Matrix Completion Embedded PDIP Method for SDP Relaxation of Large-scale OPF Problems

Journal:	*IEEE Transactions on Power Systems*
Manuscript ID	TPWRS-00905-2018
Manuscript Type:	Transactions
Date Submitted by the Author:	13-Jun-2018
Complete List of Authors:	Li, Rui; Tsinghua university, Electrical Engineering Mei, Shengwei; Tsinghua University, Electrical Engineering Liu, Feng; Tsinghua Univeristy, Electrical Engineering Wei, Wei; Tsinghua University, Department of Electrical Engineering Li, Na; Harvard University John A Paulson School of Engineering and Applied Sciences
Technical Topic Area :	Power system operations, Power system computational analysis
Key Words:	optimal power flow, semidefinite relaxation, primal-dual interior-point method, matrix completion, chordal graph

Matrix Completion Embedded PDIP Method for SDP Relaxation of Large-scale OPF Problems

Rui Li , *Student Member, IEEE,* Shengwei Mei, *Fellow, IEEE,* Feng Liu, *Member, IEEE*
Wei Wei, *Member, IEEE,* Na Li, *Member, IEEE*

Abstract—**Optimal power flow (OPF) is the basis of power system operation. Practical OPF problems are challengeable because of non-convexity. Semidefinite programming (SDP) relaxation is a valid approach to convex the non-convex OPF problem by associating a positive semidefinite (PSD) matrix constraint. Nevertheless, expensive matrix computations concerning high-dimension matrix in primal-dual interior-point (PDIP) algorithm prohibit the application of SDP relaxation for large-scale power systems. State-of-the-art approaches decompose the high-dimension PSD matrix constraint into a family of low-dimension PSD constraints associated with substantial consistency constraints to solve practical OPF problems. In this paper, we propose a matrix completion embedded PDIP method by integrating a sparse clique-factorization and matrix completion into PDIP algorithm. The proposed method avoids the consistency constraints and more flexible to be performed in a parallel fashion. Numerical experiments demonstrated that the proposed method outperforms PDIP method regarding computation efficiency and memory consumption in SDP relaxation of large-scale OPF problems. Besides, the proposed method features same exactness of traditional methods.**

Index Terms—**Optimal power flow, semidefinite relaxation, primal-dual interior-point method, matrix completion.**

I. Introduction

OPTIMAL power flow (OPF) is the foundation of numerous power system optimization applications in control, operation, markets, and planning. OPF and its variants, such as contingency-constrained OPF [1] and multi-period OPF [2], [3] are becoming increasingly crucial for modern power systems with the mushrooming of renewable generation plants and energy storage facilities. Various optimization algorithms, including nonlinear optimization methods, evolutionary algorithms, and convex relaxation methods such as second-order cone [4], [5], semidefinite programming (SDP) [6], [7], quadratic convex [8], [9], and moment-based relaxation [10] have been investigated to find OPF solutions. The convex relaxation techniques could offer either a lower bound on the optimal value, a good starting point, a certificate of global optimality, or a certificate of infeasibility. Among many convex relaxation methods, SDP features a small relaxation gap.

Numerous efforts have been spent investigating the formulation, exactness, and algorithms for the SDP relaxation of OPF problems. Real rectangular power-voltage formulation [6], [11] and the complex formulation [12] are used to develop the SDP relaxation of OPF problems for transmission networks [13], distribution networks [14], micro-grids [15], and power systems with energy storage [16]. The exactness of SDP relaxation is given for resistive networks in [7] with the over-satisfaction condition, and in [17] and references therein. To solve the SDP effectively, primal-dual interior-point (PDIP) methods have been studied intensively and extensively [18], [19]. PDIP methods can handle efficiently small- and medium-sized SDPs for the OPF problems. Nevertheless, once the problem size is large, PDIP algorithms become extremely time-consuming. For instance, denote m and n as the number of equality constraints and the size of the PSD (positive semi-definite) matrix variable of the SDP problem. Then the computation overhead of each PDIP iteration related to high-dimension matrices is i) $O(m^3)$ for Cholesky factorization of the Schur complement system in the search direction step, and ii) $O(n^3)$ for the minimum eigenvalue calculation in the step length step. Thus, PDIP fails to solve real-world power systems where m and n reach tens of thousands.

To handle the expensive matrix computations associated with large-scale SDP problems, methods have been proposed to leverage the knowledge from chordal graph theory [20], [21] which exploits the sparsity of the problem [22]–[25]. The original SDP with a large-scale PSD matrix constraint can be converted into an equivalent SDP with a family of smaller size PSD submatrices constraints with the conversion method [22], [23]. This *conversion method* can be regarded as a pre-processing procedure for prevalent small- and medium-sized SDP solvers, which incorporate block-diagonal data structures to exploit sparsity. Motivated by the conversion method, chordal decomposition, and matrix completion techniques are regularly implemented to exploit the sparsity and the low-rank structures of the SDP relaxation of large-scale OPF problems [11], [13], [26], [27]. With the conversion method, one can either directly call the general-purpose SDP solvers as in [11], [13], [26], [27], or perform first-order methods [28], [29] or distributed PDIP algorithms [30] to solve the problem.

The PSD sub-matrix constraints in the conversion method incorporate overlapping elements, and additional equality constraints are necessary to enforce the consistency between the overlapping elements among the decomposed small-size sub-matrices [22], [23]. The consistency constraints substantially increase the order of the Schur complement system m and bring extra computation overheads. The number and the sizes

This work was partly supported by National Natural Science Foundation of China (51621065, U1766203) and NSF CAREER (1553407), and was done during R. Li's visit, supervised by N. Li, in Harvard John A. Paulson School of Engineering and Applied Science. (Corresponding author: Na Li)

R. Li, S. Mei, F. Liu, and W. Wei are with the State Key Laboratory of Power Systems, Department of Electrical Engineering, Tsinghua University (Email:eeairicky@gmail.com, meishengwei@mail.tsinghua.edu.cn).

N. Li is with Harvard John A. Paulson School of Engineering and Applied Science, 33 Oxford St., Cambridge, MA 02138, USA (E-mail:nali@seas.harvard.edu).

of sub-matrices should be balanced to reach a computationally efficient SDP relaxation of OPF problems. Heuristic methods for sub-matrix augmentation or combination [13], [22], [23] and consistency constraints dropping methods [12] could realize such tradeoff. These approaches partly remove the drawback of the conversion method, but they are not easily parallelizable because they rely on off-the-shelf SDP solvers.

