

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

Jared Miller¹, Yang Zheng², Biel Roig-Solvas¹, Mario Sznaier¹, Antonis Papachristodoulou³

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 re-weighted 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 problem of subspace clustering is 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, maxcut, and optimal power flow [1]. In particular, subspace clustering is an NP-hard problem that groups points originating from a union of subspaces and admits an SDP relaxation [2].

In each case, SDP relaxations return equivalent optima if the solution satisfies rank constraints. Maxcut, optimal power flow, and subspace clustering all require rank-1 solutions. The desire for a low-rank solution can be formulated as a rank-constrained SDP

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

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 NP-hard, a great deal of interest has been put in developing

¹J. 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).

²Y. Zheng was with the Department of Engineering Science, University of Oxford. He is now with SEAS and CGBC at Harvard University, Cambridge, MA, 02138. (Email: zhengy@g.harvard.edu).

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

J. Miller, B. Roig-Solvas and M. Sznaier were partially supported by NSF grants CNS-1646121, CMMI-1638234, IIS-1814631 and ECCS1808381 and AFORS grant FA9550-19-1-0005.

tractable rank proxies. Matrix factorization (*i.e.*, $X = YY^T$) upper bounds the possible rank by the width of Y , and Burer-Monteiro results may ensure global optimality after curvilinear optimization on the nonconvex low-rank manifold [3]. Several convex relaxations of the rank constraint have been developed, one of the most popular being the nuclear norm $\|X\|_* = \sum_i \sigma_i(X)$, and $\|X\|_* = \text{Tr}(X)$ if $X \in \mathbb{S}_+^n$ [4]. The nuclear norm is the biconjugate of the rank function, and under *restricted isometry property* (RIP)/coherence conditions, the nuclear norm-minimized solution of an SDP is equivalent to the rank-minimal optimum [5]. RIP/coherence holds only in a very narrow set of problems since it weighs all singular values equally. Low rank solutions can be encouraged by penalizing smaller singular values. 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 [6].

The complexity of solving SDPs scales in a polynomial time w.r.t. the number of variables and constraints, and imposing rank penalties may further increase this complexity. In sparse cases, only a small subset of entries of X are used in the cost C and constraints A_i , and the other entries guarantee $X \succeq 0$. When this sparsity pattern is (or can be extended to) a chordal graph, chordal decomposition theory can break up a large PSD cone ($X \succeq 0$) into a set of smaller coupled PSD cone ($X_k \succeq 0$) [7]. Exploiting this structure reduces computational time, as the complexity of optimization problems is related to the graph's tree-width [8]. Chordal decomposition of SDPs can effectively reduce dimension of sparse problems; see *e.g.*, [9], [10] for details.

After performing the chordal decomposition and optimization over the X_k 's, there exist multiple matrix completions to generate a valid $X \succeq 0$ from X_k . One such choice is the minimum rank completion [11], 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 of $\text{rank}(X)$, but penalizing $\text{rank}(X_k)$ in optimization was not considered [12]. 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 how to penalize the rank of tree components [13].

In this paper, we combine minimal rank completion and the reweighted heuristic to effectively solve large-scale rank-constrained SDPs. This heuristic penalty preserves the original problem's chordal sparsity while penalizing the rank of smaller matrices, and offers a rank guarantee based on the

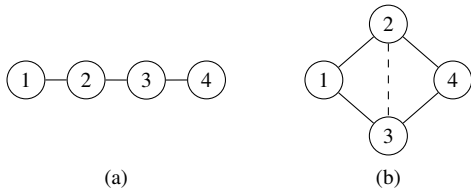


Fig. 1: Examples of chordal graphs: (a) a path graph; (b) a triangulated graph (with dashed edge). Without the dashed edge there is a cycle of length 4 without a chord, so the graph is not chordal. The graph with the dashed edge is a chordal extension.

size of the maximum clique. 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 of chordal rank minimization in interior-point and first-order methods are discussed in Section IV. These algorithms are used in Section V to solve subspace clustering problems. Section VI concludes the paper.

II. PRELIMINARIES

In this section, we cover chordal graphs, sparse (PSD) matrices, and the minimum rank completion. For a comprehensive treatment, the interested reader is referred to [7].

