

# Chordal Decomposition in Rank Minimized Semidefinite Programs with Applications to Subspace Clustering

Jared Miller<sup>1</sup>, Yang Zheng<sup>2</sup>, Biel Roig-Solvas<sup>1</sup>, Mario Sznaier<sup>1</sup>, Antonis Papachristodoulou<sup>3</sup>

**Abstract**—Semidefinite programs (SDPs) often arise in relaxations of some NP-hard problems, and if the solution of the SDP obeys certain rank constraints, the relaxation will be tight. Decomposition methods based on chordal sparsity have already been applied to speed up the solution of sparse SDPs, but methods for dealing with rank constraints are underdeveloped. This paper leverages a minimum rank completion result to decompose the rank constraint on a single large matrix into multiple rank constraints on a set of smaller matrices. The reweighted heuristic is used as a proxy for rank, and the specific form of the heuristic preserves the sparsity pattern between iterations. Implementations of rank-minimized SDPs through interior-point and first-order algorithms are discussed. The problems of maxcut and subspace clustering are used to demonstrate the computational improvement of the proposed method.

## I. INTRODUCTION

Semidefinite programs (SDPs) are a class of convex optimization problems that minimize a linear functional of a positive semidefinite (PSD) matrix under linear constraints. SDPs often arise as relaxations of some NP-hard problems, such as binary optimization, optimal power flow, and maxcut problems [1], [2]. In addition, polynomial optimization is generically NP-hard, and Lasserre has demonstrated a hierarchy of SDPs that can approximate the true global optimum [3]. Subspace clustering is an NP-hard algorithm that groups points originating from a union of subspaces and admits an SDP relaxation [4].

In each of these cases, the SDP relaxation is able to return the same optimum as the original NP-hard problem if the solution satisfies certain rank conditions. For example, maxcut problems and optimal power flow problems require rank one solutions [5]. Global optima can be extracted from the Lasserre hierarchy only if certain rank conditions are met [6]. Subspace clustering is a polynomial optimization problem that requires a rank one solution [4]. All of examples desire a low rank solution, and the path to find this solution can be formulated as a rank-constrained SDP:

$$\begin{aligned} \min_X \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i, i = 1, \dots, m, \\ & X \in \mathbb{S}_+^n, \quad \text{rank}(X) \leq t, \end{aligned} \quad (1)$$

<sup>1</sup>J. Miller, B. Roig-Solvas and M. Sznaier are with the ECE Department, Northeastern University, Boston, MA 02115. (Emails: miller.jare@husky.neu.edu, biel@ece.neu.edu, msznaier@coe.neu.edu).

<sup>2</sup>Y. Zheng was with the Department of Engineering Science, University of Oxford. He is now with the School of Engineering and Applied Sciences, and also with the Harvard Center for Green Buildings and Cities, Harvard University, Cambridge, MA 02138. (Email: yang.zheng@gsd.harvard.edu).

<sup>3</sup>A. Papachristodoulou is with the Department of Engineering Science, University of Oxford, Oxford, UK OX1 3PJ. (Email: antonis@eng.ox.ac.uk).

where  $\langle M, N \rangle = \text{Tr}(M^T N)$  is an inner product and  $C, A_1, \dots, A_m \in \mathbb{S}^n$ ,  $b \in \mathbb{R}^m$ , and  $t \in \mathbb{N}$  are problem data. Throughout this work,  $\mathbb{R}^m$  is the  $m$ -dimensional Euclidean space,  $\mathbb{S}^n$  is the space of  $n \times n$  symmetric matrices and  $\mathbb{S}_+^n$  is the subspace of symmetric PSD matrices.

While the rank-constrained SDP in (1) is in general an NP-hard problem, a great deal of interest has been put in developing tractable rank proxies. Matrix factorization (*i.e.*,  $X = YY^T$ ) is a popular nonconvex method that upper bounds the possible rank by the width of  $Y$ , and Burer-Monteiro results may ensure a global optimality after curvilinear optimization on the low-rank manifold [7], [8]. Variable splitting can also be employed through nonconvex ADMM (*e.g.*,  $X = LR^T$ ,  $L = R$ ) [9], which is vulnerable to the nonconvexity and pathological geometry of the low-rank manifold. Projection onto this manifold in intermediate iterations is computationally tractable and may lead to substantial speedup, but even with convergence guarantees the nonconvexity of the rank constraint may lead to the wrong solution [10]. Several convex relaxations of the rank constraint have been developed, one of the most popular being the nuclear norm. The nuclear norm  $\|X\|_*$  is the sum of the singular values of  $X$ , and  $\|X\|_* = \text{Tr}(X)$  if  $X \in \mathbb{S}_+^n$  [11]. The nuclear norm is the biconjugate of the rank function, and under certain *restricted isometry property* (RIP)/coherence conditions, the nuclear norm-minimized solution of an SDP is equivalent to the rank-minimal optimum [12]. RIP/coherence holds only in a very narrow set of problems, since the nuclear norm on singular values weight all contributions equally. One approach to avoid this is the *weighted nuclear norm* that penalizes each singular value differently [13]. The *reweighted heuristic* is a linearization of log-det, and adds a penalty of  $\text{Tr}(W_t X)$  instead of the normal nuclear norm  $\text{Tr}(X)$ , where  $W_t = (X + \delta I)^{-1}$  updates at each iteration [14]. We note that some recent work on the  $r$ \* norm offers convex relaxations where the parameter  $r$  more directly encourages  $\text{rank}(X^*) \approx r$ , at the expense of more complicated SDP representations [15].

It is known that the complexity of solving SDPs scales in a polynomial time w.r.t. number of variables and constraints, and the addition of rank penalties and heuristics may further increase this complexity. For large-scale instances, exploitation of structure and sparsity is vital to speed up computation. In sparse cases, only a small subset of entries of  $X$  are used in the cost  $C$  and constraints  $A_i$ . All other entries only need to guarantee that  $X$  is PSD. When the sparsity pattern spanned by the used entries is or can be extended to a chordal graph, chordal decomposition theory

can break up a large PSD constraint ( $X \succeq 0$ ) into a set of smaller and coupled PSD constraints ( $X_k \succeq 0$ ) [16]. Taking advantage of this structure rapidly speeds up computational time, as the complexity of optimization problems is related to the tree-width of the underlying graph [17]. Chordal decomposition of SDPs can effectively reduce dimension of sparse problems; see, *e.g.*, [16], [18], [19], [20] for details.

After performing the chordal decomposition and optimizing over a reduced set of variables, there exist multiple methods of matrix completion to generate a valid  $X^*$ . One such choice is the minimum rank completion [21], in which the minimal possible rank of the completion  $X$  is the maximal rank among the blocks  $X_k$ . Numerical rounding on the eigenvalues of  $X_k$  has already been used to reduce rank of  $X$ , but the next step to penalize the rank of  $X_k$  was not considered in the literature [22]. Minimum rank completions over linear matrix inequalities with general graphs has been performed in the context of optimal power flow, but few details were mentioned about the mechanism for penalizing the rank of tree components [23].