Inspired by the completion method in [22], [23], we propose a matrix completion embedded PDIP method (MC-PDIP) for the SDP relaxation of large-scale OPF problems. The MC-PDIP method perfroms the matrix completion at each iteration to obtain a sparse clique-factorization, which is utilized in the compuation of the search direction in a cheaper fashion. The determination of the step length is enforced for each maximal clique instead of the whole matrix. With the MC-PDIP method, we entirely drop the consistency constraints necessitated in the conversion method to save the computation and memory overheads. Moreover, the search direction and the step length calculation step at each iteration of the MC-PDIP method can be performed in a parallel fashion, giving promises to solve multi-period OPF and contingency-constrained OPF problems which are even in a larger scale than the standard OPF. Furthermore, we provide a comprehensive performance comparison between the conversion methods proposed in the OPF literature and our MC-PDIP method for large-scale OPF problems. Besides, we discussed the exactness of the MC-PDIP method. To the best of our knowledge, this is the first study of investigating the use of MC-PDIP in OPF problems. Our results shows that MC-PDIP is an alternative efficient approach for solving large-scale OPF problems with thousands of buses.

The rest of this paper is organized as follows. The SDP relaxation of OPF problem is reformulated in Section II. Chordal graph and matrix completion theory are presented in Section III. The proposed MC-PDIP method for solving large-scale OPF with SDP relaxations are elaborated in Section IV. Numerical experiments are conducted in Section V, followed by conclusions.

II. Problem Statement

A. OPF Problem with SDP Relaxations

Consider a power system with N buses and L transmission lines, the OPF can be formulated as,

$$\begin{array}{l} \min\limits_{V,p^g,q^g} \ \sum_k \left(c_k^2(p_k^g)^2 + c_k^1 p_j^g + c_k^0\right) \\ s.t. \left\{ \begin{array}{l} p_k^{g,\min} \leq p_k^g \leq p_k^{g,\max}, \forall k \\ q_k^{g,\min} \leq q_k^g \leq q_k^{g,\max}, \forall k \\ V_k^{\min} \leq |V_k| \leq V_k^{\max}, \forall k \\ V_k I_k^* = \left(p_k^g - p_k^d\right) + \left(q_k^g - q_k^d\right) i, \forall k \\ S_l^f = V_k^f (I_k^f)^*, S_l^t = V_k^t (I_k^t)^*, \forall l \\ \left|S_l^f\right| \leq S_l^{\max}, |S_l^t| \leq S_l^{\max}, \forall l \end{array} \right. \end{array} \tag{1}$$

where $k = 1, 2, \cdots, N$, and $l = 1, 2, \cdots, L$. c_k^2, c_k^1, c_k^0 are non-negative cost coefficients, p_k^g, q_k^g are the active and reactive power produced by generator k, and p_k^d, q_k^d are active and reactive load. V_k, I_k are the complex-valued nodal voltage and injected current while V_k^f, I_k^f and V_k^t, I_k^t are associated with the head bus and tail bus of a transmission line l. S_l^f, S_l^t are transported complex-valued at the head side and tail side of transmission line l, respectively.

Since the nodal power balance equation is non-convex, OPF (1) is non-convex. By setting $U = [\mathrm{Re}[V]^T, \mathrm{Im}[V]^T]$, $a_k = c_k^0 + c_k^1 p_k^d$, $b_k = \sqrt{c_k^2} p_k^d$, we have,

$$\left\{ \begin{array}{l} \mathrm{Re}\left\{V_k I_k^*\right\} = U^T Y_k U = Y_k \bullet UU^T \\ \mathrm{Im}\left\{V_k I_k^*\right\} = U^T \bar{Y}_k U = \bar{Y}_k \bullet UU^T \\ |V_k|^2 = U^T M_k U = M_k \bullet UU^T \end{array} \right. \tag{2}$$

where $y_k = e_k e_k^T Y$, e_k is the k-th basis vector in $\mathbb{R}^N$, Y is the admittance matrix, $M_k = diag\{e_k e_k^T, e_k e_k^T\} \in \mathbb{R}^{2N \times 2N}$, "$\bullet$" represents the trace operation or inner product, and

$$Y_k = \frac{1}{2} \left[\begin{array}{cc} \mathrm{Re}\left[y_k + y_k^T\right] & \mathrm{Im}\left[y_k^T - y_k\right] \\ \mathrm{Im}\left[y_k - y_k^T\right] & \mathrm{Re}\left[y_k + y_k^T\right] \end{array} \right] \tag{3}$$

$$\bar{Y}_k = -\frac{1}{2} \left[\begin{array}{cc} \mathrm{Im}\left[y_k + y_k^T\right] & \mathrm{Re}\left[y_k - y_k^T\right] \\ Re\left[y_k^T - y_k\right] & Im\left[y_k + y_k^T\right] \end{array} \right] \tag{4}$$

Then, OPF (1) can be relaxed with SDP by introducing a matrix variable X and replacing $X = UU^T$ with $X \succeq 0$,

$$\begin{array}{l} \min\limits_{\beta,X} \ \sum_k \beta_k \\ s.t. \left\{ \begin{array}{l} p_k^{g,\min} - p_k^d \leq Y_k \bullet X \leq p_k^{g,\max} - p_k^d, \forall k \\ q_k^{g,\min} - q_k^d \leq \bar{Y}_k \bullet X \leq q_k^{g,\max} - q_k^d, \forall k \\ \left(V_k^{\min}\right)^2 \leq M_k \bullet X \leq V_k^{\max}, \forall k \\ \left[\begin{array}{ccc} -(S_l^{\max})^2 & Z_l^f \bullet X & \bar{Z}_l^f \bullet X \\ Z_l^f \bullet X & -1 & 0 \\ \bar{Z}_l^f \bullet X & 0 & -1 \end{array} \right] \preceq 0, \forall l \\ \left[\begin{array}{ccc} -(S_l^{\max})^2 & Z_l^t \bullet X & \bar{Z}_l^t \bullet X \\ Z_l^t \bullet X & -1 & 0 \\ \bar{Z}_l^t \bullet X & 0 & -1 \end{array} \right] \preceq 0, \forall l \\ \left[\begin{array}{cc} c_k^1 Y_k \bullet X + a_k - \beta_k & \sqrt{c_k^2} Y_k \bullet X + b_k \\ \sqrt{c_k^2} Y_k \bullet X + b_k & -1 \end{array} \right] \preceq 0, \forall k \\ X \succeq 0 \end{array} \right. \end{array} \tag{5}$$

where $Z_l^f, Z_l^t, \bar{Z}_l^f, \bar{Z}_l^t \in \mathbb{R}^{2N \times 2N}$ are constant matrixes concerning transmission parameters as the definitions in [13].

OPF with SDP relaxation (5) can be solved by common SDP solvers, such as Mosek, SeDuMi, and SDPT3. Once getting the optimal solution X^* of (5), we need to check its rank. If it is rank-1, the SDP (5) is exactness for OPF (1), and then we can recover a global optimal solution of (1). Nevertheless, the SDP relaxation is not exact in general, and the non-rank-1 solution X^* corresponds to no physically meaningful power flow status. Thus, we need perform other computations based on the X^*, such as the penalized SDP [31], to obtain practical solutions. In this paper, we primarily focus on developing efficient algorithms for large-scale OPF problems with SDP relaxations.