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 $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. A cycle of length N is a set of unique nodes v_k such that $(v_1, v_2), \dots, (v_i, v_{i+1}), \dots, (v_N, v_1) \in \mathcal{E}$. A chord is an edge that connects two nonconsecutive nodes in a cycle. An undirected graph is *chordal* if all cycles of length no less than four have at least one chord [7]. A chordal extension (a.k.a. completion) $\mathcal{G}_c(\mathcal{V}_c, \mathcal{E}_c)$ of graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is a chordal graph \mathcal{G}_c where $\mathcal{V} = \mathcal{V}_c$ and $\mathcal{E} \subseteq \mathcal{E}_c$. Finding a chordal extension with a minimal number of added edges is NP-hard, but efficient heuristics exist to give good chordal extensions [14].

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

B. Sparse matrices and chordal decomposition

Considering the SDP (1), the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ arises from taking a union of the sparsity graphs corresponding to the data matrices C, A_1, A_2, \dots, A_m . Precisely, let $\hat{\mathcal{E}} = \mathcal{E} \cup \{(i, i), \forall i \in \mathcal{V}\}$ be the edge set with self loops, and we define a cone of sparse symmetric matrices as:

$$\mathbb{S}^n(\mathcal{E}, 0) = \{X \in \mathbb{S}^n \mid X_{ij} = 0, \forall (i, j) \notin \hat{\mathcal{E}}\}$$

$\mathbb{S}_+^n(\mathcal{E}, 0) = \mathbb{S}(\mathcal{E}, 0) \cap \mathbb{S}_+^n$ forms a cone of sparse PSD matrices. 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 with entries defined in \mathcal{E} ($\exists M \succeq 0 \mid X_{ij} = M_{ij} \forall (i, j) \in \mathcal{E}^*$). 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 [15]): 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 [7], [9], [10].

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 [16]. Minimum rank completions are not necessarily unique:

Theorem 2 (Minimum rank completion [11]): 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 $X \in \mathbb{S}_+(\mathcal{E}, ?)$, 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 [12], which updates a factorization of the completion while proceeding through the elimination tree.

We conclude this section with the following example:

$$\begin{bmatrix} 3 & 0.5 & ? & 0.25 \\ 0.5 & 2 & 0.75 & ? \\ ? & 0.75 & 1 & ? \\ 0.25 & ? & ? & 5 \end{bmatrix} \quad \begin{bmatrix} 3 & 0.5 & -1.25 & 0.25 \\ 0.5 & 2 & 0.75 & -3.05 \\ -1.25 & 0.75 & 1 & -1.65 \\ 0.25 & -3.05 & -1.65 & 5 \end{bmatrix}$$

Fig. 2: PSD Completable matrix (left) and rank-2 completion (right)

The maximum cliques of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ are $\{(1, 2), (2, 3), (1, 4)\}$, and their induced submatrices are all PSD with rank 2, so there exists a completion with rank at least 2 (right matrix).

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 reweighted heuristic to the decomposed problem.

A. An equivalent reformulation

Let the rank constrained SDP (1) be sparse with an aggregate sparsity pattern $\mathcal{G}(\mathcal{V}, \mathcal{E})$: $C \in \mathbb{S}^n(\mathcal{E}, 0)$, $A_i \in \mathbb{S}^n(\mathcal{E}, 0)$, $i = 1, \dots, m$. Throughout this paper, we assume that $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is chordal (or has a suitable chordal extension) with a set of maximal cliques $\{C_k\}_{k=1}^p$. 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

$$\begin{aligned} \min_X \quad & \langle C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i, i = 1, \dots, m, \\ & E_{C_k} X E_{C_k}^T \in \mathbb{S}_+^{|C_k|}, k = 1, \dots, p, \\ & \text{rank}(E_{C_k} X E_{C_k}^T) \leq t, k = 1, \dots, p, \end{aligned} \quad (2)$$

in the sense that (1) and (2) 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 (2) as f_1^* and f_2^* respectively.

