Finding sparse, equivalent SDPs using minimal coordinate projections

Frank Permenter Pablo A. Parrilo

***Abstract*— We present a new method for simplifying SDPs that blends aspects of symmetry reduction with sparsity exploitation. By identifying a subspace of sparse matrices that provably intersects (but doesn't necessarily contain) the set of optimal solutions, we both block-diagonalize semidefinite constraints and enhance problem sparsity for many SDPs arising in sums-of-squares optimization. The identified subspace is in analogy with the fixed-point subspace that appears in symmetry reduction, and, as we illustrate, can be found using an efficient combinatorial algorithm that searches over coordinate projections. Effectiveness of the method is illustrated on several examples.**

I. Introduction

Many problems in engineering and control can be posed as semidefinite programs (SDPs)—convex optimization problems over the cone of positive semidefinite (psd) matrices. While semidefinite programs are efficiently solvable in theory, solving many SDPs of practical interest is computationally infeasible unless problem specific structure is exploited. In this paper, we present a new method for exploiting problem structure that blends aspects of existing techniques, namely *symmetry reduction* [11], [25], [6], [7] and *sparsity exploitation*.

In symmetry reduction, one uses *group structure* of the data matrices to find a subspace that provably intersects, but doesn't necessarily contain, the set of optimal solutions. One then reduces the dimensionality of the problem by restricting to this subspace, which given its structure, also block-diagonalizes the semidefinite constraint after a change of basis. Inspired by this, we present a method for finding a subspace of *sparse matrices* (in the standard basis) that provably intersects, but doesn't necessarily contain, the set of optimal solutions. In addition, the subspace we find is block-diagonal, up to permutation. Hence, by restricting to this subspace, we not only reduce the dimension of the feasible set but also reduce the cost of the semidefinite constraint.

Since the identified subspace is block-diagonal (up to permutation), it is trivially the symmetric part of a *matrix *-algebra*. In contrast with *-algebra-based methods surveyed in [6], our method does not require the data matrices generate a low-dimensional algebra nor be invariant under the action of a permutation group. Similarly, in contrast with other sparsity-based techniques, our method does not require aggregate sparsity of the data matrices [10] nor does it impose sparsity at the potential cost of optimality [26] [18]. In other words, it allows one to assume sparsity not immediate from the problem description without penalty.

P. A. Parrilo and F. Permenter are with the Laboratory For Information and Decision Systems (LIDS), Massachusetts Institute of Technology, Cambridge, MA 02139. {fperment,parrilo}@mit.edu

To identify the desired subspace, we find what we call the *minimal-coordinate-projection* of a given SDP, which is analogous to the *group-average* operation from symmetry reduction. The minimal-coordinate-projection has range of minimum dimension among coordinate projections that separately leave sub-level sets of the cost function, the psd cone, and the solution set of the linear constraints invariant. These invariance properties ensure the range of this projection intersects the set of solutions. In addition, a simple combinatorial algorithm finds the minimal-coordinate-projection without leveraging—nor requiring—any group structure.

As we will illustrate with examples, the minimal-coordinate-projection frequently has low-dimensional range for SDPs arising in *sums-of-squares* (SOS) optimization [3]. Indeed, finding the minimal-coordinate-projection generalizes and strengthens extremely effective parsing algorithms used to simplify these SDPs. The first algorithm, due to [16], exploits *sign-symmetries* (where, e.g., a polynomial $f(x, y)$ is called *sign-symmetric* if $f(x, y) = f(x, -y)$, $f(x, y) = f(-x, y)$, or $f(x, y) = f(-x, -y)$). The second, due to [5], exploits polynomial sparsity. Though we omit proof, the range of the minimal-coordinate-projection is always contained in the subspaces (implicitly) identified by these algorithms when SDPs are formulated using the monomial basis. Finding this smaller subspace is also at no additional computational cost. Indeed, the minimal-coordinate-projection can be found in polynomial-time, whereas [16] and [5] (implicitly) find subspaces via exhaustive search.

An outline of this paper follows. We first introduce notation and preliminaries. We then characterize projections onto sets of sparse matrices that satisfy the desired invariance conditions (Sections II and III). Section IV gives an efficient combinatorial algorithm for finding the minimal-coordinate-projection. We conclude by showing effectiveness of our method on several SDPs arising in sums-of-squares optimization—many taken from third-party libraries (Section V)—and mention an implementation (Section VI). Proofs are omitted and will appear in a full version of this paper.