In general-purpose SDP solvers, inequality constraints are cast as equalities via slack variables. For instance, for the problem (5), active power limits of generators can be posed as equality constraints as,

$$\left\{ \begin{array}{l} p_k^g = p_k^{g,\min} + p_k^l \\ p_k^l + p_k^u = p_k^{g,\max} - p_k^{g,\min} \\ Y_k \bullet X = \left(p_k^{g,\min} + p_k^l\right) - p_k^d \end{array} \right. \tag{6}$$

where p_k^l, p_k^u are non-negative slack variables. Other constraints can be handled in similar fashions to derive an equality form for common SDP solvers.

B. Primal-Dual Interior-Point Method for SDP

Consider an SDP in the canonical form,

$$\begin{array}{ll} \min & A_0 \bullet X \\ s.t. & \begin{cases} A_p \bullet X = b_p \, (p = 1, 2, \cdots, m) \\ X \succeq 0 \end{cases} \end{array} \tag{7}$$

where X is the primal dense PSD matrix variable, $A_p(p = 1, 2, \cdots, m)$ is the sparse coefficient matrix, and m is the number of equality constraints. For SDP relaxation (5) of OPF problems, $n = 2N$ and m depends on the bus number N and line number L. Due to the sparsity of power networks, m grows linearly in n for OPF problems [1], [11], [15].

The dual form of SDP (7) reads as,

$$\begin{array}{ll} \max & \sum\limits_{p=1}^{m} b_p z_p \\ s.t. & \begin{cases} \sum\limits_{p=1}^{m} A_p z_p + W = A_0 \\ W \succeq 0 \end{cases} \end{array} \tag{8}$$

where z_p is the dual scalar variable, W is the dual PSD matrix variable, which inherits the sparsity pattern of data matrix A_p. Later, we use $\mathbb{S}^n$, $\mathbb{S}^n_+$, and $\mathbb{S}^n_{++}$ to represent symmetric, PSD, and positive definite matrix with n-dimension, respectively.

PDIP method is the stellar approach to solve SDP in polynomial time with a theoretically worst-case iteration bound $O(\sqrt{n}\log(1/\varepsilon))$ [18], here ε is the tolerance error. The algorithmic framework of a typical PDIP method [21], [22], [32] is illustrated in Algorithm 1, and consists of two basic procedures at each iteration, i.e., the search direction computation *(step 2)* and the step length calculation (*step 3*). Both of them incorporate matrix computations including either multiplication (*step 2.1, step 2.2, etc*), Cholesky factorization (*step 2.3*), or minimum eigenvalue calculation (*step 3*). Since the computation complexity of such matrix manipulation is $O(n^3)$ or $O(m^3)$, the computational overheads are overwhelming and even impractical for large-scale power system where n and m reach tens of thousands.

Motivated by the observations that most elements of X have no contribution to the primal objective value, $A_0 \bullet X$, and only a portion of them have effects on the PSD of X, matrix completion technique is adopted to convert the SDP (7) (and SDP (5)) with a large-scale PSD matrix variable to an equivalent SDP with a family of small-scale PSD submatrices, and then recover a full PSD matrix from the small-size submatrices. Another attractive feature of this transformation is that the converted SDP can be rearranged as a block-diagonal data structure [22], which sparsity is widely exploited in many general-purpose SDP solvers. To implement such transformation, the matrix completion and chordal graph theory are required and will be elaborated in Section III.

Algorithm 1 Standard PDIP Method for SDP

```
1: Initialization: choose initial point $(X^{(0)}, W^{(0)}, z^{(0)}) \in \mathbb{S}^n_{++} \times \mathbb{S}^n_{++} \times \mathbb{R}^m$, step reduction factor $\delta \in (0,1)$, set iteration index $k = 0$, and determine termination criterions.
2: Step 1: Convergence checking: Set complementarity measure $\mu^{(k)} = X^{(k)} \bullet W^{(k)}/n$, if current iterate $(X^{(k)}, W^{(k)}, z^{(k)})$ satisfies termination criterions, stop the algorithm.
3: Step 2: Compute search direction
   $(\Delta X^{(k)}, \Delta W^{(k)}, \Delta z^{(k)}) \in \mathbb{S}^n \times \mathbb{S}^n \times \mathbb{R}^m$,
   2.1 compute primal- and dual-residual,
       $\begin{cases} r_p = b_p - A_p \bullet X^{(k)} \\ R = A_0 - \sum_{p=1}^{m} A_p z_p - W^{(k)} \\ C = \mu^{(k)} I - X^{(k)} W^{(k)} \end{cases}$   (9)
   2.2 calculate coefficient of Schur complement system
       $\begin{cases} B_{pq} = A_p \bullet X^{(k)} A_q (W^{(k)})^{-1} \\ s_p = r_p - A_p \bullet \left(C - X^{(k)} R\right) (W^{(k)})^{-1} \end{cases}$   (10)
   2.3 solve Schur complement system $B \Delta z^{(k)} = s$.
   2.4 determine other directions,
       $\begin{cases} \Delta W^{(k)} = R - \sum\limits_{p=1}^{m} A_p \Delta z_p \\ \Delta \tilde{X} = \left(C - X^{(k)} \Delta W^{(k)}\right) (W^{(k)})^{-1} \\ \Delta X^{(k)} = 0.5(\Delta \tilde{X} + \Delta \tilde{X}^T) \end{cases}$   (11)
4: Step 3: Calculate step length $\alpha_p, \alpha_d$,
       $\begin{cases} \alpha_p = -1/\lambda_{\min}(\sqrt{X}^{(k)^{-1}} \Delta X^{(k)} \sqrt{X}^{(k)^{-T}}) \\ \alpha_d = -1/\lambda_{\min}(\sqrt{W}^{(k)^{-1}} \Delta W^{(k)} \sqrt{W}^{(k)^{-T}}) \end{cases}$   (12)
5: Step 4: Next iteration point
   $(X^{(k+1)}, W^{(k+1)}, z^{(k+1)}) \in \mathbb{S}^n_{++} \times \mathbb{S}^n_{++} \times \mathbb{R}^m$, i.e.,
       $\begin{cases} X^{(k+1)} \leftarrow X^{(k)} + \delta \alpha_p \Delta X^{(k)} \\ W^{(k+1)} \leftarrow W^{(k)} + \delta \alpha_d \Delta W^{(k)} \\ z^{(k+1)} \leftarrow z^{(k)} + \delta \alpha_d \Delta z^{(k)} \end{cases}$   (13)
6: Step 5 Set $k \leftarrow k + 1$, go to step 1.
7: return $(X^*, W^*, z^*)$.
```

III. Chordal Graph and Matrix Completion

Exploiting the sparsity of SDP relaxation of large-scale OPF problems could significantly improve the computational efficiency. Chordal graph and matrix completion theory benefit the sparsity exploitation through decomposing the high-dimension PSD constraint $X \succeq 0$ into a family of lower-dimension PSD constraints.