In this paper, we combine minimal rank completion and the reweighted heuristic to effectively solve large-scale rank-constrained SDPs. We first use the minimal rank completion result to derive an equivalent reformulation of the rank-constrained SDP (1) with chordal sparsity. One key feature is that the PSD and rank constraints on a single big matrix are equivalently replaced by a set of PSD constraints and rank constraints on multiple smaller matrices. The reweighted heuristic is then used to relax the rank constraints, which simply adds a penalty term to the objective function. We show that the resulting SDP relaxation for each iteration of the reweighted heuristic preserves the sparsity pattern of the original problem. This fact allows us to take advantage of normal chordal decomposition in both interior-point methods [18] and first-order methods [19] to greatly speed up the solution of sparse instances. We apply the proposed method to solve subspace clustering problems, which demonstrates the computational improvements.

The rest of this paper is structured as follows. Section II introduces chordal graphs, sparse matrices, and the minimum rank completion problem. In section III, we present an equivalent reformulation of (1). Implementations in interior-point methods and first-order methods are discussed in section IV. These algorithms are used in section V to solve MaxCut and subspace clustering problems. Section VI concludes the paper.

## II. PRELIMINARIES

In this section, we cover some preliminaries on chordal graphs, sparse PSD matrices, as well as minimum rank completion. For a comprehensive treatment, the interested reader is referred to [16].

### A. Chordal graphs

An undirected graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  is defined by a set of vertices  $\mathcal{V} = \{1, 2, \dots, n\}$  and edges  $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ . A cycle of length  $N$  is a set of nodes  $v_k$  such that  $(v_1, v_2), (v_2, v_3) \dots (v_i, v_i +$

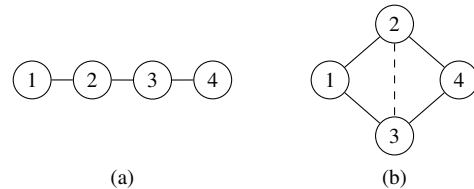


Fig. 1. Examples of chordal graphs: (a) a path graph; (b) a triangulated graph (with dashed edge). Without the dashed edge there is a cycle of length 4 without a chord, so the graph is not chordal.

$1) \dots (v_{N-1}, v_N), (v_N, v_1) \in \mathcal{E}$ , and all vertices  $v_k$  are unique. A chord is an edge that connects two nonconsecutive nodes in a cycle, which therefore breaks a large cycle into smaller cycles, *e.g.*,  $(v_3, v_7)$ . An undirected graph is *chordal* if all cycles of length four or more have at least one chord [16]. The chordal extension  $\mathcal{G}_c(\mathcal{V}_c, \mathcal{E}_c)$  of graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  is a chordal graph  $\mathcal{G}_c$  where  $\mathcal{V} = \mathcal{V}_c$  and  $\mathcal{E} \subseteq \mathcal{E}_c$ , and any non-chordal graph can be extended to be chordal. Finding a chordal extension with a minimal number of added edges is NP-hard, but efficient heuristics exist to give good chordal extensions [24].

A clique  $\mathcal{C}$  with cardinality  $|\mathcal{C}|$  is a subset of vertices in  $\mathcal{V}$  that form a complete subgraph:  $\forall v_i, v_j \in \mathcal{C}, (v_i, v_j) \in \mathcal{E}$ . A maximal clique is a clique not contained inside another clique. Finding the set of all maximal cliques is NP-hard for general graphs, but can be computed on chordal graphs in linear time. Two common chordal graphs are demonstrated in Fig. 1, where the graph in Fig. 1(a) has maximal cliques  $\mathcal{C}_i = \{i, i+1\}$ ,  $i = 1, 2, 3$ , and the graph in Fig. 1(b) has maximal cliques  $\mathcal{C}_1 = \{1, 2, 3\}$  and  $\mathcal{C}_2 = \{2, 3, 4\}$ .

### B. Sparse matrices and chordal decomposition

Considering the SDP (1), graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  arises from stacking up all the  $(C, A_i)$  and treating this as an adjacency matrix. Let  $\mathcal{E}^* = \mathcal{E} \cup \{(i, i), \forall i \in \mathcal{V}\}$  be the edge set with self loops. We define a set of sparse symmetric matrices as:

$$\begin{aligned} \mathbb{S}^n(\mathcal{E}, 0) &= \{X \in \mathbb{S}^n \mid X_{ij} = 0, \forall (i, j) \notin \mathcal{E}^*\}, \\ \mathbb{S}_+^n(\mathcal{E}, 0) &= \mathbb{S}(\mathcal{E}, 0) \cap \mathbb{S}_+^n \end{aligned}$$

$\mathbb{S}_+^n(\mathcal{E}, 0)$  forms a cone of sparse PSD matrices that can be nonzero only at specified entries defined by graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$ . The dual cone  $\mathbb{S}_+^n(\mathcal{E}, 0)^* = \mathbb{S}_+^n(\mathcal{E}, ?)$ , which is the set of matrices that can be completed to be PSD:

$$\begin{aligned} \mathbb{S}_+^n(\mathcal{E}, ?) &= \{X \in \mathbb{S}^n(\mathcal{E}, ?) \mid \exists M \in \mathbb{S}_+^n, \\ &X_{ij} = M_{ij} \forall (i, j) \in \mathcal{E}^* \} \end{aligned}$$

Note that such a completion is not usually unique, as there may be multiple  $M$  associated to each  $X$ . For chordal graphs,  $\mathbb{S}_+^n(\mathcal{E}, ?)$  can be equivalently decomposed into a set of smaller but coupled convex cones:

*Theorem 1 (Grone's Theorem [25]):* Let  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  be a chordal graph with a set of maximal cliques  $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_p\}$ . Then,  $X \in \mathbb{S}_+^n(\mathcal{E}, ?)$  if and only if

$$E_{\mathcal{C}_k} X E_{\mathcal{C}_k}^T \in \mathbb{S}_+^{|\mathcal{C}_k|}, \quad k = 1, \dots, p.$$

In Theorem 1,  $E_{\mathcal{C}_k}$  are 0/1 entry selector matrices that index out components of  $X$  involved in clique  $\mathcal{C}_k$ . Grone's

theorem provides a set equivalence  $\mathbb{S}_+^n(\mathcal{E}, ?) = \prod_{k=1}^p \mathbb{S}_+^{|\mathcal{C}_k|}$  modulo overlaps between cliques, breaking a large PSD cone into a host of smaller PSD cones and equality constraints. This fact underpins the idea of much work that exploits sparsity in large-scale SDPs [16], [18], [19], [20], [26].