- First, assume X_1^* is an optimal solution to (1) with cost $f_1^* = \langle C, X_1^* \rangle$ and $\text{rank}(X_1^*) \leq t$. Then, X_1^* is also a feasible solution to (2). Thus $f_2^* \leq f_1^*$.
- Second, assume X_2^* is an optimal solution to (2) with an cost value $f_2^* = \langle C, X_2^* \rangle$. According to Theorems 1 and 2, we can find a PSD completion \hat{X}_2^* , where $\text{rank}(\hat{X}_2^*) = \max_k \text{rank}(E_{C_k} X_2^* E_{C_k}^T) \leq t$. 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 (2) can be recovered from each other. ■

One key feature of problem (2) 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 (2) is hard to solve due to the rank constraints. In this paper, we replace the hard rank constraint with a soft reweighted heuristic [6], leading to a standard SDP with chordal sparsity. The reweighted heuristic relaxes the hard $\text{rank}(X) \leq t$ constraint to a soft $\langle W, X \rangle$ term on the objective. For problem (2), each rank constraint $\text{rank}(E_{C_k} X E_{C_k}^T) \leq t$ is replaced by a soft penalty $\langle W_k, E_{C_k} X E_{C_k}^T \rangle$ for a reweighting matrix W_k calculated in Algorithm 1. In general, the inner product between W_k and X_k promotes the concentration of energy of X_k onto the

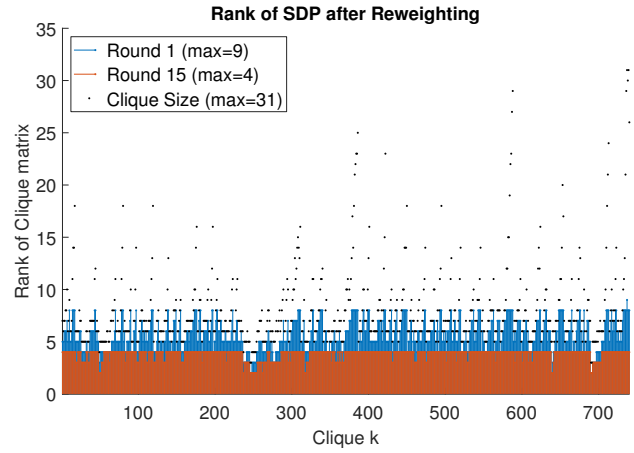


Fig. 3: Example of reweighting heuristic (Algorithm 1) on a randomly generated Maxcut problem with 1000 vertices. There are 740 cliques with $|C^{\max}| = 31$ (black dots). The maximum clique rank starts at 9 (blue), and drops to 4 (orange) after 15 rounds of optimization.