A. Notation

Let $\mathbb{S}^n$ denote the vector space of $n \times n$ symmetric matrices equipped with trace inner-product $A \cdot B := \mathrm{Tr}(A^T B)$ and let $\mathbb{S}^n_+$ denote the subset of matrices that are positive semidefinite (psd). Let $A \succeq 0$ denote the condition that $A \in \mathbb{S}^n$ is psd, and for a positive integer m, let $[m] := \{1, \ldots, m\}$. For $A, B \in \mathbb{S}^n$, let $A \circ B \in \mathbb{S}^n$ denote the *Hadamard*, or entrywise, product of A and B, defined by the equation $[A \circ B]_{ij} = A_{ij} B_{ij}$. Let $\{0, 1\}^{n \times n}$ equal the set of $n \times n$

binary matrices. Finally, let $\operatorname{rng} P$ and $\ker P$ denote the range and kernel of a linear map P.

B. Equivalent SDPs

For an affine set $\mathcal{A} \subseteq \mathbb{S}^n$ and matrix $C \in \mathbb{S}^n$, consider the semidefinite program (SDP) in decision variable X

$$\begin{array}{ll} \text{minimize} & C \cdot X \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+. \end{array} \tag{1}$$

Our goal is to efficiently find a linear subspace $\mathcal{L}$ (preferably, of minimal dimension) for which the semidefinite program (1) has optimal value equal to that of the semidefinite program (2), given by

$$\begin{array}{ll} \text{minimize} & C \cdot X \\ \text{subject to} & X \in \mathcal{A} \cap \mathbb{S}^n_+ \cap \mathcal{L}. \end{array} \tag{2}$$

We will find $\mathcal{L}$ using the following sufficient condition:

Proposition 1: For the semidefinite programs (1)-(2), let $P : \mathbb{S}^n \to \mathbb{S}^n$ be a linear map with adjoint $P^* : \mathbb{S}^n \to \mathbb{S}^n$ and assume the following conditions hold:

(a) $P(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$, i.e., P is a *positive* map;
(b) $P(\mathcal{A}) \subseteq \mathcal{A}$;
(c) $P^*(C) = C$,

where $P(\mathbb{S}^n_+) := \left\{P(X) : X \in \mathbb{S}^n_+\right\}$ and $P(\mathcal{A})$ is similarly defined. If one takes $\mathcal{L} = \operatorname{rng} P$, then the optimal values of SDP (1) and (2) are equal.

Proposition 1 follows by noting P maps feasible (resp. optimal) points to feasible (resp. optimal) points by (a)-(c).

C. The minimal-coordinate-projection

Finding a linear map $P : \mathbb{S}^n \to \mathbb{S}^n$ that satisfies (a)-(c) with range of minimal dimension is a natural procedure for finding $\mathcal{L}$. In this paper, we consider a variant of this procedure that admits an efficient combinatorial solution. Specifically, we include an additional constraint,

(d) P is a coordinate projection, i.e., for a fixed $M \in \mathbb{S}^n \cap \{0,1\}^{n\times n}$, $P(X) = M \circ X$ for all X,

and give an algorithm for finding the *minimal-coordinate-projection*, defined using this extra constraint as follows:

Definition 1: The minimal-coordinate-projection is the unique minimizer of $\dim \operatorname{rng} P$ over linear maps $P : \mathbb{S}^n \to \mathbb{S}^n$ satisfying conditions (a)-(d).

That the minimal-coordinate-projection is unique follows from the fact maps satisfying (a)-(d) are closed under composition, and the fact (d) defines a finite set of pair-wise commuting maps. Specifically, these facts imply the minimal-coordinate-projection equals the composition of all coordinate projections satisfying (a)-(c).

D. Block-diagonalization

For reasons discussed in Section II-A, a coordinate projection that is also a positive map corresponds with a sparsity pattern that is block-diagonal (up to permutation). For $n = 4$, example sparsity patterns of this type include:

$$\begin{pmatrix} * & 0 & * & 0 \\ 0 & * & 0 & * \\ * & 0 & * & 0 \\ 0 & * & 0 & * \end{pmatrix}, \begin{pmatrix} * & * & * & 0 \\ * & * & * & 0 \\ * & * & * & 0 \\ 0 & 0 & 0 & * \end{pmatrix}, \begin{pmatrix} * & 0 & 0 & * \\ 0 & * & 0 & 0 \\ 0 & 0 & * & 0 \\ * & 0 & 0 & * \end{pmatrix}. \tag{3}$$

Hence, by taking $\mathcal{L}$ equal to the range of the minimal-coordinate-projection, the cost of the semidefinite constraint in SDP (2) is reduced. Specifically, $\mathbb{S}^n_+ \cap \mathcal{L}$ decomposes into a direct-sum of smaller psd cones.

