Appl. Set-Valued Anal. Optim. 5 (2023), No. 2, pp. 265-283
Available online at http://asvao.biemdas.com
https://doi.org/10.23952/asvao.5.2023.2.10

MINIMUM-RANK POSITIVE SEMIDEFINITE MATRIX COMPLETION WITH CHORDAL PATTERNS AND APPLICATIONS TO SEMIDEFINITE RELAXATIONS

XIN JIANG[1,*], YIFAN SUN[2], MARTIN S. ANDERSEN[3], LIEVEN VANDENBERGHE[4]

[1]*Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, PA 18015-1582, USA*
[2]*Department of Computer Science, Stony Brook University, Stony Brook, NY 11794-2424, USA*
[3]*Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kongens Lyngby, Denmark*
[4]*Department of Electrical and Computer Engineering, University of California Los Angeles, Los Angeles, CA 90095, USA*

Dedicated to Henry Wolkowicz on the occasion of his 75th birthday

Abstract. We present an algorithm for computing the minimum-rank positive semidefinite completion of a sparse matrix with a chordal sparsity pattern. This problem is tractable, in contrast to the minimum-rank positive semidefinite completion problem for general sparsity patterns. We also present a similar algorithm for the Euclidean distance matrix completion with minimum embedding dimension. The two algorithms use efficient recursions over a clique tree associated with the chordal sparsity pattern. As an application, we use the minimum-rank completion method as a rounding technique to convert the solution of a sparse semidefinite optimization problem with non-unique solutions to an optimal solution of lower rank. In experiments with semidefinite relaxations of optimal power flow problems, the minimum-rank completion often results in solutions of lower rank than the solutions computed by interior-point solvers.
Keywords. Chordal graphs; Euclidean distance matrix; Matrix completion; Optimal power flow; Semidefinite optimization.
2020 Mathematics Subject Classification. 90C22, 15A83.

1. Introduction

The theory of symmetric positive semidefinite and Euclidean distance matrix completions of matrices with partial patterns characterized by chordal graphs was developed in the 1980s and 1990s, with the celebrated 1984 paper by Grone, Johnson, Sá, and Wolkowicz [1] as one of the key contributions. Let $G = (V, E)$ be an undirected graph, where $V = \{1, 2, \ldots, n\}$ is the set of vertices and E the set of edges. In this paper, G represents the sparsity pattern of matrices in $\mathbb{S}^n$ (the symmetric $n \times n$ matrices). The set of symmetric $n \times n$ matrices with sparsity pattern E is

*Corresponding author.
E-mail addresses: xjiang@lehigh.edu (X. Jiang), yifan.sun@stonybrook.edu (Y. Sun), mskan@dtu.dk (M. S. Andersen), vandenbe@ucla.edu (L. Vandenberghe).
Received June 2, 2023; Accepted July 5, 2023.

defined as

$$\mathbb{S}^n_E = \{X \in \mathbb{S}^n \mid X_{ij} = X_{ji} = 0 \ \text{ if } i \neq j \text{ and } \{i,j\} \notin E\}. \tag{1.1}$$

Absence of an edge $\{i,j\}$ in the graph indicates a zero in positions (i,j) and (j,i). The projection of a matrix $X \in \mathbb{S}^n$ on $\mathbb{S}^n_E$ is denoted by $\Pi_E(X)$. This is the matrix in $\mathbb{S}^n_E$ with entries

$$(\Pi_E(X))_{ij} = \begin{cases} X_{ij} & \{i,j\} \in E \\ X_{ii} & i = 1, \ldots, n \\ 0 & \text{otherwise.} \end{cases}$$

With this notation, the positive semidefinite (PSD) matrix completion problem is to find a matrix $X \in \mathbb{S}^n$ that satisfies

$$X \in \mathbb{S}^n_+, \qquad \Pi_E(X) = A, \tag{1.2}$$

where $A \in \mathbb{S}^n_E$ is given and $\mathbb{S}^n_+$ denotes the set of positive semidefinite matrices in $\mathbb{S}^n$. Clearly, a necessary condition for existence of a solution is that

$$A_{\gamma\gamma} \in \mathbb{S}^{|\gamma|}_+ \quad \text{for all maximal cliques } \gamma. \tag{1.3}$$

The cliques are the subsets of V that induce complete subgraphs and correspond to dense principal submatrices of matrices in $\mathbb{S}^n_E$. Therefore, if $\Pi_E(X) = A$ then $X_{\gamma\gamma} = A_{\gamma\gamma}$ for every clique γ, and these matrices must be positive semidefinite if X is positive semidefinite. If the graph G is *chordal, i.e.*, every cycle of length greater than three has a chord, then the necessary condition (1.3) is also sufficient for existence of a positive semidefinite completion [1, Theorem 7]. If the graph is not chordal, there exist matrices $A \in \mathbb{S}^n_E$ that satisfy (1.3), but do not have a positive semidefinite completion [1, Theorem 7].

A similar result holds for the Euclidean distance matrix completion problem. A matrix $X \in \mathbb{S}^n$ is a *Euclidean distance matrix* (EDM) if its entries can be expressed as squared pairwise Euclidean distances of a set of points, *i.e.*, there exist vectors $y_1, \ldots, y_n$ such that

$$X_{ij} = \|y_i - y_j\|^2, \quad i,j = 1, \ldots, n, \tag{1.4}$$

where $\|\cdot\|$ is the Euclidean norm. The dimension of the vectors y_i is arbitrary. We denote the set of $n \times n$ EDMs by $\mathbb{D}^n$. The Euclidean distance matrix completion problem is to find $X \in \mathbb{S}^n$ that satisfies

$$X \in \mathbb{D}^n, \qquad \Pi_E(X) = A, \tag{1.5}$$

where $A \in \mathbb{S}^n_E$ is given. A necessary condition for existence of a solution is that

$$A_{\gamma\gamma} \in \mathbb{D}^{|\gamma|} \quad \text{for all maximal cliques } \gamma. \tag{1.6}$$

For chordal graphs, this necessary condition is again sufficient [2, Theorem 3.3]. If the graph is not chordal, matrices $A \in \mathbb{S}^n_E$ exist that satisfy (1.6) but do not have an EDM completion [2, p.651]. We refer the interested reader to [8, 3, 6, 7, 5, 4] for surveys on the positive semidefinite and EDM completion problems, the connections between them, and their many applications.

In this paper, we consider the problem of finding a solution of (1.2) of minimum rank, and the problem of finding a solution of (1.5) of minimum embedding dimension, where embedding dimension refers to the dimension of the vectors y_i in (1.4) (the embedding dimension is the rank of the projection of X on the complement of the all-ones vector; see Section 4 and [9, §11.3]). These criteria add non-convex objectives to the convex constraints (1.2) and (1.5), and lead to optimization problems that are difficult to solve in general. For chordal patterns, however, they are very tractable. The minimum rank of any PSD completion of a matrix A

that satisfies (1.3) is given by $\max_\gamma \mathbf{rank}(A_{\gamma\gamma})$, where the maximum is over all maximal cliques [10, Theorem 1.3]. Similarly, the minimum dimension of any EDM completion of a matrix A that satisfies (1.6) is the maximum of the dimensions of $A_{\gamma\gamma}$ over all maximal cliques [2, Theorem 3.3]. The proofs of these theorems in [10, 2] are constructive and explain how to find the desired completions one matrix entry at a time. In Sections 3 and 4, we present different and more efficient algorithms that use iterations over a clique tree associated with the chordal graph. The iterations and data structures are similar to the algorithms discussed in [11, 9] for several closely related problems, including maximum-determinant positive semidefinite completion, and computing gradients and directional second derivatives of logarithmic barriers for cones of sparse positive semidefinite matrices and their dual cones. The algorithms are variants of supernodal versions of the multifrontal algorithm for sparse Cholesky factorization.