A. Chordal Graph and Chordal Extension

Denote the aggregate sparsity pattern E of SDP (7) as,

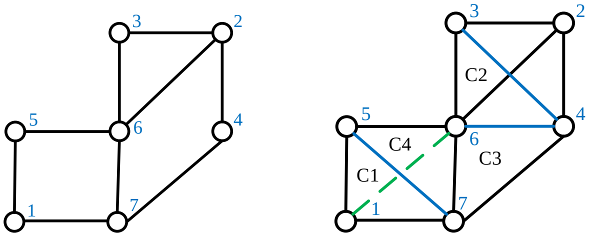
$$E = \{(i,j) \in V \times V | i \neq j, [A_p]_{ij} \neq 0, \exists p \in \{0, 1, \cdots, m\}\}.$$

and the sparse pattern of coefficient matrix A_p comes from the topology sparsity of practical power networks. We associate an undirect graph $G(V, E)$ having the vertex set V and the edge set $E \subseteq V \times V$ with the SDP relaxed OPF problem (7). Then, an edge joining two non-adjacent vertex on a circle of $G(V, E)$ is a *chord*, and the graph $G(V, E)$ is a *chordal graph* if all circles of length greater than or equal to 4 have a chord.

Moreover, a subset C_r of vertex set V forms a *clique* if the subgraph $G_r(C_r, E_r)$ induced by C_r is complete and $E_r = E \cap (C_r \times C_r)$. Furthermore, clique C_r is a *maximal clique* if there exists no other complete subgraph that contains the subgraph $G_r(C_r, E_r)$. The number and element of low-dimension PSD constraints can be entirely determined by the maximal cliques. In general, the graph $G(V, E)$ induced by the power system is not necessarily chordal. Nevertheless, it can be extended as a chordal graph $G(V, F)$ through *chordal extension* if $G(V, F)$ is chordal and $E \subseteq F$. Chordal extension of a non-choral graph is not unique, different extensions features different sparsity. For a chordal graph $G(V, F)$, the family of maximal cliques can be identified with a *perfect elimination ordering (PEO)* of the vertex [20], [22], [23]. In addition, the maximal cliques can be indexed such that for each $r = 1, 2, ..., l-1$ it holds that,

$$\exists s \geq r+1: \quad C_r \cap (C_{r+1} \cup C_{r+2} \cup \cdots \cup C_l) \subseteq C_s. \quad (14)$$

where l is the number of maximal cliques. The property (14) is called the *running intersection property (RIP)*.



(a) graph G. (b) chordal extension.

To clarify above notations, consider an imaginary 7-bus power system with the topology $G(V, E)$ shown in Fig.1a. Since the circle $\{1, 5, 6, 7, 1\}$ and $\{6, 2, 4, 7, 6\}$ have no chord, thus $G(V, E)$ is not a chordal graph. By definition, the set $\{2, 3\}$ is a clique, but it is not a maximal clique because the subgraph induced by $\{2, 3\}$ is contained in the subgraph in induced by maximal clique $\{2, 3, 6\}$. Nevertheless, we can perform a chordal extension for $G(V, E)$ in Fig.1a by adding virtual connections $\{3, 4\}$, $\{5, 7\}$, $\{6, 4\}$ and obtain a chordal graph $G(V, F)$ as in Fig.1b. The chordal graph $G(V, F)$ has four maximal cliques and satisfies the *RIP* property, i.e., $C_1 = \{1, 5, 7\}$, $C_2 = \{2, 3, 4, 6\}$, $C_3 = \{4, 6, 7\}$, and $C_4 = \{5, 6, 7\}$. Note that, we can get another chordal extension graph, which has three maximal cliques by adding a virtual connection $\{1, 6\}$, and merging maximal cliques C_1 and C_4 to a new maximal clique $\{1, 5, 6, 7\}$, this is the so-called matrix combination (M-C) method. With the M-C method, the number of total maximal cliques will be decreased while the size of each maximal clique can be increased, and there is a tradeoff.

As we discussed in Section II.B, only the elements of X specified by sparsity pattern F (or aggregate sparsity pattern E) have contributions to the objective function in (7) and other elements influence the PSD property of X. The family of maximal cliques C_r inherits the sparsity pattern information of F. Thus only the elements associated with maximal cliques determines $A_0 \bullet X$. In this regard, the PSD matrix X can be decomposed to PSD submatrices $X_{C_r \times C_r}$, and the whole PSD matrix X can be completed with the PSD submatrices by identifying unknown elements. We call the symmetric matrix $\bar{X}$ with unknown elements as a partial matrix and use the notation $\bar{X} \in \mathbb{S}^n(F, ?)$ to represent that the known elements of $\bar{X}$ is specified by sparsity pattern F and unknown entry need to be identified through matrix completion.

B. Positive Semidefinite Matrix Completion

We introduce two theorems on the existence and uniqueness of the PSD matrix completion of $\bar{X} \in \mathbb{S}^n(F, ?)$ to facilitate the identification of the unknown elements.

Theorem 1: **Existences** [22], [33]. Let $G(V, F)$ be a chordal graph, $\mathrm{K} = \{C_1, C_2, \cdots, C_l\}$ be the family of associated maximal cliques, then the partial symmetric matrix $\bar{X}$ can be completed to a PSD (positive definite) symmetric matrix X if and only if it satisfies the clique-PSD (clique-PD) condition,

$$\bar{X}_{C_r \times C_r} \in \mathbb{S}_+^{C_r}, (\bar{X}_{C_r \times C_r} \in \mathbb{S}_{++}^{C_r}), \ r = 1, 2, \cdots, l. \quad (15)$$

Once getting the submatrices $\bar{X}_{C_r \times C_r}$, we can obtain a full PSD matrix through matrix completion. The completed matrix X from the partial matrix $\bar{X}$ is not unique, and there exist many PSD completions. Nevertheless, we can guarantee a unique positive definite completion by maximizing the determinant of the completed matrix X.

Theorem 2: **Uniqueness** [22], [33] Assuming that the partial symmetric matrix $\bar{X}$ has a positive definite matrix completion X. Then there exists a unique positive definite completion $\hat{X}$ of $\bar{X}$ that maximizes the determinant, i.e.,

$$\hat{X} = \arg\max_{X} \{\det(X)\} \quad (16)$$

Moreover, the elements of $\hat{X}^{-1}$ are fully determined by the sparsity pattern F, and other elements are zero, i.e.,

$$\hat{X}^{-1} \in \mathbb{S}^n(F, 0). \quad (17)$$

IV. Matrix Completion Embedded PDIP For OPF

In this section, we illustrate comprehensive analyses on the conversion method and the proposed MC-PDIP algorithm for the SDP relaxation of large-scale OPF problems.