### C. Minimum rank completion

Given  $X \in \mathbb{S}_+(\mathcal{E}, ?)$ , many choices of PSD completions are available after determining  $X_k = E_{\mathcal{C}_k} X E_{\mathcal{C}_k}^T$ , two of which are the maximum determinant completion and minimum rank completion. There exists a unique completion with maximum determinant with an explicit formula [27]. Minimum rank completions are not necessarily unique:

*Theorem 2 (Minimum rank completion [21]):* Given a chordal graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  with a set of maximal cliques  $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_p\}$ , for any  $X \in \mathbb{S}_+^n(\mathcal{E}, ?)$ , there exists at least one minimum rank PSD completion, where  $\text{rank}(X) = \max_k \text{rank}(E_{\mathcal{C}_k} X E_{\mathcal{C}_k}^T)$ .

A procedure to perform minimum rank completion is Algorithm 3.1 in [22], which updates a factorization of the completion while proceeding through the elimination tree. We conclude this section with the following example:

$$X = \begin{bmatrix} 3 & 0.5 & ? & 0.25 \\ 0.5 & 2 & 0.75 & ? \\ ? & 0.75 & 1 & ? \\ 0.25 & ? & ? & 5 \end{bmatrix}, \quad (2)$$

where ? denotes elements that need to be filled in. In this case, the sparsity pattern of  $X$  is a 4-length path graph (see Fig. 1), which is chordal with maximal cliques  $\mathcal{C}_1 = \{1, 2\}, \mathcal{C}_2 = \{1, 4\}, \mathcal{C}_3 = \{2, 3\}$ . It is easy to verify that

$$E_{\mathcal{C}_1}^T X E_{\mathcal{C}_1} = \begin{bmatrix} 3 & 0.5 \\ 0.5 & 2 \end{bmatrix} \in \mathbb{S}_+^2,$$

and that the principal submatrices corresponding to  $\mathcal{C}_2$  and  $\mathcal{C}_3$  are PSD. Then, Theorem 1 guarantees that  $X$  is PSD completable. Also, since we have  $\max |\mathcal{C}_i| = 2$ , Theorem 2 guarantees that there exists a PSD completion of rank exactly *two* as follows

$$X^* = \begin{bmatrix} 3 & 0.5 & -1.25 & 0.25 \\ 0.5 & 2 & 0.75 & -3.05 \\ -1.25 & 0.75 & 1 & -1.65 \\ 0.25 & -3.05 & -1.65 & 5 \end{bmatrix}.$$

## III. CHORDAL DECOMPOSITION IN SPARSE SDPs WITH A RANK CONSTRAINT

In this section, we first introduce a chordal decomposition approach in sparse SDPs with a rank constraint by combining Theorems 1 and 2. Then, we discuss the application of nuclear norm relaxations to the decomposed problem.

### A. An equivalent reformulation

Let the rank constrained SDP (1) be sparse with an *aggregate sparsity pattern* described by graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$ , meaning that  $C \in \mathbb{S}^n(\mathcal{E}, 0)$ ,  $A_i \in \mathbb{S}^n(\mathcal{E}, 0)$ ,  $i = 1, \dots, m$ . Throughout this paper, we assume that the aggregate sparsity pattern  $\mathcal{G}$  is chordal (otherwise, a suitable chordal extension can

be found) with a set of maximal cliques  $\mathcal{C}_1, \dots, \mathcal{C}_p$ . Then, combining Grone's theorem (Theorem 1) with the minimum rank completion theorem (Theorem 2), leads to the following observation:

*Proposition 1:* Suppose that Problem (1) is feasible and the problem data has an aggregate sparsity pattern  $\mathcal{G}(\mathcal{V}, \mathcal{E})$ . Then, (1) is equivalent to the following reformulation

$$\begin{aligned} & \min_{X, X_k} \langle C, X \rangle \\ & \text{subject to} \quad \langle A_i, X \rangle = b_i, i = 1, \dots, m, \\ & \quad X_k = E_{\mathcal{C}_k} X E_{\mathcal{C}_k}^T, k = 1, \dots, p, \\ & \quad X_k \in \mathbb{S}_+^{|\mathcal{C}_k|}, k = 1, \dots, p, \\ & \quad \text{rank}(X_k) \leq t, k = 1, \dots, p, \end{aligned} \quad (3)$$

in the sense that (1) and (3) have the same cost value, and their optimal solutions can be mutually recovered.

*Proof:* Thanks to the aggregate sparsity pattern, the cost function and the equality constraints in (1) depend only on the elements  $X_{ij}$  with  $(i, j) \in \mathcal{E}^*$ . The PSD constraint  $X \in \mathbb{S}_+^n$  in (1) can be equivalently replaced by a PSD completable constraint  $X \in \mathbb{S}_+^n(\mathcal{E}, ?)$ .

The rest of proof directly follows the application of Theorems 1 and 2 to (1). We denote the optimal cost values to (1) and (3) as  $f_1^*$  and  $f_2^*$  respectively.

- First, assume we find an optimal solution  $X_1^*$  to (1) with an optimal cost value  $f_1^* = \langle C, X_1^* \rangle$  and  $\text{rank}(X_1^*) \leq t$ . Then, the solution  $X_1^*$  is also a feasible solution to (3). It means that  $f_2^* \leq f_1^*$ .
- Second, assume we find an optimal solution  $X_2^*$  to (3) with an optimal cost value  $f_2^* = \langle C, X_2^* \rangle$ . Then, according to Theorems 1 and 2, we can find a PSD completion  $\hat{X}_2^*$ , where  $\text{rank}(\hat{X}_2^*) = \max_k \text{rank}(E_{\mathcal{C}_k} X_2^* E_{\mathcal{C}_k}^T) \leq t$ . Thus, the PSD completion  $\hat{X}_2^*$  is a feasible solution to (1), indicating that  $f_1^* \leq f_2^*$ .

Combining these facts, we know  $f_1^* = f_2^*$ , and the optimal solutions to (1) and (3) can be recovered from each other. ■

One key feature of problem (3) is that both the PSD and rank constraints are only imposed on multiple small symmetric matrices of smaller dimension rather than on the single large symmetric matrix. The minimum rank completion automatically yields an upper bound on the minimized full matrix rank according to maximum clique size.