E. Additional notation and remarks

1) Coordinate projections and binary matrices: In the rest of the paper, we make the correspondence between coordinate projections and binary matrices explicit, i.e., for $M \in \mathbb{S}^n \cap \{0,1\}^{n\times n}$, we define $P_M : \mathbb{S}^n \to \mathbb{S}^n$ via $P_M(X) = M \circ X$. We will often state results directly in terms of M.

2) Sparsity of the cost matrix: For a coordinate projection P_M, condition (c) of Proposition 1 is equivalent to the condition that $M \circ C = C$. Hence, the proposed method works well—i.e., the range of P_M is of low dimension—only if C is sparse. While this is a strong constraint to impose in general, C is typically extremely sparse for SDPs arising in sums-of-squares optimization (e.g., [2]), and, of course, equals zero for SDP feasibility problems.

II. Positive coordinate projections and binary-psd-matrices

This section characterizes binary matrices M for which $P_M(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$, i.e., we characterize coordinate projections that satisfy condition (a) of Proposition 1. We quickly find the matrices M equal the set of binary, psd matrices, and then recall characterizations of this latter set from the literature. We remark [8] studies the cone $P_M(\mathbb{S}^n_+)$ for arbitrary $M \in \mathbb{S}^n \cap \{0,1\}^{n\times n}$. The relationship between positive maps P_M and binary-psd-matrices is now established:

Lemma 1: Fix $M \in \mathbb{S}^n \cap \{0,1\}^{n\times n}$. Then, the map P_M is positive, i.e., $P_M(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$, if and only if M is positive semidefinite.

Proof: Sufficiency follows from the Schur product theorem (e.g., [14], Chapter 7), which states the Hadamard product of psd matrices is psd. For necessity, note $P_M(\mathbf{1}\mathbf{1}^T) = M$, where $\mathbf{1}\mathbf{1}^T$ is the psd matrix of all ones. ∎

Characterizations of binary-psd-matrices using partitions of $[n]$ and transitive relations now follow.

A. Binary-psd-matrices and partitions

A non-zero, binary matrix is psd if and only if it is the sum of binary, psd matrices that are rank one [15] [9]. It is not hard to see this characterization is equivalent to the following. (A direct proof, assuming the diagonal entries of M are non-zero, appears in Proposition 19.9 of [17].)

Lemma 2: For $M \in \mathbb{S}^n \cap \{0,1\}^{n\times n}$, the following statements are equivalent:

- The matrix M is psd.
- There exists disjoint subsets $S_1, \ldots, S_p$ of $[n]$ for which

$$M_{ij} = \begin{cases} 1 & \forall (i,j) \in \cup_{k=1}^{p} S_k \times S_k \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

For $n = 4$, the following illustrates the correspondence between a binary, psd matrix M and subsets S_k of $[n]$:

$$M = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix} \quad M = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$
$$S_1 = \{1,3\}, \quad S_2 = \{2,4\} \qquad S_1 = \{1,2\}, \quad S_2 = \{4\}.$$

Notice this characterization implies M is psd if and only if M is block-diagonal up to permutation, where each block equals a matrix of all ones or a matrix of all zeros.

B. Binary-psd-matrices and relations

The following corollary of Lemma 2 characterizes binary-psd-matrices in terms of *symmetric, transitive* relations:

Lemma 3: Let $M \in \mathbb{R}^{n \times n} \cap \{0,1\}^{n \times n}$ be the adjacency matrix of a relation $R \subseteq [n] \times [n]$, i.e., $M_{ij} = 1$ iff $(i,j) \in R$. The following statements are equivalent:

- The matrix M is symmetric and positive semidefinite.
- The relation R is symmetric and transitive.

This characterization is useful algorithmically; in particular, a basic step of our combinatorial algorithm will be computing the *transitive closure* of a relation. Also observe when R is symmetric and transitive, but not *reflexive*, at least one row and column of M is zero.