This work is motivated by applications in semidefinite and Euclidean distance matrix optimization, *i.e.*, matrix optimization problems that include constraints of the form $X \in \mathbb{S}^n_+$ or $X \in \mathbb{D}^n$. The most important example is the semidefinite programming problem (SDP)

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(CX) \\ \text{subject to} & \mathbf{tr}(A_k X) = b, \quad k = 1, \ldots, m \\ & X \in \mathbb{S}^n_+. \end{array} \tag{1.7}$$

The variable X in this problem and the coefficients $C, A_1, \ldots, A_m$ are symmetric matrices of order n. In semidefinite and EDM optimization problems it is very common that, except for the constraints $X \in \mathbb{S}^n_+$ or $X \in \mathbb{D}^n$, the constraints and the objective function depend on a small subset of the entries of the matrix variable. In the SDP (1.7), for example, this happens when $C, A_1, \ldots, A_m$ are sparse. Suppose $C, A_1, \ldots, A_m \in \mathbb{S}^n_E$. The sparsity pattern E can be the common (or aggregate) sparsity pattern, *i.e.*, the union of the sparsity patterns of $C, A_1, \ldots, A_m$, or an extension of the common sparsity pattern. Since $C, A_1, \ldots, A_m \in \mathbb{S}^n_E$, the inner products in the objective function and the linear constraints of (1.7) do not depend on the variables X_{ij} with $i \neq j$, $\{i, j\} \notin E$. If $\widetilde{X}$ is feasible in (1.7), then any other positive semidefinite matrix X that satisfies $\Pi_E(X) = \Pi_E(\widetilde{X})$ is also feasible, with the same value $\mathbf{tr}(CX) = \mathbf{tr}(C\widetilde{X})$ of the objective function. This property has useful implications for optimization algorithms.

First, suppose the optimal solution of the matrix optimization problem is not unique. Let $X^\star$ be an optimal solution returned by an algorithm, for example, one of the general-purpose interior-point solvers for solving (1.7). In many applications, for example, semidefinite relaxations of combinatorial or nonconvex polynomial optimization problems, one is most interested in a low-rank solution (if possible, the minimum-rank solution) of (1.7). However, interior-point algorithms that follow the central path will return a solution close to the limit of the central path, and this is unlikely to be the minimum-rank optimal solution. The minimum-rank completions presented in Section 3 can be used to replace $X^\star$ with an optimal solution of lower rank. We first find a chordal extension E of the common sparsity pattern of the coefficient matrices, and then calculate the minimum-rank completion of $\Pi_E(X^\star)$. Note that this is not necessarily the minimum-rank solution of the SDP, since the entries of $\Pi_E(X^\star)$ for $X^\star$ in the set of optimal solutions may not be unique. In Section 5 we evaluate this rank-reduction technique on a set of SDP relaxations of the AC optimal power flow problem, and observe that it often results in a substantial reduction of the rank. Similarly, in matrix optimization problems involving EDM constraints $X \in \mathbb{D}^n$, one can replace the solution $X^\star$ computed by any algorithm by

the minimum-dimension completion of $\Pi_E(X^\star)$, where E includes the positions of the matrix entries X_{ij} that appear in the other constraints and the objective.

A second application is the class of SDP algorithms based on positive semidefinite completion techniques. These algorithms exploit the sparsity in (1.7) by computing only $\Pi_E(X)$ where E is a chordal extension of the common sparsity pattern of the coefficient matrices. In other words, they return the solution of the conic linear programming problem

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(CX) \\ \text{subject to} & \mathbf{tr}(A_k X) = b, \quad k = 1, \ldots, m \\ & X \in K, \end{array} \tag{1.8}$$

where $K = \Pi_E(\mathbb{S}^n_+)$ is the convex cone of matrices in $\mathbb{S}^n_E$ that have a positive semidefinite completion. The variable X in (1.8) is a matrix in $\mathbb{S}^n_E$, and this can be a much lower-dimensional space than $\mathbb{S}^n$. Algorithms for (1.8) use classical results from the theory of positive semidefinite completion to handle the constraint $X \in K$. The idea was first proposed in [12], and has been applied in interior-point algorithms, decomposition methods, and first-order methods [13, 14, 15, 9, 18, 16, 17]. Methods that solve (1.7) via (1.8) must be followed by a completion step to find an optimal solution of (1.7). The most appropriate choice is the completion of minimum rank, because in most applications one is interested in low-rank solutions. Moreover, since the optimal solution of (1.8) is on the boundary of the cone K, the simpler and more widely known maximum-determinant positive definite completion is not defined at the optimum of (1.8). Similar ideas apply to EDM optimization.

Contributions. We first present an algorithm for constructing the minimum-rank PSD completion of a matrix $A \in \mathbb{S}^n$ with a chordal sparsity pattern E. The expression for the minimum rank is given in [10, Theorem 1.3], and the presented algorithm serves as a constructive proof for this result. The algorithm exploits the structure of the clique tree associated with the chordal sparsity pattern.

As a second contribution, we use the algorithm as a posterior rounding step for SDP solutions, and test this technique on semidefinite relaxations of a set of optimal power flow problems. On most problems, the rounding step is observed to reduce the rank of the computed solutions. In some cases a solution of rank one is obtained, which corresponds to a global solution of the underlying nonconvex quadratic optimization problem.

The third contribution is a new algorithm for the minimum-dimension EDM completion algorithm with chordal sparsity. The algorithm is obtained by adapting the key ideas in the minimum-rank PSD completion method and uses a similar recursion over the clique tree.

Outline. The rest of the paper is organized as follows. In Section 2 we review basic concepts in chordal sparsity and graph theory. The minimum-rank PSD completion algorithm is described in Section 3. Section 4 presents the minimum-dimension EDM completion algorithm. In Section 5, the proposed algorithm is applied as a posterior rounding process to sparse semidefinite relaxations of the optimal power flow problem and numerical results are included.

2. Chordal Sparsity Patterns

In this section we review some basic properties of symmetric chordal sparsity patterns, and define the notation and assumptions that will be used throughout the paper. A more detailed discussion and proofs can be found in the surveys [19, 9, 17].

We assume the graph $G = (V, E)$ represents a symmetric $n \times n$ sparsity pattern, as defined in (1.1). The vertex set $V = \{1, 2, \ldots, n\}$ contains the row and column indices. The edge set E indicates the positions of the off-diagonal nonzeros. We assume that the graph $G = (V, E)$ is *chordal*, *i.e.*, every cycle of length at least four has a chord. It is a fundamental result that a graph is chordal if and only if it has a *perfect elimination ordering* [20]. Without loss of generality, we assume that the numerical order $1, \ldots, n$ is a perfect elimination ordering, *i.e.*,

$$i > j > k, \quad \{i, k\} \in E, \quad \{j, k\} \in E \qquad \Longrightarrow \qquad \{i, j\} \in E.$$

For simplicity we also assume that the graph is connected.

A maximal complete subgraph of a graph is called a (maximal) clique. The vertex of a clique with the smallest index is called its *representative vertex*. The clique with representative vertex i is denoted by γ_i, and the set of representative vertices by $V^c \subset V$.

We associate with the graph a clique tree. The nodes in the clique tree are the cliques, indexed by their representative vertices V^c. If the clique γ_i is not the root of the clique tree, the representative vertex of its parent in the clique tree is denoted by $p(i)$. A fundamental property of chordal graphs is that there exists a clique tree with the *induced subtree property:* for each $k \in V$, the cliques that contain the vertex k form a subtree of the clique tree [22, 21]. Given a clique tree with the induced subtree property, one can partition each clique γ_i in two sets ν_i and α_i defined as follows. If γ_i is the root of the clique tree, $\alpha_i = \emptyset$ and $\nu_i = \gamma_i$. Otherwise,

$$\alpha_i = \gamma_i \cap \gamma_{p(i)}, \qquad \nu_i = \gamma_i \setminus \alpha_i.$$

The sets ν_i are called the *(maximal) supernodes* or clique *residuals*, and the sets α_i are called the clique *separators*. It follows from the induced subtree property that the supernodes ν_i, for $i \in V^c$, partition V. If $k \in \nu_j$, then the clique γ_j is the root of the induced subtree for vertex k. The other cliques γ_i in the induced subtree contain k in the clique separators α_i.

We assume the elements in γ_i, α_i, ν_i are ordered in ascending order. By a suitable ordering of the vertices and choice of clique tree we can further assume the following properties.

- The representative vertices of the cliques in the clique tree are ordered topologically: $i < p(i)$ if $i \neq \max V^c$, and $i = \max V^c$ is the representative vertex of the root of the clique tree.
- The vertices in ν_i are ordered consecutively: $\nu_i = \{i, i + 1, \ldots, i + |\nu_i| - 1\}$.

Figure 1 shows an example.