### B. Rank relaxations

In general, problem (3) is hard to solve due to the rank constraints. One popular approach is to apply a nuclear norm relaxation [11]. For PSD matrices, we have  $\|X\|_* = \text{Tr}(X)$ . In this paper, we replace the hard rank constraint with a soft reweighted heuristic [14], leading to a standard SDP with chordal sparsity:

$$\begin{aligned} & \min_{X, X_k} \langle C, X \rangle + \sum_{k=1}^p \langle W_k, X_k \rangle \\ & \text{subject to} \quad \langle A_i, X \rangle = b_i, i = 1, \dots, m \\ & \quad X_k = E_{\mathcal{C}_k} X E_{\mathcal{C}_k}^T, \\ & \quad X_k \in \mathbb{S}_+^{|\mathcal{C}_k|}, \forall k = 1, \dots, p, \end{aligned} \quad (4)$$

---

**Algorithm 1** Chordal SDP with Reweighted Heuristic
 

---

```

1: procedure CHORDAL_RANK
2:    $W_k \leftarrow I \quad \forall k = 1, \dots, p$ 
3:   while  $X$  not converged do
4:      $(X, X_k) \leftarrow$  optimum of (4) given  $W_k$ 
5:      $\tilde{W}_k \leftarrow (X_k + \delta I)^{-1}$ 
6:      $W_k \leftarrow \tau_k \tilde{W}_k / \|W_k\|_2$ 

```

---

where  $W_k$  is the weight matrix corresponding to clique  $C_k$  that encourages low rank behavior.

As summarized in Algorithm 1, reweighted heuristic calculates a new set of weights  $W_k$  at each iteration based on the previous  $X_k$ . In general, the inner product between  $W_k$  and  $X_k$  promotes the concentration of energy of  $X_k$  onto the dominant eigenspace of  $X_{k,\text{old}}$  and thus incentivizes the reduction of its rank. Weights  $W_k$  are normalized in Algorithm 1, and  $\tau_k$  is a per clique regularization parameter. Reweighted heuristic is a local linearization of log-det and will converge to a stationary point  $X^*$ , but  $X^*$  will not in general match the log-det penalty [14]. Fig. 2 demonstrates the rank reduction behavior of Algorithm 1 on a Maxcut problem with 1000 vertices (see Section V for details).

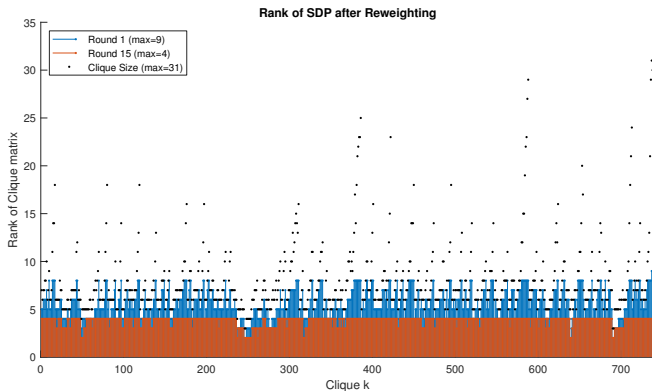


Fig. 2. Example of reweighting heuristic (Algorithm 1) on a Maxcut problem with 1000 vertices. There are 740 cliques with a maximum  $|C^{\max}| = 31$  (black dots). The maximum clique rank starts at 9 (blue), and drops to 4 (orange) after 15 rounds of optimization. See Section V-A for details, seed = 845.

#### IV. ALGORITHM IMPLEMENTATIONS

Problem (4) is convex and ready to solve by existing conic solvers, *e.g.*, SeDuMi or Mosek. However, a naive passing to SeDuMi or Mosek does not scale well to large-scale instances. Interior point methods such as SeDuMi will suffer from additional equality constraints introduced from the chordal decomposition. In this section, we modify Problem (4) to exploit its structure for adaptation in both interior point methods and first order methods.

##### A. Interior-point methods via SparseCoLO

Here, we show that Problem (4) also preserves the sparsity pattern of (1). This can be observed by eliminating the

variables  $X_k$ , which leads to

$$\begin{aligned}
 \min_{X, X_k} \quad & \langle C, X \rangle + \sum_{k=1}^p \langle W_k, E_{C_k} X E_{C_k}^T \rangle \\
 \text{subject to} \quad & \langle A_i, X \rangle = b_i, i = 1, \dots, m, \\
 & X \in \mathbb{S}_+^n(\mathcal{E}, ?).
 \end{aligned} \tag{5}$$

Next,  $E_{C_k}$  can be transferred from the right side of the inner product to the left side to find a weighted cost matrix:

$$\begin{aligned}
 \langle C, X \rangle + \sum_{k=1}^p \langle W_k, E_{C_k} X E_{C_k}^T \rangle \\
 = \langle C, X \rangle + \sum_{k=1}^p \langle E_{C_k}^T W_k E_{C_k}, X \rangle \\
 = \left\langle C + \sum_{k=1}^p E_{C_k}^T W_k E_{C_k}, X \right\rangle \\
 = \langle C + W_C, X \rangle,
 \end{aligned}$$

where  $W_C = \sum_{k=1}^p E_{C_k}^T W_k E_{C_k}$  is the accumulated clique weight. By construction, we know that  $W_C \in \mathbb{S}^n(\mathcal{E}, 0)$ . Problem (4) is then equivalent to

$$\begin{aligned}
 \min_{X, X_k} \quad & \langle C + W_C, X \rangle \\
 \text{subject to} \quad & \langle A_i, X \rangle = b_i, i = 1, \dots, m, \\
 & X \in \mathbb{S}_+^n(\mathcal{E}, ?).
 \end{aligned} \tag{6}$$

It is easy to see that  $C + W_C \in \mathbb{S}^n(\mathcal{E}, 0)$  and  $A_i \in \mathbb{S}^n(\mathcal{E}, 0)$ , indicating that Problem (6) has exactly the same aggregate sparsity pattern as that in (1). This sparsity pattern can be ready to be exploited in SparseCoLO [28]. In particular, SparseCoLO uses range-space and domain-space (r and d) conversion techniques to reduce the number of variables needed to represent the problem while attaining an equivalent optimum [28]. SparseCoLO will perform conversion routines to generate a standard conic form that is suitable for standard interior-point solvers, *e.g.*, SeDuMi. Clique-tree decompositions may be used for r-space and d-space conversions, and SparseCoLO typically tries to merge some small cliques into a big one.  $W_C$  and the minimum rank completion use cliques generated by the chordal decomposition and extension, but the choice is free.

In Algorithm 1, only the cost matrix  $C + W_C$  changes between reweighting iterations, so no repeat conversions are necessary. The output  $X^*$  will generally be full rank from SparseCoLO. The low rank solution can be extracted by indexing out and rounding the cliques  $X_k^*$  and then forming the minimal rank completion  $X_r^*$ .

*Remark 1:* As shown in Proposition 1, Problems (1) and (3) are equivalent. This tightness is lost when conducting rank relaxations. One can also directly apply the reweighted heuristic to Problem (1). Then, one need to compute the inverse of a big matrix to update the weight  $W = (X + \delta I)^{-1}$ . In addition, as the inverse is generically dense, the next iteration's cost  $C + W$  would be dense and the sparsity pattern would be destroyed. Instead, in the proposed method, only the inverses of multiple small

matrices are required and the weight  $W_C$  in (6) preserves the sparsity structure. Reweighted heuristic on problems (1) and (3) produce different weights  $W$  and  $W_C$ , and the resulting optima will not generally match as  $W \neq W_C$ . We note that using  $W$  in the formulation of (1) will still penalize rank, but may settle at a high rank solution as the upper bound on clique size is lost.

