74.01	362.15	6.97	1.27
case3120sp_mp	2145739.43	0.153	0.088	0.536	0.545	95.78	435.99	6.52	1.34
case3375wp_mp	7435697.54	0.372	0.073	14.907	14.916	103.92	449.06	8.76	1.74
Congested Operating Conditions (API)									
case118_ieee__api	10325.27	37.888	31.502	43.650	44.082	0.62	0.84	0.35	0.14
case300_ieee__api	22866.01	0.004	0.003	0.818	0.837	0.68	1.97	0.66	0.23
case2383wp_mp__api	23499.48	0.547	0.101	1.118	1.120	97.02	194.25	3.80	1.15
case2736sp_mp__api	25437.70	0.175	0.069	1.320	1.328	55.25	191.73	3.96	1.07
case2737sop_mp__api	21192.40	0.318	0.007	1.049	1.056	41.77	196.74	3.94	1.13
case2746wop_mp__api	22814.86	0.000	0.000	0.488	0.491	65.85	282.61	4.34	1.40
case2746wp_mp__api	27291.58	0.000	0.000	0.569	0.491	53.99	199.81	4.02	1.15
case3012wp_mp__api	27917.36	0.061	0.025	0.891	0.894	77.84	266.51	4.29	1.25
case3120sp_mp__api	22874.98	15.416	0.542	3.023	3.029	104.09	458.98	5.76	1.75
case3375wp_mp__api	48898.95	0.035	0.014	45.765	45.766	89.74	284.44	5.04	1.63
Small Angle Difference Conditions (SAD)									
case118_ieee__sad	4324.17	10.687	7.652	8.286	15.783	0.68	0.93	0.43	0.17
case2383wp_mp__sad	1935308.26	22.467	1.259	2.967	4.464	82.19	224.38	6.66	1.73
case2736sp_mp__sad	1337042.83	38.915	0.635	2.009	2.472	66.92	314.64	4.39	1.18
case2746wop_mp__sad	1241955.37	20.126	0.999	2.482	3.067	78.31	458.02	5.25	1.13
case3012wp_mp__sad	2635451.43	15.537	0.464	1.916	2.320	103.00	439.56	5.19	1.34

(only submatrices that are not positive semidefinite are identified as problematic). Varying σ has a much smaller impact on the solver times for other cases, such as case2746wop_mp, case2383wp_mp, and case3120sp_mp__api.

The improvement in speed with smaller σ has a trade-off with respect to the tightness of the objective value bound. As with the solver times, the impact of σ on the optimality gap is case dependent. For case2746wop_mp, case3012wp_mp__api, and case2736sp_mp__api, the optimality gaps decrease by approximately a factor of two to three as σ varies from 0% to 100%, while for case3375wp_mp, case2383wp_mp, and case3120sp_mp__api, the optimality gaps decrease by factors of 8, 20 and 200, respectively, as σ varies from 0% to 100%. These variations are explained by the quality of the local solution. When the local solution is close to the SDP solution, increasing σ only has a small impact on the bound.

In contrast to the solver times, which vary gradually with σ, the optimality gaps exhibit more significant variations. For example, the optimality gap for case2736sp_mp__api does not change significantly from $\sigma = 0\%$ until a steep decrease at $\sigma = 60\%$. This is explained by a group of buses whose dual variable values from the local solution are far from their values in the SDP solution. Choosing $\sigma < 60\%$ fixes these dual variables to the values from the local solution, restricting our method's ability to achieve a tight bound. Jumps in the optimality gaps for other test cases have similar explanations.

Table I shows results for representative test cases, specifically comparing the proposed method using $\sigma = 20\%$ with the SDP, QC, and SOCP relaxations. Three categories of NESTA test cases are considered: typical operating conditions (TOC), congested operation (API), and small angle difference (SAD).

Table II presents the average optimality gaps and solver times for each category of test cases. Our method performs well for most of the TOC and API test cases, as indicated by smaller optimality gaps than the QC and SOCP relaxations with a significant (approximately factor of five) improvement in computation speed relative to the SDP relaxation. The average results show that the proposed method yields better optimality gaps than QC and SOCP relaxations, while being significantly faster than the SDP relaxation, suggesting the method's utility as a middle ground between the the SDP relaxation and the SOCP and QC relaxations.