III. Invariant affine subspaces

This section characterizes binary matrices M that satisfy the inclusion $P_M(\mathcal{A}) \subseteq \mathcal{A}$, where $\mathcal{A}$ is an affine subspace of $\mathbb{S}^n$, i.e, it characterizes coordinate projections that satisfy condition (b) of Proposition 1. For concreteness, we define $\mathcal{A}$ as the solution set to $A(X) = b$, where $b \in \mathbb{R}^m$ and $A : \mathbb{S}^n \to \mathbb{R}^m$ is defined by $A_1, \ldots, A_m \in \mathbb{S}^n$ via $A(X) = (A_1 \cdot X, \cdots, A_m \cdot X)^T$. We also let $X^\star$ denote the solution to $A(X) = b$ with minimum Frobenius norm. Concretely:

$$A(X) := (A_1 \cdot X, \cdots, A_m \cdot X)^T, \tag{5}$$
$$\mathcal{A} := \{X \in \mathbb{S}^n : A(X) = b\}, \tag{6}$$
$$X^\star := \operatorname*{arg\,min}_{X \in \mathcal{A}} \|A\|_F. \tag{7}$$

Note with these definitions, $\mathcal{A} = X^\star + \ker A$.

A. General characterization

A general characterization of coordinate projections satisfying $P_M(\mathcal{A}) \subseteq \mathcal{A}$ arises from a general characterization of linear maps $D : \mathbb{S}^n \to \mathbb{S}^n$ with this property, given next by Lemma 4. To understand Lemma 4, it helps to view D as a matrix in $\mathbb{R}^{p \times p}$ and $\mathcal{A}$ as the set of solutions to $Ax = b$ for $A \in \mathbb{R}^{m \times p}$. With this notation, Lemma 4 generalizes a classical result that states the kernel of A is an invariant *linear* subspace of D if and only if $AD = TA$ for some $T \in \mathbb{R}^{m \times m}$ (Proposition 1.4.4, [12]). Specifically, Lemma 4 states $\mathcal{A}$ is an invariant *affine* subspace if and only if $AD = TA$ and, in addition, $Tb = b$.

Lemma 4: Let $\mathcal{A}$ be a non-empty, affine subspace defined as in (6) and let $D : \mathbb{S}^n \to \mathbb{S}^n$ be a linear map with adjoint $D^* : \mathbb{S}^n \to \mathbb{S}^n$. The following statements are equivalent.

1) $D(\mathcal{A}) \subseteq \mathcal{A}$.
2) There exists $T \in \mathbb{R}^{m \times m}$ for which

$$D^*(A_i) = \textstyle\sum_{j=1}^m T_{ij} A_j \quad \forall i \in [m]$$
$$Tb = b.$$

Direct application of Lemma 4 yields:

Theorem 1: Let $\mathcal{A}$ be a non-empty, affine subspace defined as in (6) and fix $M \in \mathbb{S}^n \cap \{0,1\}^{n \times n}$. The following statements are equivalent.

1) $P_M(\mathcal{A}) \subseteq \mathcal{A}$.
2) There exists $T \in \mathbb{R}^{m \times m}$ for which

$$M \circ A_i = \textstyle\sum_{j=1}^m T_{ij} A_j \quad \forall i \in [m]$$
$$Tb = b.$$

We now simplify Theorem 1 by assuming A_i and b_i have structured sparsity.

B. Characterization assuming disjoint support

Our next characterization assumes the A_i have *disjoint support*, i.e., $A_i \circ A_j = 0_{n \times n}$ for all $i \neq j$. This condition, while strong, frequently holds in SOS optimization (see, e.g., [2], where this condition is used to develop an efficient first-order method for unconstrained SOS optimization). It also leads to a much simpler characterization: in particular, the terms $T_{ij} A_j$ from Theorem 1 vanish for $i \neq j$:

Theorem 2: Let $\mathcal{A}$ be a non-empty, affine subspace defined as in (6), and suppose the A_i have disjoint support, i.e., suppose $A_i \circ A_j = 0_{n \times n}$ for all $i \neq j$. Fix $M \in \mathbb{S}^n \cap \{0,1\}^{n \times n}$. The following statements are equivalent.

1) $P_M(\mathcal{A}) \subseteq \mathcal{A}$.
2) $M \circ A_i \in \{A_i, 0_{n \times n}\}$, where $M \circ A_i = A_i$ if $b_i \neq 0$.

As Section IV-B.2 shows, Theorem 2 eliminates certain linear algebra computations from our combinatorial algorithm.