3. Minimum-rank Chordal PSD Completion

3.1. **PSD completion.** We assume E is a chordal sparsity pattern and use the notation in Section 2. A classical result in matrix algebra says that a matrix $A \in \mathbb{S}^n_E$ has a positive semidefinite completion if and only if

$$A_{\gamma_i \gamma_i} \in \mathbb{S}^{|\gamma_i|}_+, \quad i \in V^c; \tag{3.1}$$

see [1]. Dancis [10, Theorem 1.5] has shown that every matrix $A \in \Pi_E(\mathbb{S}^n_+)$ has a positive semidefinite completion with rank equal to

$$r = \max_{i \in V^c} \mathbf{rank}\,(A_{\gamma_i \gamma_i}). \tag{3.2}$$

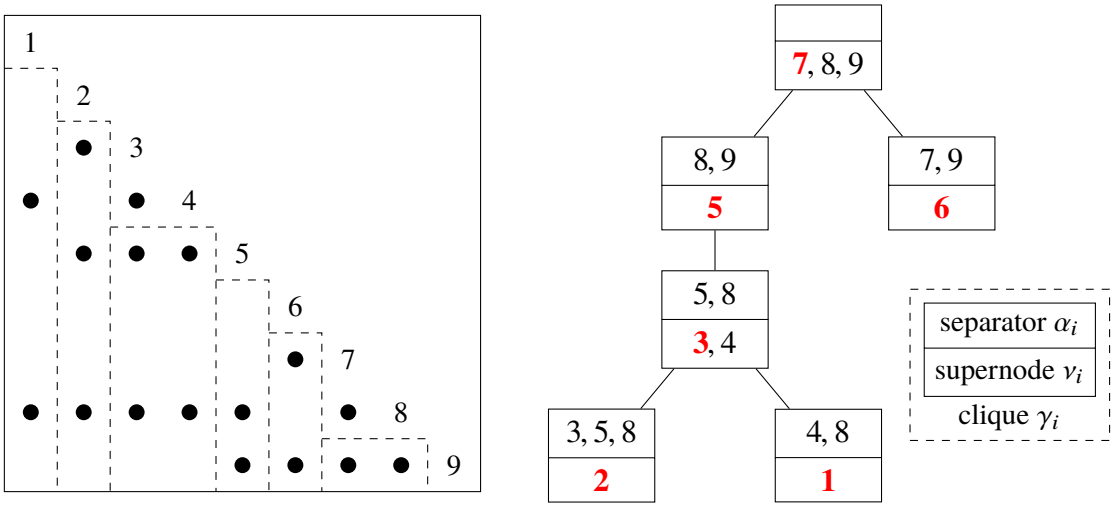

FIGURE 1. *Left.* The (symmetric) adjacency matrix of a connected, undirected, chordal graph G, with perfect elimination ordering $1, \ldots, n$. (For simplicity we only show the lower triangular part of the adjacency matrix.) A bullet in the (i, j) entry means the nodes i and j are adjacent. The dashed lines separate the supernodes. *Right.* A corresponding clique tree. Every double-row box in the clique tree is a clique γ_i. The red index is the clique representative. The bottom row in each rectangle is the supernode (clique residual) ν_i and the top row is the separator α_i.

In particular, A has a positive semidefinite completion with rank less than or equal to

$$r_{\max} = \max_{i \in V^c} |\gamma_i|, \tag{3.3}$$

the size of the largest clique in the sparsity pattern.

In Sections 3.2 and 3.3, we describe an algorithm to compute a full-rank matrix Y of size $n \times r$ with the property $\Pi_E(YY^T) = A$. We will use the following well-known result from linear algebra (see, for example, [23, Lemma 3], [24, Lemma 2.1], [25, Proposition 3.2]).

Lemma 3.1. *If A and B are matrices of the same size that satisfy $AA^T = BB^T$, then $A = BQ$ for some orthogonal matrix Q.*

Proof. Suppose $A, B \in \mathbb{R}^{n \times m}$ and $AA^T = BB^T$. The relation $AA^T = BB^T$ implies that A and B have the same rank, singular values, and left singular vectors. Therefore they have singular value decompositions of the form

$$A = P\Sigma V^T, \qquad B = P\Sigma U^T,$$

where $P \in \mathbb{R}^{n \times n}$, $\Sigma \in \mathbb{R}^{n \times m}$, and $V, U \in \mathbb{R}^{m \times m}$, with P, U, V orthogonal. Hence $A = BQ$ for $Q = UV^T$. □

The matrix $Q = UV^T$ constructed in the proof is the orthogonal factor in the polar decomposition $B^T A = QH$, where Q is orthogonal and $H = VSV^T$ with $S = \Sigma^T \Sigma$ symmetric positive semidefinite. For general $n \times m$ matrices A, B, this matrix is known to be the solution of the orthogonal Procrustes problem, *i.e.*, it minimizes $\|A - BQ\|_F$ over the orthogonal matrices; see [26], [27, §7.4.8]. The matrices U, V can be computed from a singular value decomposition

$B^T A = USV^T$, preferably via specialized algorithms that compute this SVD without forming the product $B^T A$ [28, 29].

3.2. **Two overlapping diagonal blocks.** The key part of the algorithm is first explained via an illustrative example with a sparsity pattern consisting of two overlapping diagonal blocks. The next section (Section 3.3) covers the extension to general chordal sparsity patterns, using the clique tree data structure.

Consider the completion of a symmetric matrix with two overlapping diagonal blocks, partitioned as

$$A = \begin{bmatrix} A_{11} & A_{21}^T & 0 \\ A_{21} & A_{22} & A_{32}^T \\ 0 & A_{32} & A_{33} \end{bmatrix},$$

with A_{11}, A_{22}, A_{33} of size $n_1 \times n_1$, $n_2 \times n_2$, and $n_3 \times n_3$, respectively. Here $V^{\mathrm{c}} = \{1, n_1 + 1\}$, the two cliques are

$$\gamma_1 = \{1, \ldots, n_1 + n_2\}, \qquad \gamma_{n_1+1} = \{n_1 + 1, \ldots, n_1 + n_2 + n_3\},$$

and the corresponding supernodes and separators are

$$\nu_1 = \{1, \ldots, n_1\}, \qquad \alpha_1 = \{n_1 + 1, \ldots, n_1 + n_2\}, \qquad \nu_{n_1+1} = \gamma_{n_1+1}, \qquad \alpha_{n_1+1} = \{\}.$$

Assume $A \in \Pi_E(\mathbb{S}_+^n)$, and define

$$H_1 = \begin{bmatrix} A_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix}, \qquad H_2 = \begin{bmatrix} A_{22} & A_{32}^T \\ A_{32} & A_{33} \end{bmatrix}.$$

From (3.1) and (3.2), the matrices H_1, H_2 are positive semidefinite and the minimum rank of a positive semidefinite completion of A is given by

$$r = \max\{\mathbf{rank}(H_1), \mathbf{rank}(H_2)\}.$$

A matrix Y of size $n \times r$ that satisfies $\Pi_E(YY^T) = A$ can be constructed as follows.

By definition of r, the matrices H_1 and H_2 can be decomposed as

$$H_1 = \begin{bmatrix} A_{11} & A_{21}^T \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} U_1 \\ V_1 \end{bmatrix} \begin{bmatrix} U_1 \\ V_1 \end{bmatrix}^T, \qquad H_2 = \begin{bmatrix} A_{22} & A_{32}^T \\ A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} U_2 \\ V_2 \end{bmatrix} \begin{bmatrix} U_2 \\ V_2 \end{bmatrix}^T,$$

where U_1, U_2, V_1, V_2 have r columns. The submatrix A_{22} satisfies $A_{22} = V_1 V_1^T = U_2 U_2^T$. From Lemma 3.1, the matrices V_1 and U_2 are related as $U_2 = V_1 Q$ for an $r \times r$ orthogonal matrix Q. Define

$$Y = \begin{bmatrix} U_1 Q \\ U_2 \\ V_2 \end{bmatrix} = \begin{bmatrix} U_1 Q \\ V_1 Q \\ V_2 \end{bmatrix}.$$

By construction, Y has rank r. The identity $\Pi_E(YY^T) = A$ can be verified as

$$X = YY^T = \begin{bmatrix} U_1 U_1^T & U_1 V_1^T & U_1 Q V_2^T \\ V_1 U_1^T & V_1 V_1^T & U_2 V_2^T \\ V_2 Q^T U_1^T & V_2 U_2^T & V_2 V_2^T \end{bmatrix} = \begin{bmatrix} A_{11} & A_{21}^T & U_1 Q V_2^T \\ A_{21} & A_{22} & A_{32}^T \\ V_2 Q^T U_1^T & A_{32} & A_{33} \end{bmatrix}.$$

3.3. **Minimum-rank PSD completion algorithm.** The technique in Section 3.2 can be extended to handle general chordal sparsity patterns, by traversing the clique tree in an inverse topological order (starting at the root and visiting each clique before its children in the clique tree). The algorithm is summarized in Algorithm 1. For a matrix $Y \in \mathbb{R}^{n\times m}$ and an index set $\beta \subset \{1,\ldots,n\}$, we use Y_β to denote the submatrix of Y with rows indexed by β.