### B. Alternating Direction Method of Multipliers (ADMM)

Following [19], [26], we can build an ADMM algorithm to exploit the underlying structure (4) to obtain its solution. The ADMM algorithm is able to take advantage of the variable-split  $X_k = E_{C_k} X E_{C_k}^T$  from Theorem 1 to separate affine and PSD constraints. Each iteration of ADMM has three steps [29]: The affine constraints in  $X$  are handled in step 1, the PSD projection in  $X_k$  is form step 2, and step 3 is the dual ascent on dual variables  $\Lambda_k$  to enforce the variable-split. First order algorithms such as ADMM are vulnerable to slow convergence, but have a relatively low per-iteration cost.

Applying ADMM to (4), we need to solve the following subproblems at each iteration  $t$  (with optional adjustment of  $\rho$  between iterations following [29]):

- Step 1: Solve the following quadratic program (QP)

$$\begin{aligned} \min_X \langle C, X \rangle + \frac{\rho}{2} \sum_{k=1}^p \left\| X_k^{(t)} - E_{C_k} X E_{C_k}^T + \frac{1}{\rho} \Lambda_k^{(t)} \right\|_F^2 \\ \langle A_i, X \rangle = b_i, \quad i = 1, \dots, m. \end{aligned} \quad (7)$$

- Step 2: Project onto  $\mathbb{S}_+^{|C_k|}$  in parallel

$$\begin{aligned} \min_{X_k} \text{Tr}(W_k X_k) + \frac{\rho}{2} \left\| X_k - E_{C_k} X^{(t+1)} E_{C_k}^T + \frac{1}{\rho} \Lambda_k^{(t)} \right\|_F^2 \\ X_k \succeq 0. \end{aligned} \quad (8)$$

- Step 3: Update the multipliers by dual ascent

$$\Lambda_k^{(t+1)} = \Lambda_k^{(t)} + \rho \left( X_k^{(t+1)} - E_{C_k} X^{(t+1)} E_{C_k}^T \right).$$

The QP in step 1 can be solved by vectorizing all the variables. Entry selection on the clique  $C_k$  can be replaced by a matrix  $H_k$  where  $H_k \text{vec}(X) = E_{C_k} X E_{C_k}^T$ . Likewise  $a_i = \text{vec}(A_i)$  is collated into  $A$  and  $b_i$  into  $b$ ,  $c = \text{vec}(C)$ , and  $v_k = \text{vec}(X_k^{(t)} + \frac{1}{\rho} \Lambda_k^{(t)})$ . Step 1 is vectorized into:

$$\begin{aligned} \min_X \langle c, x \rangle + \frac{\rho}{2} \sum_{k=1}^p \|H_k x - v_k\|_2^2 \\ \text{subject to } Ax = b \end{aligned} \quad (9)$$

The KKT system for the QP involves  $x$  and dual variable  $\omega$ :

$$\begin{bmatrix} D & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \omega \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^p H_k^T v_k - c \\ b \end{bmatrix}$$

Since each  $H_k$  is orthonormal,  $H_k^T H_k$  is diagonal and  $D = \sum_{k=1}^p H_k^T H_k$ . The diagonal-offset lends itself nicely to block elimination and precomputed factorization.

Step 2 involves parallel PSD projections:

$$\min_{X_k \succeq 0} \text{Tr}(W_k X_k) + \frac{\rho}{2} \|X_k - V_k\|_F^2$$

This minimization occurs by taking  $M_k = V_k - \frac{1}{\rho} W_k$  and then projecting  $M_k$  onto the PSD cone (setting all negative eigenvalues of  $M_k$  to zero). If  $W_k = I$ , this procedure is singular value thresholding [30].  $X_k$  can be computed through a full eigendecomposition of  $M_k$  or by power methods where power/subspace iteration proceeds until a zero or a negative eigenvalue is hit.

### C. Homogeneous Self-Dual Embedding

Finally, we show that the reweighted heuristic for the proposed decomposition also fits naturally into homogeneous self-dual embedding (HSDE) framework [31]. HSDE combines the primal and dual problems together to allow for the identification of infeasible SDPs. Solutions of chordally-sparse SDPs through HSDE have been already implemented [20], which is based on the general conic formulation introduced in [32]. Each iteration of HSDE is comprised of a large block-sparse linear system call, a projection onto a product of multiple cones, and a dual update step (see [20], [32] for details).

If  $s$  is the concatenation of  $\text{vec}(X_k)$  and  $m$  is the concatenation of  $\text{vec}(W_k)$  across all cliques  $C^k$ , the linear system in HSDE  $v = Qu$  is:

$$\begin{bmatrix} h \\ z \\ r \\ w \\ k \end{bmatrix} = \begin{bmatrix} & & -A^T & -H^T & c \\ & & & I & m \\ A & & & & -b \\ H & -I & & & \\ -c^T & -m^T & b & & \end{bmatrix} \begin{bmatrix} x \\ s \\ y \\ v \\ t \end{bmatrix}$$

The only difference as compared to [20] is that in the rank-unconstrained formulation,  $m = 0$ .  $m$  is the only varying parameter (last column/row), and the large blocks of the linear system remain identical between iterations.

## V. NUMERICAL EXAMPLES

In this section, we demonstrate the performance of Algorithm 1 to solve maxcut and subspace clustering: problems that each involves SDP relaxations with rank-one constraints. In our experiments, we solved the subproblem (4) through a variety of solvers. Sedumi [33], SparseCoLO [28], SDPT3 [34], and CDCS [35] are used for maxcut, and subspace clustering is tested on Mosek [36] using Yalmip [37].

### A. Maxcut

Maxcut is an NP-hard problem that is standard for testing SDPs [1]. Given a graph  $\mathcal{G}(\mathcal{V}, \mathcal{E})$  with  $N$  vertices where edges  $e_{ij} \in \mathcal{V}$  have weight  $w_{ij}$ , vertices in  $v_i \in \mathcal{V}$  are partitioned into two groups ( $x_i = \pm 1$ ). A cut  $T \subsetneq \mathcal{E}$  is a set of edges such that on each edge  $e_{ij} \in T$ :  $x_i \neq x_j$ . The goal of Maxcut is to find a cut with maximal weight:

$$\begin{aligned} \max_x \sum_{(i,j) \in \mathcal{E}} w_{ij} \frac{1 - x_i x_j}{2} \\ \text{subject to } |x_i| = 1. \end{aligned}$$

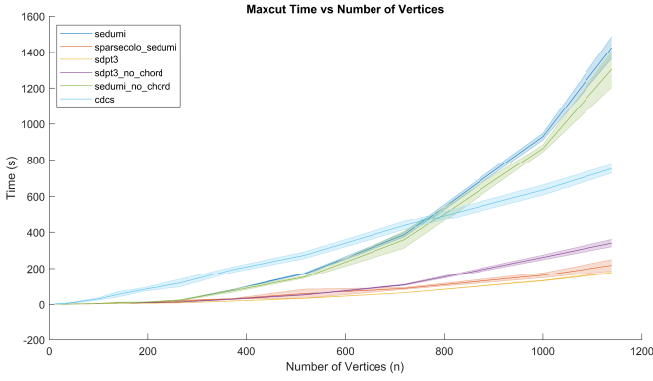


Fig. 3. Execution time of Maxcut SDPs. Error bars are  $\pm$  one standard deviation over 5 iterations. SDPT3 with chordal regularization is fastest.