TABLE II
AVERAGE OPTIMALITY GAPS AND SOLVER TIMES

Approach	Average Optimality Gaps (%)			Average Solver Times (sec)		
	TOC	API	SAD	TOC	API	SAD
Our Method	0.77	1.84	21.55	46.9	65.6	66.2
SDP	0.07	0.08	2.20	212.7	230.8	287.5
QC	1.80	6.12	3.53	5.0	4.0	4.4
SOCP	1.85	6.11	5.62	1.2	1.2	1.1

However, despite promising performance on many problems, there are test cases which challenge our method. For instance, our method yields large gaps relative to the SOCP, QC, and SDP relaxations for the SAD test cases, case2383wp_mp, and case3120sp_mp__api. For these cases, the local solution is not close to the SDP relaxation's solution such that large values of σ (with correspondingly slow solution times) are necessary to achieve small optimality gaps. Explaining our method's poor performance for these test cases, the dual correspondence

TABLE III
DUAL CORRESPONDENCE RATIOS AND PERCENTAGES OF POSITIVE SEMIDEFINITE SUBMATRICES FOR SELECTED TEST CASES

System Model	Dual Correspondence Ratio (%)	Percentage of PSD Submatrices	Optimality Gaps (%)
case2736sp_mp	0.16	99.3	0.000
case2746wop_mp	0.45	98.4	0.001
case3375wp_mp__api	0.93	97.5	0.035
case3012wp_mp	1.32	97.3	0.850
case2869_pegase	4.32	96.7	0.085
case3375wp_mp	7.00	95.7	0.372
case2383wp_mp	28.91	78.0	6.643
case3120sp_mp__api	462.70	92.0	15.416
case2383wp_mp__sad	105.10	58.4	22.467
case2736sp_mp__sad	61.30	89.8	38.915
case3012wp_mp__sad	104.31	86.2	15.537

ratios (13) presented in Table III have smaller values for the test cases where our method performs poorly relative to the test cases with tighter optimality gaps.

Finally, we demonstrate the a-priori heuristic proposed in Section III-B via selected test cases in Table III. Our method performs well for the test cases where the percentages of positive semidefinite submatrices are greater than 95%. For these cases, our method yields optimality gaps between those from the SDP relaxation and the SOCP and QC relaxations with significant speed improvements compared to the SDP relaxation. Conversely, our method often performs poorly for test cases that do not satisfy this heuristic, such as case2383wp_mp, case3120sp_mp__api, and the SAD cases. This shows the appropriateness of our heuristic recommendation for applying our method to problems where the percentage of submatrices $\hat{\mathbf{A}}_i$ that are positive semidefinite is greater than 95%.

V. CONCLUSIONS

This paper has proposed an SDP-based method that leverages knowledge of a local solution to quickly compute objective value bounds for optimal power flow problems. Numerical experiments indicate that the proposed method is approximately four to five times faster than the SDP relaxation for a variety of test cases without excessive degradation in the quality of the objective bound. These experiments also show that the proposed method is slower than the SOCP and QC relaxations but can obtain tighter bounds. Thus, the proposed method provides a middle ground between the SDP relaxation and the SOCP and QC relaxations with a trade-off between computational speed and tightness.

REFERENCES

[1] J. Carpentier, "Contribution to the Economic Dispatch Problem," *Bull. Soc. Franc. Elect*, vol. 8, no. 3, pp. 431–447, 1962.

[2] D. Bienstock and A. Verma, "Strong NP-Hardness of AC Power Flows Feasibility," *arXiv:1512.07315*, 2015.

[3] K. Lehmann, A. Grastien, and P. Van Hentenryck, "AC-Feasibility on Tree Networks is NP-Hard," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 798–801, Jan. 2016.

[4] W. A. Bukhsh, A. Grothey, K. McKinnon, and P. A. Trodden, "Local Solutions of the Optimal Power Flow Problem," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4780–4788, 2013.

[5] J. Momoh, R. Adapa, and M. El-Hawary, "A Review of Selected Optimal Power Flow Literature to 1993. Parts I and II," *IEEE Trans. Power Syst.*, vol. 14, no. 1, pp. 96–111, Feb. 1999.

[6] A. Castillo and R. P. O'Neill, "Survey of Approaches to Solving the ACOPF (OPF Paper 4)," US Federal Energy Regulatory Commission, Tech. Rep., Mar. 2013.

[7] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, 2011.

[8] D. K. Molzahn and L. A. Roald, "Towards an AC Optimal Power Flow Algorithm with Robust Feasibility Guarantees," *20th Power Syst. Comput. Conf. (PSCC)*, June 2018.

[9] D. K. Molzahn and I. A. Hiskens, "A Survey of Relaxations and Approximations of the Power Flow Equations," invited submission to *Found. Trends Electric Energy Syst.*, 2018.

[10] A. Gopalakrishnan, A. Raghunathan, D. Nikovski, and L. T. Biegler, "Global Optimization of Optimal Power Flow using a Branch & Bound Algorithm," in *50th Annu. Allerton Conf. Commun., Control, Comput.*, Oct. 2012, pp. 609–616.