Algorithm 1. *Minimum-rank chordal PSD completion.*

Input. A matrix $A \in \Pi_E(\mathbb{S}^n_+)$, where $G = (V,E)$ is a chordal sparsity pattern, and a clique tree for G.

Output. A matrix $Y \in \mathbb{R}^{n\times r}$ with r equal to (3.2) and $\Pi_E(YY^T) = A$.

Algorithm.

- Compute $r = \max_{i\in V^c} \mathbf{rank}(A_{\gamma_i\gamma_i})$ and, for each $j \in V^c$, a factorization
$$A_{\gamma_j\gamma_j} = \begin{bmatrix} A_{\nu_j\nu_j} & A_{\nu_j\alpha_j} \\ A_{\alpha_j\nu_j} & A_{\alpha_j\alpha_j} \end{bmatrix} = \begin{bmatrix} U_j \\ V_j \end{bmatrix} \begin{bmatrix} U_j \\ V_j \end{bmatrix}^T,$$
with U_j of size $|\nu_j| \times r$ and V_j of size $|\alpha_j| \times r$.
- Enumerate the clique representatives $j \in V^c$ in an inverse topological order. If j is the root of the clique tree, set $Y_{\nu_j} := U_j$. Otherwise, compute an $r \times r$ orthogonal matrix Q such that $Y_{\alpha_j} = V_jQ$ and set $Y_{\nu_j} := U_jQ$.

The two parts of the algorithm can be combined. We can start with $r = 0$ and increase it to $\max\{\mathbf{rank}(A_{\gamma_j\gamma_j}), r\}$ in step j of the recursion. When r is increased, we add zero columns to the blocks of Y that have already been computed. Alternatively, we can set r to the upper bound (3.3), and determine the rank of the matrix Y after finishing the algorithm.

To show correctness of the algorithm, we verify that YY^T is a completion of A, *i.e.*,
$$Y_{\gamma_j} Y_{\gamma_j}^T = A_{\gamma_j\gamma_j} \tag{3.4}$$
for all $j \in V^c$. Recall that the supernodes ν_j partition the index set $V = \{1,2,\ldots,n\}$, and that each set α_j is a subset of the union of the sets ν_i for the ancestors γ_i of clique γ_j in the clique tree. At each step in the recursion over the tree, we add a new block Y_{ν_j}. The blocks Y_{ν_i} for the cliques γ_i that precede γ_j in the inverse topological ordering are left unchanged. It is therefore sufficient to verify that after $j \in V^c$ has been processed, the identity (3.4) holds, *i.e.*,
$$\begin{bmatrix} Y_{\nu_j} \\ Y_{\alpha_j} \end{bmatrix} \begin{bmatrix} Y_{\nu_j} \\ Y_{\alpha_j} \end{bmatrix}^T = \begin{bmatrix} A_{\nu_j\nu_j} & A_{\nu_j\alpha_j} \\ A_{\alpha_j\nu_j} & A_{\alpha_j\alpha_j} \end{bmatrix}.$$
If j is the root representative, then
$$Y_{\nu_j} Y_{\nu_j}^T = U_jU_j^T = A_{\nu_j\nu_j} = A_{\gamma_j\gamma_j}.$$
Otherwise, suppose $Y_{\gamma_i} Y_{\gamma_i}^T = A_{\gamma_i\gamma_i}$ for all cliques γ_i that are ancestors of clique γ_j in the clique tree. Then
$$A_{\alpha_j\alpha_j} = Y_{\alpha_j} Y_{\alpha_j}^T = V_jV_j^T$$
because α_j is a subset of the parent clique $\gamma_{p(j)}$. By Lemma 3.1 there exists an orthogonal matrix Q such that $Y_{\alpha_j} = V_jQ$. By choosing $Y_{\nu_j} = U_jQ$ we obtain
$$\begin{bmatrix} Y_{\nu_j} \\ Y_{\alpha_j} \end{bmatrix} \begin{bmatrix} Y_{\nu_j} \\ Y_{\alpha_j} \end{bmatrix}^T = \begin{bmatrix} U_jQ \\ Y_{\alpha_j} \end{bmatrix} \begin{bmatrix} U_jQ \\ Y_{\alpha_j} \end{bmatrix}^T$$

$$= \begin{bmatrix} U_j U_j^T & U_j Q Y_{\alpha_j}^T \\ Y_{\alpha_j} Q^T U_j^T & Y_{\alpha_j} Y_{\alpha_j}^T \end{bmatrix}$$
$$= \begin{bmatrix} U_j U_j^T & U_j V_j^T \\ V_j U_j^T & V_j V_j^T \end{bmatrix}$$
$$= \begin{bmatrix} A_{\nu_j \nu_j} & A_{\nu_j \alpha_j} \\ A_{\alpha_j \nu_j} & A_{\alpha_j \alpha_j} \end{bmatrix}.$$

4. Minimum-dimension Chordal EDM Completion

4.1. **EDM completion.** We now turn to the EDM completion problem. As mentioned in Section 1, a symmetric $n \times n$ matrix X is an EDM if there exist vectors $y_1, \ldots, y_n$ such that (1.4) holds. The condition (1.4) can be written in matrix form as

$$X = \mathbf{diag}(YY^T)\mathbf{1}^T + \mathbf{1}\,\mathbf{diag}(YY^T)^T - 2YY^T, \tag{4.1}$$

where $\mathbf{1}$ is the n-vector of ones, Y is the matrix with rows y_i^T, and the linear operator $\mathbf{diag}\colon \mathbb{S}^n \to \mathbb{R}^n$ maps a matrix to the vector of its diagonal elements. We refer to the matrix Y as a *realization* of X. From (4.1), $X \in \mathbb{D}^n$ if and only if there exists a matrix $W \in \mathbb{S}_+^n$ that satisfies

$$X = \mathbf{diag}(W)\mathbf{1}^T + \mathbf{1}\,\mathbf{diag}(W)^T - 2W.$$

An equivalent characterization of EDMs is due to Schoenberg [30, 31]: X is an EDM if its diagonal is zero and its projection on the complement of the all-ones vector is negative semidefinite, *i.e.*,

$$\mathbf{diag}(X) = 0, \qquad P^T X P \preceq 0,$$

where P is a matrix whose columns span the orthogonal complement of $\mathbf{1}$. We define the *dimension* of the EDM X as the rank of its projection on the orthogonal complement of the all-ones vector:

$$\dim(X) = \mathbf{rank}(P^T X P).$$

The following lemma implies that $\dim(X)$ is the minimum dimension of any realization Y of X. The lemma will be exploited in the algorithm in Section 4.2.

Lemma 4.1. *Let μ be an n-vector that satisfies $\mathbf{1}^T\mu = 1$, and suppose $X \in \mathbb{D}^n$. A matrix $Y \in \mathbb{R}^{n\times m}$ is a realization of X that satisfies $\mu^T Y = 0$ if and only if*

$$YY^T = -\frac{1}{2}(I - \mathbf{1}\mu^T)X(I - \mu\mathbf{1}^T). \tag{4.2}$$

Proof. Suppose Y is a realization of X and satisfies $\mu^T Y = 0$. From (4.1) we obtain

$$(I - \mathbf{1}\mu^T)X(I - \mu\mathbf{1}^T) = -2YY^T.$$

For the other direction, assume Y satisfies (4.2). Clearly, $\mu^T YY^T\mu = 0$, so $\mu^T Y = 0$. Also,

$$\mathbf{diag}(YY^T) = -\frac{1}{2}\,\mathbf{diag}(X - \mathbf{1}\mu^T X - X\mu\mathbf{1}^T + (\mu^T X\mu)\mathbf{1}\mathbf{1}^T)$$
$$= X\mu - \frac{1}{2}(\mu^T X\mu)\mathbf{1},$$

because $\mathbf{diag}(X) = 0$. Hence

$$\begin{aligned}
&\mathbf{diag}(YY^T)\mathbf{1}^T + \mathbf{1}\,\mathbf{diag}(YY^T)^T - 2YY^T \\
&= X\mu\mathbf{1}^T + \mathbf{1}\mu^T X - (\mu^T X\mu)\mathbf{1}\mathbf{1}^T + (I - \mathbf{1}\mu^T)X(I - \mu\mathbf{1}^T) \\
&= X.
\end{aligned}$$

Therefore, Y is a realization of X. □

Lemma 4.1 shows that if we impose the condition $\mu^T Y = 0$, the product YY^T of the realization Y is uniquely defined. Hence, from Lemma 3.1, the realization Y that satisfies $\mu^T Y = 0$ is unique, up to right multiplication with an orthogonal matrix. The condition $\mu^T Y = 0$ places the origin at a specified affine combination of the rows of Y.