Edges only contribute to the cost function if  $x_i \neq x_j$ . Maxcut can be equivalently rewritten as a rank-minimized SDP [5]

$$\begin{aligned} \max_X \quad & \frac{1}{2} \langle W, X \rangle \\ \text{subject to} \quad & \text{Diag}(X) = e, \\ & X \in \mathbb{S}_+^N, \quad \text{rank}(X) = 1. \end{aligned}$$

Where  $e$  is the vector of all 1's. If  $\text{rank}(X) = 1$ , by definition  $X = xx^T$ , and the vertex partition  $x$  can be recovered from the solitary nonzero eigenvector. This rank constraint is NP-hard, and methods such as Goemans-Williamson hyperplane rounding based on the SDP relaxation are provably within 0.878 of the optimum cost [38]. In this SDP formulation, the matrices  $A_i$  each have only 1 nonzero entry on the main diagonal, and all other problem structure comes from the Laplacian  $L$ . In the typical case where the graph is sparse, the chordal decomposition can remove redundant variables and provide rank guarantees based on the minimum rank completion.

Data was synthesized through the `maxCutSDP` routine in the `SparseCoLo` experiments section [28]. Between 10 and 1140 Points were placed on the unit square  $[0, 1]^2$  and connected with degree parameter 4. The maxcut SDPs were run for 15 reweighting rounds with chordal regularization  $W_k$  (`sedumi`, `sparsecolo_sedumi`, `sdpt3`, `cdcs`) and with nonchordal regularization  $W$  (`sdpt3_no_chord` and `sedumi_no_chord`) (difference between  $W_k$  and  $W$  in 1). Data is evaluated over 5 iterations (random seeds 2968, 1649, 845, 232, 1334), and all solvers have roughly equivalent cost functions at optima. Maxcut experiments were run on an Intel i7 6820HQ CPU on Matlab R2018a. The results are shown in Fig. 3 and Fig. 4. GPU acceleration was not fully utilized in this set of experiments, which especially hurt CDCS (parallel nuclear norm updates). Parallelism will be leveraged for future experiments.

### B. Subspace clustering

The problem of subspace clustering refers to the task of, given a set of  $N_p$  points  $x_j \in \mathbb{R}^D \forall_{j=1}^{N_p}$  sampled from  $N_s$  subspaces, find the normals  $r_i \in \mathbb{R}^D \forall_{i=1}^{N_s}$  of the sampled subspaces [39]. Alternatively, it can be posed as finding

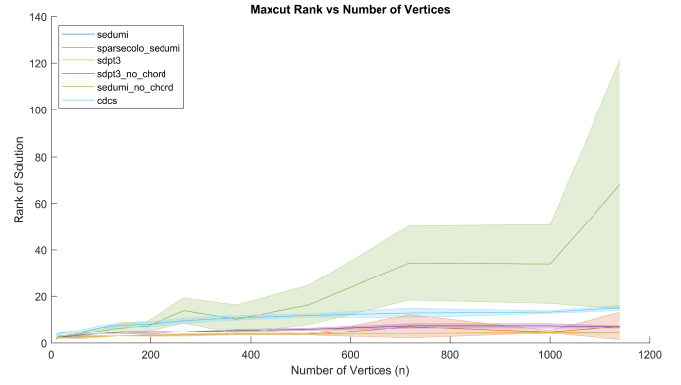


Fig. 4. Maxcut Rank  $\pm$  one standard deviation. With chordal regularization, SeDuMi and SDPT3 produces the lowest rank solutions. Without chordal regularization, SeDuMi produces extremely high rank solutions.

the set of binary labels  $s_{i,j} \forall_{i=1}^{N_s} \forall_{j=1}^{N_p}$  that assign each point  $x_j$  to the subspace spanned by the normal vector  $r_i$ . This problem arises in many practical applications, and in particular, in systems theory, in the context of identification of switched models from noisy input/output data (see *e.g.*, [40], [41]), where each subspace normal is defined by the coefficients of each switching system.

In the general formulation, a point  $x_j$  is said to belong to a given subspace if it is orthogonal to the subspace normal, *i.e.*,  $r_i^T x_j = 0$ . Under the presence of bounded noise, this orthogonality constraint is relaxed to  $|r_i^T x_j| \leq \epsilon$ , where  $\epsilon$  is the noise bound. The task of subspace clustering can be cast as the following feasibility problem:

$$\min_{r,s} \quad 0$$

$$s_{i,j} |r_i^T x_j| \leq s_{i,j} \epsilon, \quad \forall_{i=1}^{N_s}, \forall_{j=1}^{N_p}, \quad (10a)$$

$$s_{i,j} = s_{i,j}^2, \quad \forall_{i=1}^{N_s}, \forall_{j=1}^{N_p}, \quad (10b)$$

$$\sum_{i=1}^{N_s} s_{i,j} = 1, \quad \forall_{j=1}^{N_p}, \quad (10c)$$

$$r_i^T r_i = 1, \quad \forall_{i=1}^{N_s}, \quad (10d)$$

where (10a) controls the orthogonality constraint and is only active when  $s_{i,j} \neq 0$ , (10b) enforces the binary nature of the labels, (10c) assigns every point  $x_j$  to a subspace and (10d) forces the normal vectors to have unity norm (otherwise,  $r_i = 0$  would be a trivial feasible solution). The feasibility problem in (10) is nonconvex due to quadratic equality constraints and bilinear interactions between  $s$  and  $r$ .