A. Conversion Method for OPF-SDP

Theorem 1 implies that the SDP (7) can be converted to,

$$\begin{aligned} \min \ & \sum_{(i,j)\in F} [A_0]_{ij} X_{ij} \\ s.t. \ & \sum_{(i,j)\in F} [A_p]_{ij} X_{ij} = b_p, \ \ p = 1, 2, \cdots, m \\ & X_{C_r \times C_r} \in \mathbb{S}_+^{C_r}, \ r = \ 1, 2, \cdots l \end{aligned} \quad (18)$$

The motivation of the conversion method is to reformulate the sparse SDP as an equivalent block-diagonal SDP. Since the submatrices $X_{C_r \times C_r}$ have overlapping elements, auxiliary variables and consensus equality constraints need be associated to convert (18) into a standard form SDP as in [13], [25]. In this regard, SDP (18) can be reformulated as,

$$\begin{aligned} \min & \sum_{(i,j)\in F} [A_0]_{ij} X_{ij}^{\hat{r}(i,j)} \\ s.t. & \sum_{(i,j)\in F} [A_p]_{ij} X_{ij}^{\hat{r}(i,j)} = b_p, \;\; p = 1, 2, \cdots, m \\ & X_{C_r \times C_r} \in \mathbb{S}_+^{C_r}, \; r = \; 1, 2, \cdots l \\ & X_{ij}^r = X_{ij}^s, \;\; (i,j) \in (C_r \cap C_s) \times (C_r \cap C_s) \end{aligned} \tag{19}$$

where $\hat{r}(i,j) = \min\{r | (i,j) \in C_r \times C_r\}$. The coefficient matrix in SDP (19) can be rearranged to a block-diagonal structure which results in the main matrix computations inside PDIP Algorithm 1, consume less CPU time. It is worth pointing out that, although the conversion utilized in SDP (19) introduces sparsity to the dense Schur-complement system, it increases its order m substantially. The number of necessitated consistency equalities Δm is,

$$\Delta m = \sum_{(C_r, C_s)} \frac{1}{2} |C_r \cap C_s| \left(|C_r \cap C_s| + 1\right) \tag{20}$$

Without any matrix combining techniques, Δm reaches 4645 for the 1354 pegase system in our experiments, which is overwhelming for the PDIP Algorithm 1 even it exploits the block-diagonal sparsity structure with SDP (19). Alternatively, we propose to directly embed the matrix completion into the PDIP Algorithm 1 to avoid additional equality constraints, through performing a sparse clique factorization for the maximum determinant positive-definite completion $\hat{X}$, as indicated in Lemma 1.

B. Sparse Clique Factorization

***Lemma** 1:* [22] Let $G(V, F)$ be a chordal graph, and $\bar{X} \in \mathbb{S}^n(F, ?)$ be a partial symmetric matrix satisfying the clique-PD condition (15). Let P be a permutation matrix of $G(V, F)$ such that $(1, 2, \cdots, n)$ is a PEO for $P\bar{X}P^T$. Let $(C_1, C_2, \cdots, C_l)$ be a family of maximal cliques that enjoys the RIP (14). Define,

$$S_r = C_r \backslash \left(C_{r+1} \cup C_{r+2} \cup \cdots \cup C_l\right), \; r = 1, 2, \cdots, l$$

$$U_r = C_r \cap \left(C_{r+1} \cup C_{r+2} \cup \cdots \cup C_l\right), \; r = 1, 2, \cdots, l$$

Then the maximal determinant completion $\hat{X}$ of $\bar{X}$ can be factorized as,

$$P\hat{X}P^T = L_1^T L_2^T \cdots L_{l-1}^T D L_{l-1} \cdots L_2 L_1 \tag{21}$$

where $L_r \;\; (r = 1, 2, \cdots, l-1)$ is a lower triangular matrix given by

$$[L_r]_{ij} = \begin{cases} 1 & i = j \\ \left[\bar{X}_{U_r U_r}^{-1} \bar{X}_{U_r S_r}\right]_{ij} & i \in U_r, j \in S_r \\ 0 & otherwise \end{cases} \tag{22}$$

and D is a positive definite diagonal matrix given by,

$$D = diag\{D_{S_1 S_1}, D_{S_2 S_2}, \cdots, D_{S_l S_l}\} \tag{23}$$

with

$$D_{S_r S_r} = \begin{cases} \bar{X}_{S_r S_r} - \bar{X}_{S_r U_r} \bar{X}_{U_r U_r}^{-1} \bar{X}_{U_r S_r} & (r \neq l), \\ \bar{X}_{S_l S_l} & (r = l). \end{cases} \tag{24}$$

Since the clique-factorization (21) is sparse, we can incorporate this factorization into the search direction, and step length steps in Algorithm 1 to save computation time and memory, as the completion method in [22]. Since we embedded this factorization into the PDIP method, we do not need to associate consistency constraints to handle the overlapping as in the conversion method (19).

C. MC-PDIP Approach for OPF

1) Algorithm Framework: The MC-PDIP algorithm for SDP relaxations of OPF problems is elaborated in Algorithm 2. The essence of MC-PDIP is to perform the sparse clique factorization (21) of the primal matrix variable X, and benefit matrix computations by incorporating this sparsity. The main difference between PDIP Algorithm 1 and MC-PDIP Algorithm 2 includes three aspects: (i) The matrix completion (*step 0*) is performed at each iteration to obtain the sparse clique factorization (21) for further utilization; (ii) The search direction step (*step 2*) is performed with the sparse clique factorization in a much computation-cheaper and memory-saving fashion, while Algorithm 1 needs store the primal matrix variable X directly; (iii) The step length calculation (*step 3*) is performed on submatrices $X_{C_r C_r}$ induced by the chordal extension (*step 0*) while in Algorithm 1 it is performed on the entire X. Besides, we can also incorporate Mehrotra-type predictor-corrector rule as in [30] to the MC-PDIP algorithm to further improve its performance.

2) Computation Complexity: Define γ as the sparsity measure of the extended sparsity pattern F, i.e., $\gamma = |F|/n^2$, then we can roughly compare the computation complexity of the conversion method and our proposed MC-PDIP method. We use Cons-B, CF-SCS, and Step-L to represent the construction of the Schur complement system (*step 2.2*), Cholesky factorization for the Schur complement system (*step 2.3*), and the step length calculation (*step 3*), respectively, and the computation complexity is shown in Table I. Since γ becomes very small with the growth of power system scale, for instance, it reaches 0.0002 for the 13659 pegase system. Thus, Algorithm 2 is expected to provide an computationally friendly approach for large-scale OPF with SDP relaxation. In addition, Algorithm 1 need consumes $O((m + \Delta m)^2)$ memory to store B (*step 2.3*) while the MC-PDIP method only consumes $O(m^2)$ memory (*step 2.3*) and $O(\gamma n^2)$ memory (*step 3*), which is also attractive for further speed-up techniques.