4.2. **Minimum-dimension EDM completion algorithm.** We consider the EDM completion problem with chordal sparsity pattern E. Using the same notation as in the previous sections, the main result on this problem is as follows [2, Theorem 3.3], [4, Theorem 5.8.5]: a matrix $A \in \mathbb{S}_E^n$ has an EDM completion if and only if

$$A_{\gamma_i\gamma_i} \in \mathbb{D}^{|\gamma_i|}, \quad i \in V^c.$$

Moreover, there exists a completion with dimension

$$\dim(X) = \max_{i \in V^c} \dim(A_{\gamma_i\gamma_i}). \tag{4.3}$$

The following algorithm computes a completion with this dimension. In the algorithm, the vector $e_1 = (1, 0, \ldots, 0)$ is the first unit vector of compatible size.

Algorithm 2. *Minimum-dimension chordal EDM completion.*

Input. A matrix $A \in \Pi_E(\mathbb{D}^n)$, where $G = (V, E)$ is a chordal sparsity pattern, and a clique tree for G.

Output. A realization $Y \in \mathbb{R}^{n\times r}$ of an EDM completion of A with r equal to (4.3).

Algorithm.

- Compute $r = \max_{i\in V^c} \dim(A_{\gamma_i\gamma_i})$ and, for each clique representative $j \in V^c$, a realization $\begin{bmatrix} U_j^T & V_j^T \end{bmatrix}^T$ of the EDM

$$\begin{bmatrix} A_{\nu_j\nu_j} & A_{\nu_j\alpha_j} \\ A_{\alpha_j\nu_j} & A_{\alpha_j\alpha_j} \end{bmatrix} \tag{4.4}$$

  with U_j of size $|\nu_j| \times r$ and V_j of size $|\alpha_j| \times r$. If j is not the root of the tree, we choose a realization that satisfies $e_1^T V_j = 0$.
- Enumerate the clique representatives $j \in V^c$ in an inverse topological order. If j is the root, define $Y_{\nu_j} = U_j$. Otherwise, compute an $r \times r$ orthogonal matrix Q such that $(I - \mathbf{1}e_1^T)Y_{\alpha_j} = V_j Q$, and define

$$Y_{\nu_j} = U_j Q + \mathbf{1}e_1^T Y_{\alpha_j}. \tag{4.5}$$

The realization of (4.4) is computed by factorizing

$$-\frac{1}{2}(I - \mathbf{1}\mu^T)\begin{bmatrix} A_{\nu_j\nu_j} & A_{\nu_j\alpha_j} \\ A_{\alpha_j\nu_j} & A_{\alpha_j\alpha_j} \end{bmatrix}(I - \mu\mathbf{1}^T) = \begin{bmatrix} U_j \\ V_j \end{bmatrix}\begin{bmatrix} U_j \\ V_j \end{bmatrix}^T,$$

with $\mu = e_{|\nu_j|+1}$ if j is not the root, and otherwise an arbitrary μ that satisfies $\mathbf{1}^T\mu = 1$.

To show the correctness of the algorithm, we verify that YY^T is an EDM completion of A. At step j we modify block Y_{ν_j}. Since the cliques are enumerated in an inverse topological order, the block Y_{α_j} has already been computed in earlier steps. It is sufficient to show that after $j \in V^c$ is visited, the matrix

$$\begin{bmatrix} Y_{\nu_j} \\ Y_{\alpha_j} \end{bmatrix}$$

is a realization of $A_{\gamma_j\gamma_j}$. The matrices V_j and Y_{α_j} are two realizations of $A_{\alpha_j\alpha_j}$, and the first row of V_j is zero by construction ($e_1^T V_j = 0$). If we translate the rows of Y_{α_j} to make the first row zero, then, from Lemmas 3.1 and 4.1, the two realizations must be identical up to a right multiplication with an orthogonal matrix. Hence, there exists an orthogonal Q with

$$V_j Q = (I - \mathbf{1}e_1^T)Y_{\alpha_j},$$

as needed in (4.5). The constructed matrix

$$\begin{bmatrix} Y_{\nu_j} \\ Y_{\alpha_j} \end{bmatrix} = \begin{bmatrix} U_j Q + \mathbf{1}e_1^T Y_{\alpha_j} \\ Y_{\alpha_j} \end{bmatrix} = \begin{bmatrix} U_j \\ V_j \end{bmatrix} Q + \mathbf{1}e_1^T Y_{\alpha_j}$$

is a realization of $A_{\gamma_j\gamma_j}$ because, by construction, $[U_j^T \ V_j^T]^T$ is a realization of $A_{\gamma_j\gamma_j}$.

5. Posterior Rounding for the Semidefinite Relaxation of OPF

The AC optimal power flow (OPF) problem [32, 33] is to find a cost-optimal operating point of a power distribution network that consists of a set of power buses and a network of transmission lines. The general OPF problem is a difficult nonconvex optimization problem. Since its introduction in 1962 [32], several different formulations have been proposed [34, 36, 37, 38, 35], and recently, semidefinite relaxation (SDR) techniques for OPF have become an active research area [39, 40, 41, 42]. Semidefinite relaxations provide lower bounds for the optimal value of the OPF problem, and in some cases, the global optimum [43, 40, 44]. The computational cost of SDR, however, grows rapidly with the size of the power system. Thus, solving the SDR of a large-scale OPF problem is often impractical, mainly due to the large, dense matrix variable in the SDR formulation. To this end, the sparse structure of the power network has been extensively exploited to reduce the computational cost of solving the SDR [13, 45, 46, 47, 48, 49]. These methods solve the SDR as a sparse SDP (1.7) and yield a sparse, PSD completable solution.

We describe in this section a posterior rounding technique to obtain a low-rank solution for the SDR of the OPF problem. We first solve the semidefinite relaxation of the OPF problem and obtain a positive semidefinite solution $X^\star$. Then a minimum-rank PSD completion of $\Pi_E(X^\star)$ can be constructed via Algorithm 1, where E is the chordal sparsity pattern for the OPF problem. The completed matrix is by construction optimal for the semidefinite relaxation of OPF, and may have a smaller rank than the SDP solution computed by a general-purpose solver.

The rest of this section is organized as follows. In Section 5.1 we briefly describe the power flow model and the OPF problem. Section 5.2 reformulates the OPF problem and describes

the semidefinite relaxation. In Section 5.3 we discuss the posterior rounding technique, and Section 5.4 contains results of numerical experiments.

5.1. **Power flow model.** The power system model consists of a network of power buses (nodes). We denote the set of power buses by $\mathcal{N}$ (with $|\mathcal{N}| = n$), and the set of transmission line (edges) by $\mathcal{L} \subset \mathcal{N} \times \mathcal{N}$, *i.e.*, $(i, j) \in \mathcal{L}$ if there is a transmission line from node i to node j. Transmission lines may not be symmetric, and thus we model the network graph as a directed graph. We denote $i \sim j$ if $(i, j) \in \mathcal{L}$ or $(j, i) \in \mathcal{L}$.

The optimal power flow problem can be formulated as a nonconvex quadratically constrained quadratic program (QCQP)

$$\begin{aligned}
\text{minimize} \quad & \sum_{(i,j)\in\mathcal{L}} f(v_i, v_j) && && \text{(5.1a)}\\
\text{subject to} \quad & p_{ij}^2 + q_{ij}^2 \le S_{ij}^2, && (i,j) \in \mathcal{L} && \text{(5.1b)}\\
& P_i^{\min} \le \sum_{(i,j)\in\mathcal{L}} p_{ij} \le P_i^{\max}, && i \in \mathcal{N} && \text{(5.1c)}\\
& Q_i^{\min} \le \sum_{(i,j)\in\mathcal{L}} q_{ij} \le Q_i^{\max}, && i \in \mathcal{N} && \text{(5.1d)}\\
& V_i^{\min} \le |v_i| \le V_i^{\max}, && i \in \mathcal{N}, && \text{(5.1e)}\\
& p_{ij} + \hat{\jmath} q_{ij} = v_i(v_i^* - v_j^*) y_{ij}^*, && (i,j) \in \mathcal{L} && \text{(5.1f)}
\end{aligned}$$

where the optimization variables are the real and reactive power flows p_{ij}, q_{ij} over the transmission line $(i, j) \in \mathcal{L}$, and the complex voltage v_i at node i. The positive scalars $P_i^{\min}$, $P_i^{\max}$, $Q_i^{\min}$, $Q_i^{\max}$, $V_i^{\min}$, $V_i^{\max}$, and S_{ij} are given, y_{ij} is the conductivity of $(i, j) \in \mathcal{L}$, and $\hat{\jmath} = \sqrt{-1}$ is the imaginary unit. The objective function consists of fuel cost functions f_g of generator $g \in \mathcal{G}$, and can be any so-called semidefinite representable convex function. Here we model the power loss f as a convex quadratic function of the form

$$f(v_i, v_j) = g_{ij}|v_i - v_j|^2 \tag{5.2}$$

where g_{ij} is the conductance of the transmission line $(i, j) \in \mathcal{L}$. Other choices of semidefinite representable convex functions are also available; see, for example, [50]. The constraints (5.1b) are on the capacity of transmission lines, (5.1c)–(5.1d) are constraints on real and reactive power flows, (5.1e) are voltage magnitude constraints, and (5.1f) describe the relation between power flows and voltages and are derived from the admittance–impedance relation.