In [4], it was proposed to reformulate (10) as an SDP by defining a matrix  $X = [1, v][1, v]^T$ , where  $v = [r_1, \dots, r_{N_s}, s_{1,1}, \dots, s_{N_s,1}, \dots, s_{N_s, N_p}]$  contains all the variables of (10) in vectorized form.  $X$  is a symmetric PSD matrix of size  $(1 + N_s(D + N_p))$ , and all the constraints in (10) become linear with respect to the entries of  $X$  at the cost of adding a non-convex rank 1 constraint. The reweighted nuclear norm heuristic was then employed to



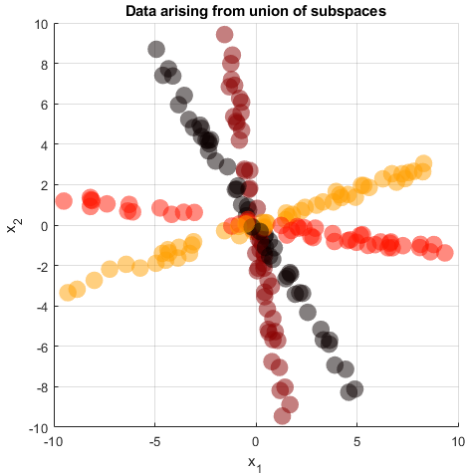


Fig. 5. A typical problem in Subspace Clustering

relax the rank constraint into a convex problem.<sup>1</sup>

We note that in problem (10) there are no bilinear interactions between different  $s_{i,j}$  terms, nor any interactions between  $s_{i,j}$  and  $r_i$  when  $\hat{i} \neq \hat{j}$ . As a result, only a very reduced number of entries of  $X$  are actually used in the constraints of (10), leading to a very sparse pattern. This pattern is shown, in dark grey, in Figure 6.a. To exploit this underlying sparsity, [4] proposed to solve (10) only in a chordal extension of the sparsity graph, shown by the union of light grey and red cells in Figure 6.a, instead of using all the entries of  $X$ . This approach led to a split from a single  $(1 + N_s(D + N_p))$ -sized PSD constraint to  $N_p$  PSD constraints of size  $(1 + N_s(D + 1))$ , and a reduction of the rank 1 constraint from a  $(1 + N_s(D + N_p))$ -sized matrix onto a  $(1 + N_sD)$  one, effectively decoupling the size of the constraints from the number of points  $N_p$ .

In this paper, we propose a reduced chordal extension of  $X$ , shown in red in Figure 6.a. This extension, besides being significantly smaller than the one in [4] with respect to the number of edges (a 13% increase in  $|\mathcal{E}|$ , compared to more than a 350% increase for [4]), allows us to simplify the aforementioned rank 1 constraint of size  $(1 + N_sD)$  into a collection of  $N_s$  rank 1 constraints of size  $(1 + D)$ , and breaking the  $N_p$  PSD constraints of size  $(1 + N_sD)$  into  $N_p N_s$  constraints of size  $(2 + D)$ , decoupling this time the size of the constraints from both the number of points  $N_p$  and the number of subspace  $N_s$ .

Next we analyze the performance of the proposed approach with respect to the number of subspaces and points of the subspace clustering problem, and compare it to that of [4]. Figure 6.b shows the average runtime of each reweighted heuristic iteration for a problem of  $D = 2$ ,  $N_p = 50$  and  $N_s = 2, \dots, 8$ . It is observed that the proposed approach presents a close-to-linear complexity with respect to the number of subspaces  $N_s$ , while [4] scales in a clear

<sup>1</sup>Due to the particular structure of the subspace clustering problem, enforcing a rank 1 constraint on a particular principal submatrix of  $X$  is equivalent to enforcing rank 1 on the overall matrix. The interested reader is referred to [4] for the proof of this exact relaxation.

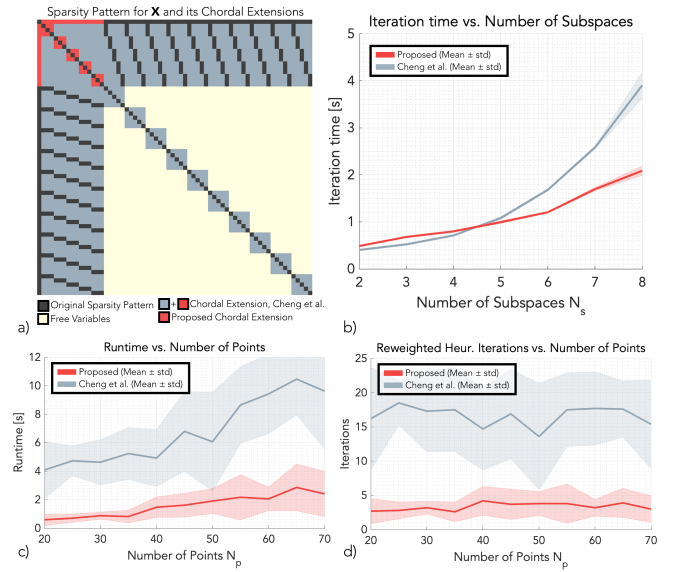


Fig. 6. a) Variable structure of  $X$  in Problem (10) and chordal extensions for [4] and this work. b) Runtime for a single iteration of reweighted heuristic (i.e. a whole SDP execution) for fixed  $D = 2$  and  $N_p = 50$  and variable  $N_s = 2, \dots, 8$  for the two chordal extensions in a). c) and d) Runtime and number of iterations of Algorithm 1 for  $D = 2$ ,  $N_s = 3$  as a function of points  $N_p$ .

superlinear fashion. These trends are consistent with our expectation of problems scaling worse with the size of the PSD constraints than with the amount of said constraints (the proposed approach has  $N_s$  times more constraints, but each constraint is  $N_s$  times smaller than those of [4]).

Finally, we test Algorithm 1 on the subspace clustering problem, for fixed  $D = 2$  and  $N_s = 3$  and variable  $N_p = 20, \dots, 70$ . A value of  $\delta = 10^{-3}$  is used for the update of  $W_k$  and the algorithm terminates whenever the dominant eigenvalue of the reweighted matrices is above 0.98 of the sum of eigenvalues or the iteration count reaches 20, whichever happens first. Figure 6.c shows the runtime of the algorithm for the proposed chordal extension and [4], and Figure 6.d shows the number of iterations until convergence. The proposed approach clearly outperforms [4] both in terms of runtime and iterations needed to converge, with [4] average being close to 20 due to many runs stopping by reaching the iteration threshold. Interestingly, the use of the reweighted heuristic on separate cliques seems to promote a faster convergence to an overall rank 1 solution than applying the heuristic to bigger submatrices of  $X$ . These results seem to show that there might be additional advantages to the use of chordal decompositions in rank-constrained SDPs beyond the computational speed-ups offered by the use of smaller cliques, and hint to possible size-dependencies in low-rank promotion techniques.

## VI. CONCLUSIONS

In this paper, we have applied the minimum rank completion to show an equivalence between minimizing rank over a matrix  $X$  and of its cliques  $X_k$  in a rank constrained

SDP. We then relaxed the rank penalty using the reweighted heuristic, and showed that this weighting  $W_C$  over cliques preserves the sparsity pattern. We discussed implementations of chordal rank minimized SDPs by interior-point and first-order methods. Scalability and efficiency of the chordal decomposition for rank minimization were demonstrated in the specific examples of maxcut and subspace clustering. We expect that these gains will hold in many other chordally sparse rank-minimized SDPs. Future work includes utilizing other rank surrogate functions and applying chordal rank minimization to more general polynomial and rational optimization problems.