C. Characterization of projections

We now characterize orthogonal projections P satisfying $P(\mathcal{A}) \subseteq \mathcal{A}$. in terms of $\operatorname{rng} P$ and $X^\star$, the minimum-Frobenius-norm solution to $A(X) = b$:

Theorem 3: Let $\mathcal{A}$, A and $X^\star$ be defined as in (5)-(7), let $P_{\ker A}$ denote the orthogonal projection onto $\ker A$, and let $P : \mathbb{S}^n \to \mathbb{S}^n$ be an orthogonal projection. The following statements are equivalent.

1) $P(\mathcal{A}) \subseteq \mathcal{A}$.
2) $\operatorname{rng} P$ is an invariant subspace of $P_{\ker A}$ containing $X^\star$.

Our algorithm uses this characterization for general A_i.

IV. Combinatorial algorithm

We now give a simple combinatorial algorithm for finding the minimal-coordinate-projection of an SDP with cost matrix C and feasible set $\mathcal{A} \cap \mathbb{S}^n_+$—where Section I-C defined this projection as the minimizer of $\dim \operatorname{rng} P_M$ among coordinate projections P_M mapping $\mathbb{S}^n_+$ into $\mathbb{S}^n_+$, $\mathcal{A}$ into $\mathcal{A}$ and C to C. The algorithm is based on the following observations:

- For any $\mathcal{S} \subseteq \mathbb{S}^n$, the set of maps P_M satisfying $P_M(\mathcal{S}) \subseteq \mathcal{S}$ is closed under composition;
- For binary matrices M_1 and M_2, the composition of P_{M_1} and P_{M_2} equals $P_{M_1 \circ M_2}$, i.e., it is defined by the Hadamard product $M_1 \circ M_2$.

From these observations, we conclude the set of *supports* of matrices M_i satisfying $P_{M_i}(\mathcal{S}) \subseteq \mathcal{S}$ is closed under intersection, where the support of a binary matrix $M \in \mathbb{S}^n$ is the following subset of $[n] \times [n]$:

$$\operatorname{supp} M := \{(i,j) : M_{ij} = 1\}.$$

This intersection property allows us to define the following *closure operator* $\mathrm{cl}_{\mathcal{S}}(\cdot)$:

Definition 2: For $\mathcal{S} \subseteq \mathbb{S}^n$ and a binary matrix $M \in \mathbb{S}^n$, let $\{M_i\}$ denote the set of all binary matrices in $\mathbb{S}^n$ satisfying

$$P_{M_i}(\mathcal{S}) \subseteq \mathcal{S},$$
$$\operatorname{supp}(M_i) \supseteq \operatorname{supp}(M).$$

Then, $\mathrm{cl}_{\mathcal{S}}(M)$ is defined to be the unique binary matrix with support equal to $\bigcap_i \operatorname{supp}(M_i)$.

The coordinate projection $P_{\mathrm{cl}_{\mathcal{S}}(M)}$ has the following interpretation: among projections mapping $\mathcal{S}$ into $\mathcal{S}$ whose range contains $\operatorname{rng} P_M$, it has range of minimum dimension, since—by definition—it equals the composition of all coordinate projections with these properties. Based on this, our combinatorial algorithm finds the minimal-coordinate-projection using only evaluations of $\mathrm{cl}_{\mathcal{A}}(\cdot)$ and $\mathrm{cl}_{\mathbb{S}^n_+}(\cdot)$. It appears next in Algorithm 1:

Algorithm 1: Finds minimal-coordinate-projection P_M.

Initalize M to the zero matrix.
Set $M_{ij} = 1$ for all $(i,j) = \operatorname{supp}(C)$.
repeat
  Set $M = \mathrm{cl}_{\mathbb{S}^n_+}(M)$.
  Set $M = \mathrm{cl}_{\mathcal{A}}(M)$.
until converged;

Correctness follows from a few basic observations: if P_{M^*} is the minimal-coordinate-projection, then $P_{M^* \circ M} = P_M$ at each step of the algorithm, by definition of $\mathrm{cl}_{\mathbb{S}^n_+}(\cdot)$ and $\mathrm{cl}_{\mathcal{A}}(\cdot)$. In addition, the algorithm terminates only if $P_M(\mathcal{A}) \subseteq \mathcal{A}$, $P_M(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$, and $P_M(C) = C$—hence, at termination P_M must equal P_{M^*}. Moreover, it must terminate given any closure operator $\mathrm{cl}_{\mathcal{S}}(\cdot)$ is *extensive*—i.e., $\operatorname{supp}(M) \subseteq \operatorname{supp}(\mathrm{cl}_{\mathcal{S}}(M))$.