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:      $(X) \leftarrow$  optimum of (3) given  $W_k$ 
5:      $\tilde{W}_k \leftarrow (E_{C_k} X_k E_{C_k}^T + \delta I)^{-1}$ 
6:      $W_k \leftarrow \tau_k \tilde{W}_k / \|\tilde{W}_k\|_2$ 

```

dominant eigenspace of $X_{k,\text{old}}$ and thus incentivizes the reduction of its rank. Weights W_k are normalized in Algorithm 1, and τ_k is a per clique regularization parameter. Reweighted heuristic is a local linearization of log-det and has been successfully used to solve rank minimization problems [6]. Fig. 3 demonstrates the rank reduction behavior of Algorithm 1 on a Maxcut SDP.

The individual W_k 's can be combined together into an agglomerated penalty $W_C = \sum_{k=1}^p E_{C_k}^T W_k E_{C_k}$ by the cyclical property of Trace:

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

This allows us to write the rank relaxed problem into the following form:

$$\begin{aligned} \min_X \quad & \langle C, X \rangle + \langle W_C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i, i = 1, \dots, m, \\ & E_{C_k} X E_{C_k}^T \geq 0, \forall k = 1, \dots, p. \end{aligned} \quad (3)$$

The augmented cost matrix $C + W_C \in \mathbb{S}^n(\mathcal{E}, 0)$ and $A_i \in \mathbb{S}^n(\mathcal{E}, 0)$, indicating that Problem (3) has the same aggregate sparsity pattern as (1). If the original sparsity structure \mathcal{G} is non-chordal with a chordal extension $\mathcal{G}_c(\mathcal{V}, \mathcal{E}_c)$, the first reweighting iteration will have $W_C \in \mathbb{S}^n(\mathcal{E}_c, 0)$. After the initial reweighting, there will be no more fill-in.

Remark 1: As shown in Proposition 1, Problems (1) and (2) are equivalent. This tightness is lost when conducting rank relaxations. Applying the reweighted heuristic directly to Problem (1) requires the inverse of the big X to update the weight $W = (X + \delta I)^{-1}$. As inverses are generically dense, the next iteration's cost $C + W$ would be dense and the sparsity pattern would be destroyed. In the proposed method, only the inverses of the small X_k matrices are required and the weight W_C in (3) preserves the sparsity structure. Reweighted heuristic on problems (1) and (2) 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. The convergence time of reweighted heuristic increases as the size of X goes up, so in general it will take fewer iterations for a W_C -based scheme to converge than an algorithm using W [17].

IV. ALGORITHM IMPLEMENTATIONS

Problem (3) is convex and readily solved by conic solvers, e.g., SeDuMi or Mosek. However, naively passing to solvers does not scale well to large-scale instances. In this section, we modify Problem (3) to exploit its structure for adaptation in both interior point methods and first order methods.

A. Interior-point methods

Interior point methods (IPM) such as SeDuMi will suffer from additional equality constraints introduced from the chordal decomposition forcing equality between cliques (clique-tree method). A different chordal extension $\tilde{\mathcal{G}} \supset \mathcal{G}$ may be used for optimization than the reweighting \mathcal{G} that forms W_C . SparseCoLo [18] or other methods may perform this decomposition to find an equivalent problem that is kinder to IPM, but reweighting is strictly based on the original pattern. IPM will be given Problem (3) as input, after a possible decomposition. In Algorithm 1, only the cost matrix $C + W_C$ changes between reweighting iterations, so no repeat conversions are necessary. Interior point methods generically find full rank solutions, so 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^* .

B. Alternating Direction Method of Multipliers (ADMM)

Following [10], we build an ADMM algorithm to solve (2). Grone's theorem has a natural variable split $X_k = E_{C_k} X E_{C_k}^T$ that separates affine and PSD constraints. The resulting optimization problem is:

$$\begin{aligned} \min_{X, X_k} \quad & \langle C + W_C, X \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i, i = 1, \dots, m \\ & X_k = E_{C_k} X E_{C_k}^T, \\ & X_k \succeq 0, \forall k = 1, \dots, p. \end{aligned} \quad (4)$$