5.2. **Reformulation and semidefinite relaxation.** The nonconvexity of (5.1) is due to the equality constraints (5.1f) as well as the inequality constraints $|v_i| \ge V_i^{\min}$. To formulate a tractable convex relaxation for (5.1), we first introduce the variable $X = vv^H$ and replace the constraints (5.1e) and (5.1f) by

$$(V_i^{\min})^2 \le X_{ii} \le (V_i^{\max})^2,\ i \in \mathcal{N}, \quad p_{ij} + \hat{\jmath} q_{ij} = (X_{ii} - X_{ij}) y_{ij}^*,\ (i,j) \in \mathcal{L}, \quad \mathbf{rank}\, X = 1.$$

The first two sets of constraints are now linear inequality and equality constraints, and the only nonconvex rank constraint can be relaxed to a positive semidefinite cone constraint $X \succeq 0$. Thus

we obtain a convex semidefinite relaxation (SDR) for the OPF problem

$$\begin{array}{lll} \text{minimize} & \tilde{f}(X) = \displaystyle\sum_{(i,j)\in\mathcal{L}} g_{ij}(X_{ii} + X_{jj} - 2X_{ij}) & \\ \text{subject to} & p_{ij}^2 + q_{ij}^2 \le S_{ij}^2, & (i,j) \in \mathcal{L} \\ & P_i^{\min} \le \displaystyle\sum_{(i,j)\in\mathcal{L}} p_{ij} \le P_i^{\max}, & i \in \mathcal{N} \\ & Q_i^{\min} \le \displaystyle\sum_{(i,j)\in\mathcal{L}} q_{ij} \le Q_i^{\max}, & i \in \mathcal{N} \\ & (V_i^{\min})^2 \le X_{ii} \le (V_i^{\max})^2, & i \in \mathcal{N} \\ & p_{ij} + \hat{\jmath} q_{ij} = (X_{ii} - X_{ij}) y_{ij}^*, & (i,j) \in \mathcal{L} \\ & X \succeq 0. & \end{array} \tag{5.3}$$

The objective function $\tilde{f}(X)$ is transformed from (5.2). The optimal value of (5.3) provides a lower bound on that of (5.1), and if the optimal solution $X^\star$ satisfies $\mathbf{rank}(X^\star) = 1$, the SDR (5.3) is exact. In this case, we can obtain a global optimal solution for (5.1) by computing a rank-one factorization $X^\star = uu^H$. Conditions for exactness have been extensively studied; see, for example, [52, 41, 51]. If $\mathbf{rank}(X^\star) > 1$, the optimal solution $X^\star$ is not feasible for (5.1) and only provides a lower bound for the optimal value of the original nonconvex problem (5.1). But it can still provide an approximation of optimal powers and voltage magnitudes. A rank-one approximation of $X^\star$ can be computed as

$$\widehat{X} = \lambda_1 u_1 u_1^H,$$

where λ_1 is the largest eigenvalue of $X^\star$, and u_1 is the principal eigenvector. If the numerical rank of $X^\star$ is not much larger than one and the largest eigenvalue is substantially larger than the others, then the above heuristic gives a solution close to the ideal outcome and the principal eigenvector can serve as a good approximation of the optimal complex voltages.

5.3. **Posterior rounding.** In the SDR (5.3), the objective and all the constraints are linear in the matrix variable X except for the positive semidefinite cone constraint. Moreover, all the linear constraints in X, as well as the objective, involve only the elements X_{ij} with $(i,j) \in \mathcal{L}$ or $(j,i) \in \mathcal{L}$. These elements form a (symmetric) sparsity pattern E, and without loss of generality, we can assume E is a chordal sparsity pattern. The resulting SDP can be solved efficiently via a variety of algorithms [13, 45, 53, 14, 54, 48, 49]. If $X^\star$ is an optimal solution, any PSD completion of $\Pi_E(X^\star)$ is also optimal. For semidefinite relaxations of OPF, the minimum-rank completion is of special interest. We denote the minimum-rank PSD completion of $\Pi_E(X^\star)$ by $X^\bullet$. If $X^\bullet$ has rank one, *i.e.*, $X^\bullet = ww^H$, we can construct from it an optimal solution w of the OPF problem. Even when $\mathbf{rank}(X^\bullet) > 1$, the principal eigenvector u_1 of $X^\bullet$ can still be used as an approximation of the globally optimal voltages. Replacing the solution $X^\star$ computed by any SDP solver with the minimum-rank PSD completion of $\Pi_E(X^\star)$ can therefore be interpreted as a *posterior rounding* step to find an optimal solution of lower rank than $X^\star$.

5.4. **Numerical experiments.** In this section we evaluate the performance of the posterior rounding technique applied to the semidefinite relaxation (5.3) of OPF. The experiments are based on the benchmark problems from the MATPOWER package [55], and the Python library CHOMPACK [56] for chordal matrix computations and, in particular, its implementation of Algorithm 1.

Table 1 lists the test cases along with relevant problem dimensions. The value $n = |\mathcal{N}|$ is

Case	$n = \lvert\mathcal{N}\rvert$	$\lvert\mathcal{L}\rvert$	$\lvert\mathcal{G}\rvert$
IEEE-118	118	186	0
IEEE-300	300	409	0
2383wp	2383	2896	92
2736sp	2736	3269	118
2737sp	2737	269	165
2746wop	2746	3307	346
2746wp	2746	3279	352
3012wp	3012	3572	7
3120sp	3120	3693	9
3375wp	3375	3693	25
89pegase	89	210	12
1354pegase	1354	1991	260
2869pegase	2869	4582	510
1888rte	1888	2531	297
1951rte	1951	2596	391
2848rte	2848	3776	547
2868rte	2868	3808	599
6468rte	6468	9000	1295

TABLE 1. Test cases and problem dimensions.

the number of power buses in the network while the value $|\mathcal{L}|$ is the number of transmission lines. The number of generators $|\mathcal{G}|$ is listed in the last column of the table. In the cases where the underlying network is non-chordal, we construct a chordal extension with the AMD reordering [57]. In addition, following the convention in [55, 58, 45], we eliminate transmission line flow constraints in (5.1b) that are not active at the local optimal solution provided by [55].

For each test case, we solve the SDR (5.3) using SeDuMi 1.3 with tolerance 10^{-7} and denote the optimal solution by $X^\star$. Although we explicitly build the complex-valued SDR, we cast the problem as a real-valued problem before passing it to SeDuMi. The minimum-rank PSD completion $X^\bullet$ of $\Pi_E(X^\star)$ is then constructed via Algorithm 1. Note that the problems solved by SeDuMi are real-valued and the computed solution is then transformed back into the complex form. Rank-one solutions of the complex SDP correspond to rank-two solutions of the equivalent real SDP.

The ratio between the largest and the other eigenvalues is used to compute the numerical rank of the solution. When we compute the numerical rank of a matrix, the eigenvalues that are below a certain ratio of the largest one $\lambda_{\max}$ are considered to be zero. In particular, a matrix has numerical rank one if the ratio between the largest and the second largest eigenvalue is sufficiently large. In the experiments, we compute the numerical rank of the solutions $X^\star$ and $X^\bullet$ with the tolerance ϵ equal to $10^{-4}\sqrt{n}$, $10^{-5}\sqrt{n}$, and $10^{-6}\sqrt{n}$. Eigenvalues smaller than $\epsilon\lambda_{\max}$ are considered to be zero.