Remark 1: Viewing the set of all closure operators as a lattice (as in Section 3 of [4]), Algorithm 1 evaluates the *meet* of $\mathrm{cl}_{\mathbb{S}^n_+}$ and $\mathrm{cl}_{\mathcal{A}}$ at the binary matrix with support equal to $\operatorname{supp} C$. Note the meet is not (in general) equal to $\mathrm{cl}_{\mathcal{A} \cap \mathbb{S}^n_+}(\cdot)$. The follows given that projections may leave $\mathcal{A} \cap \mathbb{S}^n_+$ invariant but *not* $\mathbb{S}^n_+$ and $\mathcal{A}$.

A. Evaluating the closure operator $\mathrm{cl}_{\mathbb{S}^n_+}(\cdot)$

We describe evaluation of $\mathrm{cl}_{\mathbb{S}^n_+}(\cdot)$ by first recalling two facts:

- $P_M(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$ if and only if M is a binary-psd-matrix (Lemma 1);
- Binary-psd-matrices are in one-to-one correspondence with symmetric, transitive relations (Lemma 3).

From these facts, it follows that $\mathrm{cl}_{\mathbb{S}^n_+}(M)$ corresponds to the *transitive closure* of the relation defined by M. Formally:

$$M = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0^* \\ 0 & 1 & 0 & 0^* & 1 \end{pmatrix}$$

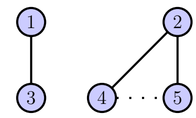

Fig. 1: Graphical illustration of $\mathrm{cl}_{\mathbb{S}^n_+}(\cdot)$

Proposition 2: Fix $M \in \mathbb{S}^n \cap \{0,1\}^{n \times n}$, let $R \subseteq [n] \times [n]$ denote a relation for which $(i,j) \in R$ iff $M_{ij} = 1$, and let $\overline{R}$ denote the transitive closure of R. Then,

$$[\mathrm{cl}_{\mathbb{S}^n_+}(M)]_{ij} = \begin{cases} 1 & \forall (i,j) \in \overline{R} \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Evaluation of $\mathrm{cl}_{\mathbb{S}^n_+}(M)$ also has a graphical interpretation if we assume $M_{ii} = 1$ for all i. Viewing M as an adjacency matrix of a graph on n nodes, $\mathrm{cl}_{\mathbb{S}^n_+}(M)$ is the adjacency matrix of the graph obtained by completing connected-components to complete graphs. As an example, consider the matrix M and corresponding graph in Figure 1. Here, the entries marked 0^* are set to one in $\mathrm{cl}_{\mathbb{S}^n_+}(M)$ and correspond to the dashed edge.

B. Evaluating the closure operator $\mathrm{cl}_{\mathcal{A}}(\cdot)$

To describe evaluation of $\mathrm{cl}_{\mathcal{A}}(\cdot)$, we employ the definition of $\mathcal{A}$ given by (6). We first make no assumptions on the matrices A_i. When then assume the A_i have disjoint support, i.e., $A_i \circ A_j = 0$ for $i \neq j$.

1) General case: The general case is based off Theorem 3, which states $P_M(\mathcal{A}) \subseteq \mathcal{A}$ if and only if the following two conditions hold:

- The range of P_M contains $X^\star$, the minimum-Frobenius-norm solution to $A(X) = b$;
- The range of P_M is an invariant subspace of $P_{\ker A}$, the orthogonal projection onto $\ker A$ with respect to the trace inner-product.

Since P_M is a coordinate projection, these conditions are equivalent to conditions on the support of M. Specifically, $P_M(\mathcal{A}) \subseteq \mathcal{A}$ if and only if the following conditions hold:

- The support of M contains the support of $X^\star$;
- If $(i,j) \in \operatorname{supp}(M)$, then $P_{\ker A}(E_{ij} + E_{ji})$ is as well, where E_{ij} is a standard basis vector of $\mathbb{R}^{n \times n}$.