TABLE I
COMPUTATION COMPLEXITY

Method	Cons-B (step 2.2)	CF-SCS (step 2.3)	Step-L (step 3)
PDIP for (19)	$O((m + \Delta m)^2)$	$O((m + \Delta m)^3$	$O(\gamma^{1.5} n^3)$
MC-PDIP for (7)	$O(m^2 + \gamma m n^2)$	$O(m^3)$	$O(\gamma n^3)$

3) Parallel Promises: The convex relaxations concerning multi-period OPF and contingency-constrained OPF enforce overwhelming computation overhead because of multiple time periods and target set of contingencies. Thus, the parallel capability is important for the scalability of the PDIP algorithm to such applications. As indicated previously, the PDIP Algorithm 1 has less parallel potentials, while the proposed MC-PDIP algorithm offers several parallel promises. First,

Algorithm 2 Matrix Completion Embedded PDIP Method

```
***Initialisation***: choose initial point $\left(\bar{X}^{(0)}, W^{(0)}, z^{(0)}\right) \in \mathbb{S}^n_{++} \times \mathbb{S}^n_{++} \times \mathbb{R}^m$, $\delta \in (0,1)$, set iteration index $k = 0$, set termination criterion.
***Step 0: Matrix completion*** Perform the maximal determinant completion for partial matrix $\bar{X}^{(k)}$, got the sparse clique factorization (21).
***Step 1: Convergence checking***: Set complementarity measure $\mu^{(k)} = \beta \hat{X}^{(k)} \bullet W^{(k)}/n,\ \beta \in [0,1]$, if current iterate $(\hat{X}^{(k)}, W^{(k)}, z^{(k)})$ satisfies the termination criterion, stop the algorithm.
***Step 2: Compute search direction*** $(\Delta X^{(k)}, \Delta W^{(k)}, \Delta z^{(k)}) \in \mathbb{S}^n \times \mathbb{S}^n \times \mathbb{R}^m$,
    *2.1* compute primal- and dual-residual,
        $$\begin{cases} r_p = b_p - A_p \hat{X}^{(k)} \\ R = A_0 - \sum\limits_{p=1}^{m} A_p z_p^{(k)} - W^{(k)} \end{cases} \tag{25}$$
    *2.2* calculate coefficient of Schur complement system
        $$\begin{cases} B_{pq} = A_p \bullet \hat{X}^{(k)} A_q (W^{(k)})^{-1} \\ s_p = r_p - A_p \bullet \left(K - \hat{X}^{(k)} R\right) \bar{W}^{-1} \left(W^{(k)}\right)^{-1} \\ C = \mu^{(k)} I - \hat{X}^{(k)} W^{(k)} \end{cases} \tag{26}$$
    *2.3* solve Schur complement system $B\Delta z^{(k)} = s$
    *2.4* determine other directions
        $$\begin{cases} \Delta W^{(k)} = R - \sum\limits_{p=1}^{m} A_p \Delta z_p^{(k)} \\ \Delta \tilde{X} = (C - \hat{X}^{(k)} \Delta W^{(k)})(W^{(k)})^{-1} \\ \Delta X^{(k)} = (\Delta \tilde{X} + \Delta \tilde{X}^T)/2 \end{cases} \tag{27}$$
***Step 3: Calculate step length*** $\alpha_p, \alpha_d$ through the minimum eigenvalue decomposition,
    $$(\bar{M}_r^{(k)})^{-1} \Delta \bar{X}^{(k)}_{C_r C_r} (\bar{M}_r^{(k)})^{-T}, (N^{(k)})^{-1} \Delta W^{(k)} (N^{(k)})^{-T} \tag{28}$$
    where $\bar{X}^{(k)}_{C_r C_r} {=} \bar{M}_r^{(k)} (\bar{M}_r^{(k)})^T, W^{(k)} = N^{(k)} (N^{(k)})^T$, and,
    $$\begin{cases} \bar{X}^{(k)}_{C_r C_r} + \alpha_p \Delta \bar{X}^{(k)}_{C_r C_r} \in \mathbb{S}^{C_r}_{+} \\ W^{(k)} + \alpha_d \Delta W^{(k)} \in \mathbb{S}^n_{+} (E, 0) \end{cases} \tag{29}$$
***Step 4: Next iteration point*** $\left(\bar{X}^{(k+1)}, W^{(k+1)}, z^{(k+1)}\right) \in \mathbb{S}^n_{++}(F, ?) \times \mathbb{S}^n_{++}(E, 0) \times \mathbb{R}^m$, i.e.,
    $$\left(\bar{X}^{(k+1)}, W^{(k+1)}, z^{(k+1)}\right) = \left(\bar{X}^{(k)}, W^{(k)}, z^{(k)}\right) + \delta \left(\alpha_p \Delta \bar{X}^{(k)}, \alpha_d \Delta W^{(k)}, \alpha_d \Delta z^{(k)}\right) \tag{30}$$
***Step 5*** Set $k \leftarrow k + 1$, go to step 0.
**return** $(X^*, W^*, z^*)$.
```

the sparse clique factorization (21) (*step 0*) relies on the number of maximal cliques, and the computation of each L_r can be paralleled; (ii) The calculation of coefficient matrix B_{pq} can be performed in a parallel fashion with the sparse clique factorization of $\hat{X}$; (iii) The determination of the step length for each submatrice $X_{C_r \times C_r}$ regarding each maximal clique can be parallelized. Note that, we perform Cholesky factorization to solve the Schur complement system *(step 2.3)* in Algorithm 2 to facilitate the comparison with PDIP Algorithm 1. Nevertheless, the Schur complement system can be addressed in a parallel manner with conjugate gradient method [34] or algebraic multi-grid method [35]. In this regard, the computational performance of Algorithm 2 can be further improved to develop a tailored parallel large-scale OPF solver. We are investigating the performance of the parallel MC-PDIP for the SDP relaxations of multi-period OPF problems, and more results shall be reported in future.

4) Exactness Justification: The exactness of SDP relaxations for OPF is essential to guide the operation of practical power systems. The proposed MC-PDIP method focus on the algorithm itself, not on the modeling, thus MC-PDIP shares similar exactness result with the widely used conversion method such as in [11]–[13]. If the relaxation is not tight, we can perform a penalized SDP relaxation for OPF problems [31] and then re-deploy the MC-PDIP algorithm. A low-rank solution is typically hard to get, so it would be more practical to get a feasible power flow solution with other local algorithms initialized with the SDP relaxation results of the MC-PDIP method.

V. Numerical Studies

We implemented the MC-PDIP on several benchmark power systems to validate its effectiveness to solve large-scale OPF problems with SDP relaxation. The experiments are based on the benchmark problems from the MATPOWER package [36], and we build the data matrices with YALMIP [37]. All experiments are conducted on a laptop with Intel i5-4210M CPU and 16GB RAM. Our algorithm is implemented in C++ to utilize several numerical routines, while the PDIP is deployed with SeDuMi [38].