Numerical results. Table 2 shows the numerical ranks of $X^\star$ and $X^\bullet$ for different tolerances. Overall, the posterior rounding process via Algorithm 1 provides a solution of SDR with rank

			$\epsilon = 10^{-4}\sqrt{n}$		$\epsilon = 10^{-5}\sqrt{n}$		$\epsilon = 10^{-6}\sqrt{n}$	
Case	n	max. clique	**rank**$(X^\star)$	**rank**$(X^\bullet)$	**rank**$(X^\star)$	**rank**$(X^\bullet)$	**rank**$(X^\star)$	**rank**$(X^\bullet)$
IEEE-118	118	20	1	1	1	1	1	1
IEEE-300	300	17	5	1	5	1	36	1
2383wp	2383	31	13	1	17	1	19	3
2736sp	2736	30	1	1	1	1	14	8
2737sop	2737	29	1	1	43	1	87	9
2746wop	2746	30	1	1	32	1	76	11
2746wp	2746	31	1	1	1	1	268	17
3012wp	3012	32	281	5	346	13	578	17
3120sp	3120	32	445	32	572	32	761	32
3375wp	3375	33	442	19	451	19	518	33
89pegase	89	12	7	1	17	5	19	6
1354pegase	1354	19	97	3	111	7	124	19
2869pegase	2869	29	101	13	181	15	199	19
1888rte	1888	16	197	1	251	1	271	3
1951rte	1951	28	23	1	71	1	135	5
2848rte	2848	35	87	1	133	1	210	3
2868rte	2868	31	133	7	255	16	301	21
6468rte	6468	33	214	7	356	11	456	33

TABLE 2. Numerical rank of computed SDP solution $X^\star$ and minimum-rank PSD completion $X^\bullet$ of $\Pi_E(X^\star)$ for different tolerances.

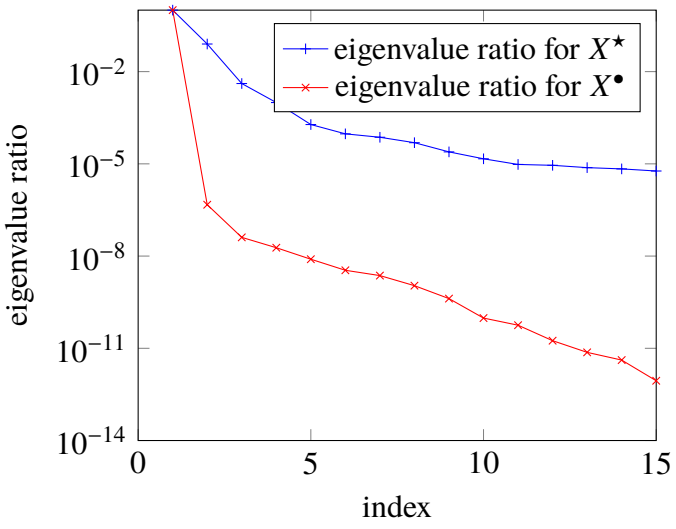

FIGURE 2. Eigenvalue ratio for case IEEE300.

lower than that achieved by general-purpose interior-point solver SeDuMi. In some cases, the improvement is significant and the completed matrix $X^\bullet$ has rank one. On the other hand, the numerical rank depends on the tolerance we use. For example, in the case 2736sp, the numerical rank is one when $\epsilon = 10^{-5}\sqrt{n}$ and becomes eight when the threshold is tightened.

The difference in numerical rank between the two solutions is illustrated in Figure 2. The figure plots the eigenvalue ratio $\lambda_i/\lambda_{\max}$ of the two PSD matrices $X^\star$ and $X^\bullet$, in the test case

			MOSEK 8		SeDuMi v1.3		SDPT3 v4.0	
Case	n	max. clique	**rank**($X^\star$)	**rank**($X^\bullet$)	**rank**($X^\star$)	**rank**($X^\bullet$)	**rank**($X^\star$)	**rank**($X^\bullet$)
IEEE-118	118	20	1	1	1	1	1	1
IEEE-300	300	17	5	1	5	1	5	1
2383wp	2383	31	17	1	17	1	17	1
2736sp	2736	30	1	1	1	1	1	1
2737sop	2737	29	44	1	43	1	43	1
2746wop	2746	30	32	1	32	1	32	1
2746wp	2746	31	1	1	1	1	1	1
3012wp	3012	32	346	13	346	13	337	17
3120sp	3120	32	514	27	572	32	519	27
3375wp	3375	33	451	19	451	19	454	21
89pegase	89	12	19	5	17	5	17	5
1354pegase	1354	19	123	7	111	7	93	8
2869pegase	2869	29	183	14	181	15	167	13
1888rte	1888	16	175	15	251	1	175	15
1951rte	1951	28	71	1	71	1	70	1
2848rte	2848	35	142	1	133	1	133	1
2868rte	2868	31	255	16	255	16	223	13
6468rte	6468	33	356	11	356	11	751	13

TABLE 3. Numerical rank results with different solvers ($\epsilon = 10^{-5}\sqrt{n}$).

IEEE300. For the completed matrix $X^\bullet$, eigenvalues λ_i for $i \geq 2$ are small compared to $\lambda_{\max}(X^\bullet)$ while for the SDR solution $X^\star$ the eigenvalues decay more slowly.

Table 3 shows the numerical ranks of the matrices $X^\star$ and $X^\bullet$, computed by three different solvers, SeDuMi, SDPT3 and MOSEK. Here, we found it helpful to omit inactive line constraints in SeDuMi and SDPT3, but this process was not necessary with MOSEK. The numerical rank is calculated in the method mentioned above with ratio tolerance $\epsilon = 10^{-5}\sqrt{n}$. Results are slightly different for the three solvers, but within a small variation. We can also see that the posterior rounding process generates favorable low-rank results in all the three solvers, and the ranks of the completed matrix do not vary too much in different solvers.

6. Conclusions

We described algorithms for two matrix completion problems with chordal sparsity patterns: the minimum-rank positive semidefinite (PSD) completion, and the minimum-dimension Euclidean distance matrix (EDM) completion. The algorithms use efficient recursions over the clique tree associated with the chordal sparsity pattern. As an application, we investigated the use of the minimum-rank PSD completion algorithm as a posterior rounding step for semidefinite programs (SDPs). If the optimal solution of the SDP is not unique, the rounding step allows us to replace the computed solution by an optimal solution of lower rank. Numerical experiments with SDP relaxations of the optimal power flow (OPF) problem show that the rounding step often substantially reduces the rank of the solution of the semidefinite relaxation, and thus yields better sub-optimal solutions and sometimes optimal solutions for the nonconvex OPF problem. It will be of interest to apply the same techniques in other applications of semidefinite and EDM optimization, where low rank and low embedding dimension are important. Another topic for further research is the development of efficient algorithms to find the nearest matrix with a

low-rank PSD completion or low-dimension EDM completion. Some results in this direction can be found in [59].

References

[1] R. Grone, C. R. Johnson, E. M. Sá, H. Wolkowicz, Positive definite completions of partial Hermitian matrices, Linear Algebra Appl. 58 (1984), 109–124.

[2] M. Bakonyim C. R. Johnson, The Euclidean distance matrix completion problem, SIAM J. Matrix Anal. Appl. 16 (1995), 646–654.

[3] A. Alfakih, H. Wolkowicz, Matrix completion problems, In: H. Wolkowicz, R. Saigal, L. Vandenberghe, (ed.) Handbook of Semidefinite Programming. Theory, Algorithms, and Applications, chapter 18, pages 533–545, Kluwer Academic Publishers, 2000.

[4] M. Bakonyi, H. J. Woerdeman, Matrix Completions, Moments, and Sums of Hermitian Squares, Princeton University Press, 2011.

[5] C. R. Johnson, Matrix completion problems: A survey, In: m Matrix Theory and Applications, volume 40, pages 171–198, 1990.

[6] N. Krislock, H. Wolkowicz, Euclidean distance matrices and applications, In: M. F. Anjos and J. B. Lasserre, (ed.) Handbook on Semidefinite, Conic and Polynomial Optimization, chapter 30, pages 879–914. Springer, 2012.

[7] M. Laurent, A connection between positive semidefinite and Euclidean distance matrix completion problems, Linear Algebra Appl. 273 (1998), 9–22.

[8] M. Laurent, A tour d'horizon on positive semidefinite and Euclidean distance matrix completion problems, In: P. M. Pardalos and H. Wolkowicz, (ed.) Topics in Semidefinite and Interior-Point Methods, volume 18 of Fields Institute Communications, pages 51–76. The American Mathematical Society, 1998.

[9] L. Vandenberghe, M. S. Andersen, Chordal graphs and semidefinite optimization, Foundations and Trends in Optimization, 1 (2015), 241–433.