[11] D. Phan, "Lagrangian Duality and Branch-and-Bound Algorithms for Optimal Power Flow," *Oper. Res.*, vol. 60, no. 2, pp. 275–285, 2012.

[12] M. Lu, H. Nagarajan, R. Bent, S.D. Eksioglu, and S. J. Mason, "Tight Piecewise Convex Relaxation for Global Optimization of Optimal Power Flow," *20th Power Syst. Comput. Conf. (PSCC)*, June 2018.

[13] C. Coffrin, H. L. Hijazi, and P. Van Hentenryck, "Network Flow and Copper Plate Relaxations for AC Transmission Systems," in *19th Power Syst. Comput. Conf. (PSCC)*, June 2016.

[14] R. A. Jabr, "Radial Distribution Load Flow using Conic Programming," *IEEE Trans. Power Syst.*, vol. 21, no. 3, pp. 1458–1459, 2006.

[15] S. H. Low, "Convex Relaxation of Optimal Power Flow–Part I: Formulations and Equivalence," *IEEE Trans. Control Network Syst.*, vol. 1, no. 1, pp. 15–27, Mar. 2014.

[16] B. Kocuk, S. S. Dey, and X. A. Sun, "Strong SOCP Relaxations of the Optimal Power Flow Problem," *Oper. Res.*, vol. 64, no. 6, pp. 1177–1196, 2016.

[17] C. Coffrin, H. L. Hijazi, and P. Van Hentenryck, "The QC Relaxation: A Theoretical and Computational Study on Optimal Power Flow," *IEEE Trans. Power Syst.*, vol. 31, no. 4, pp. 3008–3018, 2016.

[18] J. Lavaei and S. H. Low, "Zero Duality Gap in Optimal Power Flow Problem," *IEEE Trans. Power Syst.*, vol. 27, no. 1, pp. 92–107, 2012.

[19] C. Josz and D. K. Molzahn, "Lasserre Hierarchy for Large Scale Polynomial Optimization in Real and Complex Variables," *SIAM J. Optimiz.*, vol. 28, no. 2, pp. 1017–1048, 2018.

[20] N. Z. Shor, "Quadratic Optimization Problems," *Sov. J. Comput. Syst. Sci.*, vol. 25, pp. 1–11, 1987.

[21] B. C. Lesieutre, D. K. Molzahn, A. R. Borden, and C. L. DeMarco, "Examining the Limits of the Application of Semidefinite Programming to Power Flow Problems," in *49th Annu. Allerton Conf. Commun., Control, Comput.*, Sept. 2011.

[22] R. A. Jabr, "Exploiting Sparsity in SDP Relaxations of the OPF Problem," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 1138–1139, 2012.

[23] D. K. Molzahn, J. T. Holzer, B. C. Lesieutre, and C. L. DeMarco, "Implementation of a Large-Scale Optimal Power Flow Solver based on Semidefinite Programming," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 3987–3998, 2013.

[24] D. K. Molzahn, B. C. Lesieutre, and C. L. DeMarco, "A Sufficient Condition for Global Optimality of Solutions to the Optimal Power Flow Problem," *IEEE Trans. Power Syst.*, vol. 29, no. 2, pp. 978–979, 2014.

[25] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ. Press, 2009.

[26] R. Gron, C. R. Johnson, E. M. Sá, and H. Wolkowicz, "Positive Definite Completions of Partial Hermitian Matrices," *Linear Algebra Appl.*, vol. 58, pp. 109–124, 1984.

[27] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, "Exploiting Sparsity in Semidefinite Programming via Matrix Completion I: General Framework," *SIAM J. Optimiz.*, vol. 11, no. 3, pp. 647–674, 2001.

[28] G. Valiente, *Algorithms on Trees and Graphs*. Springer, 2013.

[29] P. R. Amestoy, T. A. Davis, and I. S. Duff, "Algorithm 837: AMD, An Approximate Minimum Degree Ordering Algorithm," *ACM Trans. Math. Softw.*, vol. 30, no. 3, pp. 381–388, 2004.

[30] J. Lofberg, "YALMIP: A Toolbox for Modeling and Optimization in MATLAB," in *IEEE Int. Symp. Compu. Aided Control Syst. Des.*, 2004, pp. 284–289.

[31] C. Coffrin, H. L. Hijazi, and P. Van Hentenryck, "Strengthening the SDP Relaxation of AC Power Flows with Convex Envelopes, Bound Tightening, and Valid Inequalities," *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 3549–3558, Sept. 2017.

[32] C. Coffrin, D. Gordon, and P. Scott, "NESTA, The NICTA Energy System Test Case Archive," *arXiv:1411.0359*, August 2016.