## VII. ACKNOWLEDGMENTS

This work was supported in part by NSF grants CNS-1646121, CMMI-1638234, IIS-1814631 and ECCS-1808381; AFOSR grant FA9550-15-1-0392; and the Alert DHS Center of Excellence under Award Number 2013-ST-061-ED0001.

## REFERENCES

- [1] L. Vandenberghe and S. Boyd, "Applications of semidefinite programming," *Applied Numerical Mathematics*, vol. 29, no. 3, 1999.
- [2] 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. 6-7, pp. 383–392, 2008.
- [3] J. B. Lasserre, *Moments, Positive Polynomials And Their Applications*, ser. Imperial College Press Optimization Series. World Scientific Publishing Company, 2009.
- [4] Y. Cheng, Y. Wang, M. Sznaier, and O. Camps, "Subspace clustering with priors via sparse quadratically constrained quadratic programming," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5204–5212.
- [5] S. Burer, R. D. Monteiro, and Y. Zhang, "Rank-two relaxation heuristics for max-cut and other binary quadratic programs," *SIAM Journal on Optimization*, vol. 12, no. 2, pp. 503–521, 2002.
- [6] D. Henrion and J.-B. Lasserre, "Detecting global optimality and extracting solutions in gloptipoly," in *Positive polynomials in control*. Springer, 2005, pp. 293–310.
- [7] M. Journée, F. Bach, P.-A. Absil, and R. Sepulchre, "Low-rank optimization on the cone of positive semidefinite matrices," *SIAM Journal on Optimization*, vol. 20, no. 5, pp. 2327–2351, 2010.
- [8] N. Boumal, V. Voroninski, and A. Bandeira, "The non-convex burer-monteiro approach works on smooth semidefinite programs," in *Advances in Neural Information Processing Systems*, 2016.
- [9] Y. Liu, L. Jiao, F. Shang, F. Yin, and F. Liu, "An efficient matrix bi-factorization alternative optimization method for low-rank matrix recovery and completion," *Neural Networks*, vol. 48, pp. 8–18, 2013.
- [10] M. Souto, J. D. Garcia, and Á. Veiga, "Exploiting low-rank structure in semidefinite programming by approximate operator splitting," *arXiv preprint arXiv:1810.05231*, 2018.
- [11] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.
- [12] E. J. Candes and Y. Plan, "Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements," *IEEE Transactions on Information Theory*, vol. 57, no. 4, pp. 2342–2359, 2011.
- [13] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 2862–2869.
- [14] K. Mohan and M. Fazel, "Reweighted nuclear norm minimization with application to system identification," in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 2953–2959.
- [15] C. Grussler and P. Giselsson, "Low-rank inducing norms with optimality interpretations," *SIAM Journal on Optimization*, vol. 28, no. 4, pp. 3057–3078, 2018.
- [16] L. Vandenberghe, M. S. Andersen *et al.*, "Chordal graphs and semidefinite optimization," *Foundations and Trends® in Optimization*, vol. 1, no. 4, pp. 241–433, 2015.
- [17] V. Chandrasekaran, N. Srebro, and P. Harsha, "Complexity of inference in graphical models," *arXiv preprint arXiv:1206.3240*, 2012.
- [18] 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.
- [19] 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*, 2017.
- [20] —, "Fast admm for homogeneous self-dual embedding of sparse sdps," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8411–8416, 2017.
- [21] J. Dancis, "Positive semidefinite completions of partial hermitian matrices," *Linear algebra and its applications*, vol. 175, 1992.
- [22] X. Jiang, "Minimum rank positive semidefinite matrix completion with chordal sparsity pattern," Ph.D. dissertation, UCLA, 2017.
- [23] R. Madani, S. Sojoudi, G. Fazelnia, and J. Lavaei, "Finding low-rank solutions of sparse linear matrix inequalities using convex optimization," *SIAM Journal on Optimization*, vol. 27, no. 2, 2017.
- [24] M. Yannakakis, "Computing the minimum fill-in is np-complete," *SIAM Journal on Algebraic Discrete Methods*, vol. 2, no. 1, 1981.
- [25] R. Grone, C. R. Johnson, E. M. Sá, and H. Wolkowicz, "Positive definite completions of partial hermitian matrices," *Linear Algebra Appl.*, vol. 58, pp. 109–124, 1984.
- [26] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn, "Fast admm for semidefinite programs with chordal sparsity," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 3335–3340.
- [27] W. W. Barrett, C. R. Johnson, and M. Lundquist, "Determinantal formulae for matrix completions associated with chordal graphs," *Linear Algebra and its Applications*, vol. 121, pp. 265–289, 1989.
- [28] K. Fujisawa, S. Kim, M. Kojima, Y. Okamoto, and M. Yamashita, "Users manual for sparsecolo: Conversion methods for sparse conic-form linear optimization problems," *Research Report B-453, Dept. of Math. and Comp. Sci. Japan, Tech. Rep.*, pp. 152–8552, 2009.
- [29] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [30] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.
- [31] Y. Ye, M. J. Todd, and S. Mizuno, "An  $o(\sqrt{nl})$ -iteration homogeneous and self-dual linear programming algorithm," *Mathematics of Operations Research*, vol. 19, no. 1, pp. 53–67, 1994.
- [32] B. ODonoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," *Journal of Optimization Theory and Applications*, vol. 169, no. 3, pp. 1042–1068, 2016.
- [33] 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, 1999.
- [34] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, "Sdpt3a matlab software package for semidefinite programming, version 1.3," *Optimization methods and software*, vol. 11, no. 1-4, pp. 545–581, 1999.
- [35] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn, "CDCS: Cone decomposition conic solver, version 1.1," <https://github.com/giofantuzzi/CDCS>, Sep. 2016.
- [36] E. D. Andersen and K. D. Andersen, "The mosek interior point optimizer for linear programming: an implementation of the homogeneous algorithm," in *High performance optimization*. Springer, 2000, pp. 197–232.
- [37] J. Löfberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *Proceedings of the CACSD Conference*, vol. 3. Taipei, Taiwan, 2004.
- [38] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM (JACM)*, vol. 42, no. 6, pp. 1115–1145, 1995.
- [39] R. Vidal, "A tutorial on subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.
- [40] "Identification of hybrid systems a tutorial," *European Journal of Control*, vol. 13, no. 2, pp. 242 – 260, 2007.
- [41] Y. Cheng, Y. Wang, and M. Sznaier, "A convex optimization approach to semi-supervised identification of switched arx systems," in *53rd IEEE Conference on Decision and Control*, Dec 2014, pp. 2573–2578.