Each iteration of ADMM has three steps [19]: The affine constraints in X are handled in step 1, the PSD projection in X_k is from step 2, and the dual ascent on dual variables

Λ_k to enforce the variable-split is on step 3. First order algorithms such as ADMM sometimes converge slowly, but have a relatively low per-iteration cost. Applying ADMM to (4) requires solving subproblems at each iteration t (with optional adjustment of ρ between iterations following [19]):

Step 1: Solve the following quadratic program (QP)

$$\begin{aligned} \min_X \quad & \langle C + W_C, X \rangle + \frac{\rho}{2} \sum_{k=1}^p \left\| X_k^{(t)} - E_{C_k} X E_{C_k}^T + \frac{1}{\rho} \Lambda_k^{(t)} \right\|_F^2 \\ & \langle A_i, X \rangle = b_i, i = 1, \dots, m. \end{aligned} \quad (5)$$

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

$$\min_{X_k \succeq 0} \frac{\rho}{2} \left\| X_k - E_{C_k} X^{(t+1)} E_{C_k}^T + \frac{1}{\rho} \Lambda_k^{(t)} \right\|_F^2 \quad (6)$$

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 $H_k \text{vec}(X) = \text{vec}(E_{C_k} X E_{C_k}^T)$. Likewise $a_i = \text{vec}(A_i)$ is collated into A and b_i into b , $c = \text{vec}(C + W_C)$, and $v_k = \text{vec}(X_k^{(t)} + \frac{1}{\rho} \Lambda_k^{(t)})$. Step 1 is vectorized into:

$$\begin{aligned} \min_x \quad & \langle c, x \rangle + \frac{\rho}{2} \sum_{k=1}^p \|H_k x - v_k\|_2^2 \\ \text{subject to} \quad & Ax = b. \end{aligned} \quad (7)$$

The KKT system for the QP involves x and dual variable ω :

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

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; see [10] for detailed discussions on a numerical implementation.

Step 2 involves parallel PSD projections: find the eigen-decomposition of $V_k = E_{C_k} X^{(t+1)} E_{C_k}^T - \frac{1}{\rho} \Lambda_k^{(t)}$ and keep the positive eigenvectors. If desired, the reweighting W_C on X can be moved to an individual weighting W_k on X_k .

C. Homogeneous Self-Dual Embedding

Problem (4) can be solved through the homogeneous self-dual embedding (HSDE) framework [20]. 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 [21], which is based on the general conic formulation introduced in [22]. Each iteration of HSDE is comprised of a large block-sparse linear system, projection onto a product of multiple cones, and a dual update step (see [21], [22] for details). If s is the concatenation of $\text{vec}(X_k)$ and $c = \text{vec}(C + W_C)$, the linear system in HSDE $v = Qu$ is:

$$\begin{bmatrix} h \\ z \\ r \\ w \\ k \end{bmatrix} = \begin{bmatrix} & & -A^T & -H^T & c \\ & & & I & 0 \\ A & & & & -b \\ H & -I & & & \\ -c^T & 0^T & b & & \end{bmatrix} \begin{bmatrix} x \\ s \\ y \\ v \\ t \end{bmatrix}.$$

V. SUBSPACE CLUSTERING

In this section, we demonstrate the performance of Algorithm 1 to solve subspace clustering (see Fig. 4 a.). 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 [23]. Alternatively, it can be posed as finding 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 including switched system identification from noisy input/output data (see *e.g.*, [24], [25]), where each subspace normal is defined by the coefficients of each switching system.

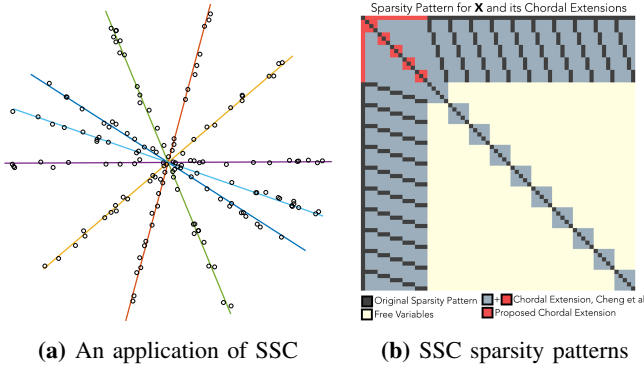


Fig. 4: SSC example and sparsity patterns

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

$$\text{find}_{r,s} \quad s_{i,j} |r_i^T x_j| \leq s_{i,j} \epsilon, \quad \forall_{i=1}^{N_s}, \forall_{j=1}^{N_p}, \quad (8a)$$

$$s_{i,j} = s_{i,j}^2, \quad \forall_{i=1}^{N_s}, \forall_{j=1}^{N_p}, \quad (8b)$$

$$\sum_{i=1}^{N_s} s_{i,j} = 1, \quad \forall_{j=1}^{N_p}, \quad (8c)$$

$$r_i^T r_i = 1, \quad \forall_{i=1}^{N_s}, \quad (8d)$$

where (8a) controls the orthogonality constraint and is only active when $s_{i,j} \neq 0$, (8b) enforces binary labels, (8c) assigns every point x_j to a subspace and (8d) forces normal vectors to have unit norm (otherwise $r_i = 0$ is a trivial feasible solution). Problem (8) is nonconvex due to quadratic equality constraints and bilinear interactions between s and r .

Cheng *et al.* proposed to reformulate (8) as an SDP by defining a matrix $X = [1, v] [1, v]^T$, with