As indicated previously, the number of m and the number of maximal cliques l, as well as the size of submatrix, depends on the specific Chordal extension. To attain the chordal extension F, we can perform a symbolic Cholesky factorization to the sparsity pattern E. Since the sparsity of OPF problem comes from the sparse power network connections, we can perform Cholesky factorization for $(|Im(Y)|)$ (Y is the admittance matrix) to obtain the chordal extension graph. However, the $|Im(Y)|$ cannot be guaranteed to be PSD for some power networks. Instead, we use the Cholesky factorization of the matrix $E = D^T D + I$, where D represents the incidence matrix associated with the power networks and $I \in \mathbb{R}^{n \times n}$ is identity matrix, as in [13]. With the Cholesky factorization, the sparsity pattern and extend sparsity pattern of the 13659 pegase system in are shown in Fig 1.

We compare the number of maximal cliques l, equality consistency constraints Δm (needed in the PDIP method) and sparsity measure γ for several benchmark power systems in

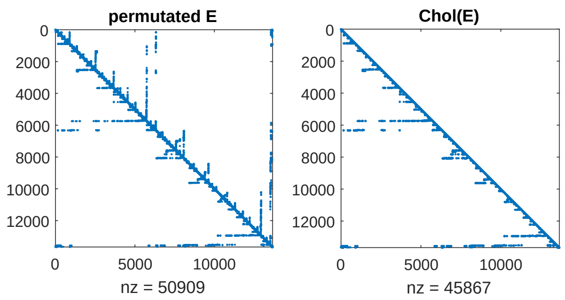


Fig. 1. Sparsity pattern and extended sparsity pattern.

TABLE II
Analysis of OPF data with SDP relaxation

Case	No. clique	Δm	γ	Δm (M-C)	γ (M-C)
IEEE-30	26	70	0.0944	17	0.17
IEEE-118	108	408	0.0276	95	0.0512
IEEE-300	278	1053	0.0107	258	0.0213
1354pegase	1287	4645	0.0022	1183	0.0049
2869pegase	2700	13090	0.0012	3205	0.0026
9241pegase	8577	57134	0.0004	13392	9.0270E-4
13659pegase	12997	61084	0.0002	14483	5.5791E-4

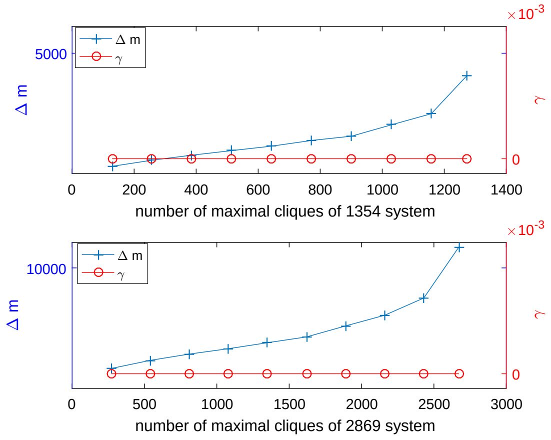


Fig. 2. Trajectory of Δm and γ with l.

TABLE III
Computational time of different maximal cliques (sec).

Clique ratio	IEEE-300	pegase 1354	pegase 2869
0.1	13.0488	162.3106	675.689
0.2	12.9625	188.2139	546.863
0.3	11.4738	250.6208	703.164
0.4	13.1454	172.6957	645.979
0.5	13.5853	173.2263	640.194
0.6	13.6754	278.0583	804.927
0.7	15.4896	188.9548	955.849
0.8	17.1464	324.5384	1047.745
0.9	18.859	318.7931	993.160
0.99	32.5032	263.2875	1294.362
1.0	**110.9108**	**FAIL**	**FAIL**

Table II. The term M-C denotes performing maximal cliques combination techniques as in [13] to obtain a 30% maximal clique of the original maximal cliques from Cholesky factorization listed in the second column. It is observed that Δm is typically huge for large-scale power systems, matrix combining techniques is adopted to decrease Δm and increase the size of submatrices. Nevertheless, the sparsity pattern γ is still small even after matrix combination to decrease the sparsity, which means Algorithm 2 is less sensitive with the number of maximal cliques. Also, we compare the trajectory of Δm and γ with the number of maximal cliques l for 1354 pegase system and 2869 pegase system in Fig.2. It is observed that the number of Δm is more sensitive to l, while the sparsity measure γ keep at the order of 1E3 and is less sensitive to the number of maximal cliques for large-scale power systems. In this regard, in Algorithm 2 we use the initial chordal extension with Cholesky factorization at step 0, for ease of analysis.

We compared the computational time of the conversion method on IEEE 300 system, pegase 1354 system and pegase 2869 system with different maximal cliques in Table III. The first column of Table III represents the ratio between the number of maximal cliques l compared with that of the initial Cholesky factorization. The ratio 0.99 stands for the initial factorization while the last line represents the Algorithm 1 without performing matrix completion. We can observe that the conversion method benefits the speed-up of the computational performance. Nevertheless, it is sensitive to the number of maximal cliques. The term FAIL indicates that the solver fails to solve the problem within 5 hours.

We illustrated the total execution time of the MC-PDIP and traditional conversion method for the PDIP on several benchmark power systems as in Table IV. Since the pegase-9241 and pegase-13659 system have numerous unbinding transmission line constraints in MatPower data, we failed to store the SDP problem, and we shall further reduce the unbinding constraints for ultra large-scale systems and implement the MC-PDIP method. From Table IV, it is observed that the MC-PDIP method is an alternative computationally friendly approach for large-scale OPF while its computation performance can be further enhanced with the parallel techniques we proposed in Section IV. We want to emphasize that SeDuMi used to perform Algorithm1 adopts the Mehrotra predictor-corrector (PC) rule to speed-up the PDIP method while in Algorithm 2 we didn't implement the PC rule and the performance can also be improved by incorporating PC rule further.

VI. Conclusion

We incorporated the chordal decomposition and matrix completion to the PDIP method to develop an MC-PDIP algorithm for the SDP relaxation of large-scale OPF problems. The proposed approach is an alternative computationally friendly method, and it features better efficiency, less memory, and the

TABLE IV
COMPARISON ON COMPUTATIONAL TIME (SEC).

Case	Conversion	Conversion-M-C	MC-PDIP
IEEE-30	3.6774	3.2539	2.8785
IEEE-118	18.3221	11.1288	10.665
IEEE-300	32.5032	11.4738	23.610
1354pegase	263.2875	250.6208	267.774
2869pegase	1294.36	703.164	448.039

same exactness compared to existing SDP relaxation models. Besides, the MC-PIDP method is easy to be performed in a parallel fashion, which is promising in multi-period OPF and contingency-constrained OPF problems. The applications of the parallel MC-PDIP method in other OPF variants should be validated, and we shall report the results on exactness, multi-period OPF in future investigations.