[10] J. Dancis, Positive semidefinite completions of partial Hermitian matrices, Linear Algebra Appl. 175 (1992), 91–114.

[11] M. S. Andersen, J. Dahl, L. Vandenberghe, Logarithmic barriers for sparse matrix cones, Optim. Meth. Soft. 28 (2013), 396–423.

[12] K. Fujisawa, M. Kojima, K. Nakata, Exploiting sparsity in primal–dual interior-point methods for semidefinite programming, Math. Program. 79 (1997), 235–253.

[13] M. S. Andersen, J. Dahl, L. Vandenberghe, Implementation of nonsymmetric interior-point methods for linear optimization over sparse matrix cones, Math. Program. Comput. 2 (2010), 167–201.

[14] M. Fukuda, M. Kojima, K. Murota, K. Nakata, Exploiting sparsity in semidefinite programming via matrix completion I: general framework, SIAM J. Optim. 11 (2001), 647–674.

[15] X. Jiang, L. Vandenberghe, Bregman primal–dual first-order method and applications to sparse semidefinite programming, Comput. Optim. Appl. 81 (2022), 127–159.

[16] R. Y. Zhang, J. Lavaei, Sparse semidefinite programs with guaranteed near-linear time complexity via dualized clique tree conversion, Math. Program. 188 (2021), 351–393.

[17] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, Chordal and factor-width decompositions for scalable semidefinite and polynomial optimization, Ann. Rev. Control 52 (2021), 243–279.

[18] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, A. Wynn, Chordal decomposition in operator-splitting methods for sparse semidefinite programs, Math. Program. 180 (2020), 489–532.

[19] J. R. S. Blair, B. Peyton, An introduction to chordal graphs and clique trees, In: Graph theory and sparse matrix computation, pp. 1–29. Springer, 1993.

[20] D. R. Fulkerson, O. Gross, Incidence matrices and interval graphs, Pacific J. Math. 15 (1965), 835–855.

[21] P. Buneman, A characterisation of rigid circuit graphs, Discrete Math. 9 (1974) 205–212.

[22] F. Gavril, The intersection graphs of subtrees in trees are exactly the chordal graphs, J. Combin. Theory Ser. B 16 (1974), 47–56.

[23] A. Rantzer, On the Kalman–Yakubovich–Popov lemma, Systems Control Lett. 28 (1996), 7–10.

[24] S. Burer, R. D. C. Monteiro, Local minima and convergence in low-rank semidefinite programming, Math. Program. 103 (2005), 427–444.
[25] N. Krislock, Semidefinite facial reduction for low-rank Euclidean distance matrix completion, PhD thesis, University of Waterloo, 2010.
[26] N. J. Higham, Computing the polar decomposition—with applications, SIAM J. Sci. Statistical Comput. 7 (1986), 1160–1174.
[27] R. A. Horn, C. R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 2012.
[28] M. T. Heath, A. J. Laub, C. C. Paige, R. C. Ward, Computing the singular value decomposition of a product of two matrices, SIAM J. Sci. Statistical Comput. 7 (1986), 1147–1159.
[29] A. W. Bojanczyk, L. M. Ewerbring, F. T. Luk, P. Van Dooren, An accurate product SVD algorithm, Signal Process. 25 (1991), 189–201.
[30] I. J. Schoenberg, Remarks to Maurice Fréchet's article "Sur la définition axiomatique d'une classe d'espaces vectoriels distanciés applicables vectoriellement sur l'espace de Hilbert". Ann. Math. 36 (1935), 724–732.
[31] I. J. Schoenberg, Metric spaces and positive definite functions, Trans. Amer. Math. Sco. 44 (1938), 522–536.
[32] J. A. Momoh, M. E. El-Hawary, R. Adapa, Contribution to the economic dispatch problem, Bulletin de la Societe Francoise des Electriciens, 8 (1962), 431–447.
[33] J. A. Taylor, Convex Optimization of Power Systems, Cambridge University Press, Cambridge, 2015.
[34] M. Huneault, F. D. Galiana, A survey of the optimal power flow literature, IEEE Trans. Power. Systems 6 (1991), 762–770.
[35] J. A. Momoh, Electric Power System Applications of Optimization, Power Engineering (Willis). CRC Press, second edition, 2017.
[36] J. A. Momoh, R. Adapa, M. E. El-Hawary, A review of selected optimal power flow literature to 1993. I. nonlinear and quadratic programming approaches, IEEE Trans. Power Systems 14 (1999), 96–104.
[37] J. A. Momoh, M. E. El-Hawary, R. Adapa, A review of selected optimal power flow literature to 1993. II. Newton, linear programming and interior point methods, IEEE Trans. Power Systems, 14 (1999), 105–111.
[38] J. Zhu, Optimization of Power System Operation, John Wiley & Sons, Ltd, 2015.
[39] X. Bai, H. Wei, K. Fujisawa, Y. Wang, Semidefinite programming for optimal power flow problems, Int. J. Electrical Power Energy Systems 30 (2008), 383–392.
[40] J. Lavaei, S. H. Low, Convexification of optimal power flow problem, In: 48th Annual Allerton Conference on Communication, Control, and Computing, pp. 223–232, 2010.
[41] J. Lavaei, S. H. Low, Zero duality gap in optimal power flow problem, IEEE Trans. Power Sys. 27 (2012), 92–107.
[42] D. K. Molzahn, I. A. Hiskens, A survey of relaxations and approximations of the power flow equations, Foundations and Trends® in Electric Energy Systems, 4 (2019), 1–221.
[43] S. Bose, D. F. Gayme, S. Low, K. M. Chandy, Optimal power flow over tree networks, In: 49th Annual Allerton Conference on Communication, Control, and Computing, pp. 1342–1348, 2011.
[44] J. Lavaei, D. Tse, B. Zhang, Geometry of power flows and optimization in distribution networks, IEEE Trans. Power Systems 29 (2014), 572–583.
[45] M. S. Andersen, A. Hansson, L. Vandenberghe, Reduced-complexity semidefinite relaxations of optimal power flow problems, IEEE Trans. Power Systems 29 (2014), 1855–1863.
[46] S. J. Benson, Y. Ye, Algorithm 875: DSDP5—Software for semidefinite programming, ACM Trans. Math. Software 34 (2008), 16.
[47] R. A. Jabr, Exploiting sparsity in SDP relaxations of the OPF problem, IEEE Trans. Power Systems 27 (2012), 1138–1139.
[48] A. Y. S. Lam, B. Zhang, D. Tse, Distributed algorithms for optimal power flow problem, In: CDC, 2012.
[49] R. Madani, A. Kalbat, J. Lavaei, ADMM for sparse semidefinite programming with applications to optimal power flow problem, In: 54th IEEE Conference on Decision and Control (CDC), pp. 5932–5939, 2015.
[50] A. Ben-Tal, A. Nemirovski, Lectures on Modern Convex Optimization. Analysis, Algorithms, and Engineering Applications, SIAM, 2001.
[51] L. Gan, N. Li, U. Topcu, S. H. Low, Exact convex relaxation of optimal power flow in radial networks, IEEE Trans. Auto. Control 60 (2015), 72–87.

[52] S. H. Low, Convex relaxation of optimal power flow Part II: Exactness, IEEE Trans. Control Netw. Syst. 1 (2014), 177–189.

[53] A. Eltved, J. Dahl, M. S. Andersen, On the robustness and scalability of semidefinite relaxation for optimal power flow problems, Optim. Eng. 21 (2020), 375–392.

[54] S. Kim, M. Kojima, M. Mevissen, M. Yamashita, Exploiting sparsity in linear and nonlinear matrix inequalities via positive semidefinite matrix completion, Math. Program. 129 (2011), 33–68.

[55] R. D. Zimmerman, C. E. Murillo-Sanchez, R. J. Thomas, MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education, IEEE Trans. Power Systems 26 (2011), 12–19.

[56] M. S. Andersen, L. Vandenberghe, Chompack: Library for chordal matrix computations, 2018. Available at `github.com/cvxopt/chompack`.

[57] P. Amestoy, T. Davis, I. Duff, An approximate minimum degree ordering algorithm, SIAM J. Matrix Anal. Appl. 17 (1996), 886–905.

[58] D. K. Molzahn, J. T. Holzer, B. C. Lesieutre, C. L. DeMarco, Implementation of a large-scale optimal power flow solver based on semidefinite programming, IEEE Trans. Power Systems 28 (2013), 3987–3998.

[59] S. Al-Homidan, H. Wolkowicz, Approximate and exact completion problems for Euclidean distance matrices using semidefinite programming, Linear Algebra Appl. 406 (2005), 109–141.