$$v = [r_1, \dots, r_{N_s}, s_{1,1}, \dots, s_{N_s,1}, \dots, s_{N_s, N_p}]$$

containing all the variables of (8) in vectorized form [2]. X is a symmetric PSD matrix of size $(1 + N_s(D + N_p))$, and all the constraints in (8) 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 relax the rank constraint into a convex problem.¹

We note that problem (8) lacks bilinear interactions between different $s_{i,j}$ terms or any interactions between $s_{i,j}$ and r_i when $\bar{i} \neq \hat{i}$. As a result, only a very reduced number of entries of X are actually used in the constraints of (8), leading to a very sparse pattern. The sparsity pattern for an SSC problem with parameters ($D = 3, N_p = 10, N_s = 5$) is shown in dark grey in Figure 4.b. To exploit this underlying sparsity, [2] proposed to solve (8) with variables in a chordal extension, shown by the union of light grey and red cells in Figure 4.b. In this paper, we propose a reduced chordal extension of X , shown in red in Figure 4.b. The new extension is linear in N_s and N_p , and imposes $N_s [D + 1]$ -size rank-1 constraints instead of a single larger $[1 + N_s D]$ constraint. A summary of the clique matrix sizes $|C_k|$ is:

Problem	Rank 1 PSD		Other PSD		% edges
	# Cliques	Size Cliques	# Cliques	Size Cliques	
Full X	1	$1 + N_s(D + N_p)$	\emptyset	\emptyset	1637%
Cheng	1	$1 + N_s D$	N_p	$1 + N_s(D + 1)$	350%
Ours	N_s	$D + 1$	$N_p N_s$	$D + 2$	13%

TABLE I: Clique sizes in SSC. %edges measures size of chordal extension over baseline \mathcal{E} (variables in (8)). Parameters are ($D = 3, N_p = 10, N_s = 5$) as in Fig. 4 b.

For our extension, each clique $v_{i,j} = [1, r_i, s_{i,j}]$ where r_i includes all coordinates of the normal. Experiments were run on SSC using YALMIP and Mosek. Data was generated by randomly choosing N_s subspace normals $r_i \in \mathbb{R}^D$, then sampling N_p total points x_j from these subspaces. Points were corrupted by uniform noise with bounds $\pm \epsilon$. W_C -based regularization is applied until a rank-1 solution ($\sigma_1 / (\sum \sigma_i) > 99\%$) is found or until 20 iterations. Results are recorded in Fig. 6, all plots are mean ± 1 stdev.

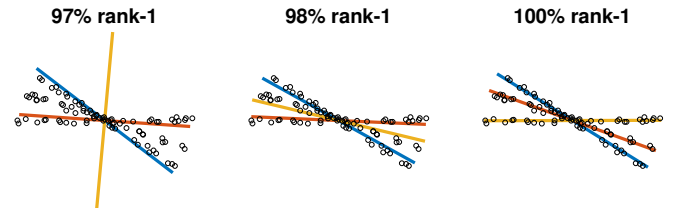


Fig. 5: SSC SDP convergence to satisfying Problem (8) as solution approaches rank-1. $N_s = 3, N_p = 90, D = 2, \epsilon = 0.15$

Experiments are run on five penalty schemes: imposing that X is rank-1, that the top-left corner of X is rank-1, with the Cheng scheme, by Problem (3), and by preprocessing Problem (3) with SparseCoLO. The SparseCoLO scheme uses W_C for rank penalization, and finds another decomposition before passing to Mosek. As an example with $N_s = 4, N_p = 80, D = 5$, Problem (3) has 320 PSD blocks of size $D + 2 = 7$, with 320 rank-1 constraints of size

¹Due 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 Appendix A of [2] for the proof of this exact relaxation.

$D+1 = 6$. For this problem, SparseCoLO generates 25 PSD blocks with size between 12 and 30, and W_C regularization and rank-guarantees are performed on 6×6 sub-blocks. In all tests, Problem (3) with SparseCoLO is an order of magnitude faster than other problem formulations.

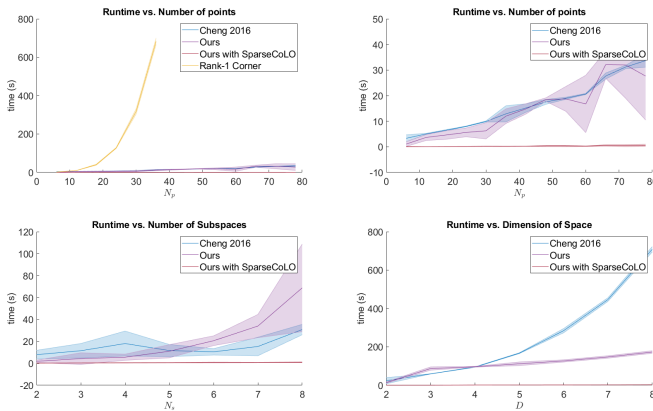


Fig. 6: SSC Experiments, changing individually N_p (top), N_s (bottom left), and D (bottom right).

The top left plot of Fig. 6 has $N_s = 3, N_p = 6 * (1 \dots 13), D = 3$. Imposing $\text{rank}(X) = 1$ directly never converges to a rank-1 solution. Setting $\text{rank}(X_{\text{top left corner}}) = 1$ is shown in yellow, and with $N_p = 42$ each run takes 2 hours (not shown). The top right plot zooms in, and shows the Cheng formulation, Problem (3), and Problem (3) as preprocessed by SparseCoLO [18]. Adding in clique-overlap constraints in Problem (3) degrades performance as compared to Cheng, but with SparseCoLO preprocessing it wins out. In the bottom left, $N_s = 2 \dots 8, N_p = 120, D = 2$ over 12 tests. $\text{rank}(X) = 1$ and $\text{rank}(X_{\text{corner}}) = 1$ do not converge to rank-1 solutions, and in this case Problem (3) without SparseCoLO is slower than Cheng. The bottom right plot has $N_s = 4, N_p = 80, D = 2 \dots 8$ over 8 tests, and as D increases our proposed formulation clearly beats Cheng. Variability in the plots is often due to iterations needed to find a low rank solutions. In some cases, rank-1 solutions are found in the first iterations, others require 20 iterations to become nearly rank-1 (95%). These experiments also involve all $\tau_k = 1$. Unequal weights of cliques (possibly by size of block) is of future interest.