REFERENCES

[1] R. Madani, M. Ashraphijuo, and J. Lavaei, "Promises of conic relaxation for contingency-constrained optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1297–1307, Mar 2016.

[2] R. A. Jabr, S. Karaki, and J. A. Korbane, "Robust multi-period OPF with storage and renewables," *IEEE Transactions on Power Systems*, vol. 30, no. 5, pp. 2790–2799, Sep 2015.

[3] J. F. Marley, D. K. Molzahn, and I. A. Hiskens, "Solving multiperiod opf problems using an AC-QP algorithm initialized with an SOCP relaxation," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3538–3548, Sep 2017.

[4] M. Farivar and S. H. Low, "Branch flow model: Relaxations and convexificationpart I," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2554–2564, Aug 2013.

[5] B. Kocuk, S. S. Dey, and X. A. Sun, "Strong SOCP relaxations for the optimal power flow problem," *Operations Research*, vol. 64, no. 6, pp. 1177–1196, May 2016.

[6] X. Bai, H. Wei, K. Fujisawa, and Y. Wang, "Semidefinite programming for optimal power flow problems," *International Journal of Electrical Power & Energy Systems*, vol. 30, no. 6, pp. 383–392, Sep 2008.

[7] J. Lavaei and S. H. Low, "Zero duality gap in optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 92–107, Feb 2012.

[8] C. Coffrin, H. L. Hijazi, and P. Van Hentenryck, "The qc relaxation: A theoretical and computational study on optimal power flow," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 3008–3018, Jul 2016.

[9] H. Hijazi, C. Coffrin, and P. Van Hentenryck, "Convex quadratic relaxations for mixed-integer nonlinear programs in power systems," *Mathematical Programming Computation*, vol. 9, no. 3, pp. 321–367, Sep 2017.

[10] D. K. Molzahn and I. A. Hiskens, "Sparsity-exploiting moment-based relaxations of the optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 30, no. 6, pp. 3168–3180, Nov 2015.

[11] R. A. Jabr, "Exploiting sparsity in SDP relaxations of the OPF problem," *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 1138–1139, May 2012.

[12] M. S. Andersen, A. Hansson, and L. Vandenberghe, "Reduced-complexity semidefinite relaxations of optimal power flow problems," *IEEE Transactions on Power Systems*, vol. 29, no. 4, pp. 1855–1863, Jul 2014.

[13] 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 Transactions on Power Systems*, vol. 28, no. 4, pp. 3987–3998, Nov 2013.

[14] Y. Liu, J. Li, L. Wu, and T. Ortmeyer, "Chordal relaxation based ACOPF for unbalanced distribution systems with ders and voltage regulation devices," *IEEE Transactions on Power Systems*, 2017.

[15] E. Dall'Anese, H. Zhu, and G. B. Giannakis, "Distributed optimal power flow for smart microgrids," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1464–1475, Sep 2013.

[16] D. Gayme and U. Topcu, "Optimal power flow with large-scale storage integration," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 709–717, May 2013.

[17] B. Kocuk, S. S. Dey, and X. A. Sun, "Inexactness of SDP relaxation and valid inequalities for optimal power flow," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 642–651, Jan 2016.

[18] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM review*, vol. 38, no. 1, pp. 49–95, Mar 1996.

[19] R. D. Monteiro, "First-and second-order methods for semidefinite programming," *Mathematical Programming*, vol. 97, no. 1, pp. 209–244, Jul 2003.

[20] A. George, J. R. Gilbert, and J. W. Liu, *Graph theory and sparse matrix computation*. Springer Science & Business Media, Dec 2012, vol. 56.

[21] K. Fujisawa, M. Kojima, and K. Nakata, "Exploiting sparsity in primal-dual interior-point methods for semidefinite programming," *Mathematical Programming*, vol. 79, no. 1, pp. 235–253, Oct 1997.

[22] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, "Exploiting sparsity in semidefinite programming via matrix completion I: General framework," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 647–674, 2001.

[23] K. Nakata, K. Fujisawa, M. Fukuda, M. Kojima, and K. Murota, "Exploiting sparsity in semidefinite programming via matrix completion II: Implementation and numerical results," *Mathematical Programming*, vol. 95, no. 2, pp. 303–327, Feb 2003.

[24] S. Kim, M. Kojima, M. Mevissen, and M. Yamashita, "Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion," *Mathematical programming*, vol. 129, no. 1, pp. 33–68, Sep 2011.

[25] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn, "Chordal decomposition in operator-splitting methods for sparse semidefinite programs," *arXiv preprint arXiv:1707.05058*, Jul 2017.

[26] R. Madani, M. Ashraphijuo, and J. Lavaei, "Promises of conic relaxation for contingency-constrained optimal power flow problem," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1297–1307, Mar 2016.

[27] B. Ghaddar, J. Marecek, and M. Mevissen, "Optimal power flow as a polynomial optimization problem," *IEEE Transactions on Power Systems*, vol. 31, no. 1, pp. 539–546, Jan 2016.

[28] R. Madani, A. Kalbat, and J. Lavaei, "A low-complexity parallelizable numerical algorithm for sparse semidefinite programming," *IEEE Transactions on Control of Network Systems*, 2017.

[29] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn, "Chordal decomposition in operator-splitting methods for sparse semidefinite programs," *arXiv preprint arXiv:1707.05058*, Jul 2017.

[30] S. K. Pakazad, A. Hansson, M. S. Andersen, and A. Rantzer, "Distributed semidefinite programming with application to large-scale system analysis," *IEEE Transactions on Automatic Control*, Aug 2017.

[31] R. Madani, S. Sojoudi, and J. Lavaei, "Convex relaxation for optimal power flow problem: Mesh networks," *IEEE Transactions on Power Systems*, vol. 30, no. 1, pp. 199–211, Jan 2015.

[32] F. Alizadeh, J.-P. A. Haeberly, and M. L. Overton, "Primal-dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 746–768, Aug 1998.

[33] R. Grone, C. R. Johnson, E. M. Sá, and H. Wolkowicz, "Positive definite completions of partial hermitian matrices," *Linear algebra and its applications*, vol. 58, pp. 109–124, Apr 1984.

[34] R. Barrett, M. W. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, *Templates for the solution of linear systems: building blocks for iterative methods*. SIAM, 1994, vol. 43.

[35] C. Zhuo, J. Hu, M. Zhao, and K. Chen, "Power grid analysis and optimization using algebraic multigrid," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 4, pp. 738–751, Apr 2008.

[36] 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 Transactions on power systems*, vol. 26, no. 1, pp. 12–19, Feb 2011.

[37] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, Sep 2004, pp. 284–289.

[38] J. F. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, Jan 1999.