VI. CONCLUSIONS

In this paper, we combined the minimum rank completion with reweighted heuristic to solve rank-minimization problems with chordal sparsity. We showed that rank constraints only need to be placed on the cliques X_k , and that using W_C maintains the sparsity pattern. We discussed implementations of chordal rank minimized SDPs by interior-point and first-order methods. We demonstrated the scalability and efficiency of the chordal decomposition for rank minimization in the specific example of subspace clustering. We expect that these gains will hold in many other chordally sparse rank-minimized SDPs, such as group (e.g. \mathbb{Z}_2) synchronization. Future work includes utilizing other rank surrogate functions

and applying chordal rank minimization to more general polynomial and rational optimization problems.

REFERENCES

- [1] L. Vandenberghe and S. Boyd, "Applications of semidefinite programming," *Applied Numerical Mathematics*, vol. 29, no. 3, 1999.
- [2] Y. Cheng, Y. Wang, M. Sznaier, and O. Camps, "Subspace clustering with priors via sparse quadratically constrained quadratic programming," in *IEEE CVPR*, 2016, pp. 5204–5212.
- [3] 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.
- [4] B. Recht, M. Fazel, and P. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM review*, vol. 52, no. 3, pp. 471–501, 2010.
- [5] E. Candes and Y. Plan, "Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2342–2359, 2011.
- [6] 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.
- [7] 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.
- [8] V. Chandrasekaran, N. Srebro, and P. Harsha, "Complexity of inference in graphical models," *arXiv preprint arXiv:1206.3240*, 2012.
- [9] M. Fukuda, M. Kojima, K. Murota, and K. Nakata, "Exploiting sparsity in semidefinite programming via matrix completion i: General framework," *SIAM J. on Opt.*, vol. 11, no. 3, pp. 647–674, 2001.
- [10] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn, "Chordal decomposition in operator-splitting methods for sparse semidefinite programs," *Mathematical Programming*, pp. 1–44, 2019.
- [11] J. Dancis, "Positive semidefinite completions of partial hermitian matrices," *Linear algebra and its applications*, vol. 175, 1992.
- [12] X. Jiang, "Minimum rank positive semidefinite matrix completion with chordal sparsity pattern," 2017.
- [13] 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.
- [14] M. Yannakakis, "Computing the minimum fill-in is np-complete," *SIAM Journal on Algebraic Discrete Methods*, vol. 2, no. 1, 1981.
- [15] R. Grone, C. Johnson, E. Sá, and H. Wolkowicz, "Positive definite completions of partial hermitian matrices," *Linear Algebra Appl.*, vol. 58, pp. 109–124, 1984.
- [16] 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.
- [17] K. Mohan and M. Fazel, "Iterative reweighted algorithms for matrix rank minimization," *Journal of Machine Learning Research*, vol. 13, no. Nov, pp. 3441–3473, 2012.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn, "Fast admm for homogeneous self-dual embedding of sparse sdps," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 8411–8416, 2017.
- [22] B. ODonoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," *J. of Opt. Theory and Applications*, vol. 169, no. 3, pp. 1042–1068, 2016.
- [23] R. Vidal, "A tutorial on subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 52–68, 2011.
- [24] S. Paoletti, A. L. Juloski, G. Ferrari-Trecate, and R. Vidal, "Identification of hybrid systems a tutorial," *European journal of control*, vol. 13, no. 2-3, pp. 242–260, 2007.
- [25] 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.