This conditions yield a procedure for evaluating $\mathrm{cl}_{\mathcal{A}}(M)$:

Proposition 3: Let $\mathcal{A}$, A and $X^\star$ be defined as in (5)-(7) and let $P_{\ker A}$ denote the orthogonal projection onto $\ker A$. For a binary matrix $M \in \mathbb{S}^n$, the following holds:

$$[\mathrm{cl}_{\mathcal{A}}(M)]_{ij} = \begin{cases} 1 & \forall (i,j) \in S_N \\ 0 & \text{otherwise,} \end{cases}$$

where $S_0 := \operatorname{supp}(X^\star) \cup \operatorname{supp}(M)$,

$$S_{k+1} := S_k \bigcup \left(\bigcup_{(i,j) \in S_k} \operatorname{supp} P_{\ker A}(E_{ij} + E_{ji}) \right),$$

and N is any integer for which $S_N = S_{N+1}$.

2) With disjoint support: Now suppose the matrices $A_1, \dots, A_m$ defining the map $A : \mathbb{S}^n \to \mathbb{R}^m$ have disjoint support. By Theorem 2, the following statements are equivalent:

1) $P_M(\mathcal{A}) \subseteq \mathcal{A}$.
2) $M \circ A_i \in \{A_i, 0_{n\times n}\}$, where $M \circ A_i = A_i$ if $b_i \neq 0$.

From these statements, a procedure for evaluating $\text{cl}_{\mathcal{A}}(M)$ easily arises that does not require access to $P_{\ker A}$ or $X^\star$. In particular, $\text{cl}_{\mathcal{A}}(M)$ is determined only by the sparsity patterns of M, A_i and b. Formally:

Proposition 4: Let $\mathcal{A}$ be defined as in (6), and assume the matrices $A_1, \dots, A_m$ have disjoint support, i.e., $A_i \circ A_j = 0$ for all $i \neq j$. For a binary matrix $M \in \mathbb{S}^n$, let $\mathcal{T} := \{i \in [m] : M \circ A_i \notin \{A_i, 0_{n\times n}\}\}$,

$$S_1 := \bigcup_{i\in\mathcal{T}} \text{supp}\, A_i, \quad \text{and} \quad S_2 := \bigcup_{i:b_i\neq 0} \text{supp}(A_i).$$

The following holds:

$$[\text{cl}_{\mathcal{A}}(M)]_{ij} = \begin{cases} 1 & \forall (i,j) \in S_1 \cup S_2 \cup \text{supp}(M) \\ 0 & \text{otherwise.} \end{cases}$$

V. Computational Results

We now present examples illustrating effectiveness of our technique. For each example, we apply Algorithm 1 to find the minimal-coordinate-projection P_M and report the sizes of the psd cones in the resulting block-diagonalizations. If, for instance,

$$M = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix},$$

we report block-sizes $(2, 1, 1)$. For the examples of V-A and V-C, the data matrices have disjoint support, i.e., $A_i \circ A_j = 0$ for $i \neq j$; hence, evaluating the closure operator $\text{cl}_{\mathcal{A}}$ in Algorithm 1 be be simplified per Section IV-B.2.

A. Copositivity of quadratic forms

This example pertains to SDPs that demonstrate *copositivity* of certain quadratic forms. A quadratic form $x^T Jx$ is copositive if and only if $x^T Jx \geq 0$ for all x in the non-negative orthant. Deciding copositivity is $\mathcal{NP}$-hard, but a sufficient condition can be checked using sum-of-squares techniques and semidefinite programming. To see this, note a polynomial $f(x_1, \dots, x_p)$ is copositive if $f(x_1^2, \dots, x_p^2)$ is globally non-negative. In turn, $f(x_1^2, \dots, x_p^2)$ is non-negative if, for instance, $\left(\sum_{i=1}^p x_i^2\right) f(x_1^2, \dots, x_p^2)$ is a sum-of-squares—a condition that can be checked with SDP.

Using this approach, we formulate SDPs for demonstrating copositivity of the parametrized quadratic form $B(x; m) := \left(\sum_{i=1}^{3m+2} x_i\right)^2 - 2\sum_{i=1}^{3m+2} x_i \sum_{j=0}^{m} x_{i+3j+1}$, where m is an integer parameter and the subscript for x wraps cyclically, i.e., $x_{r+n} = x_r$. Note $B(x; m)$ was studied in [1] and, for $m = 1$, equals the so-called Horn form.

For these SDPs, Algorithm 1 yields dramatic reductions, both in problem size and solve time (Table I). Indeed, for $m \geq 4$, the SDPs are otherwise unsolvable on a modern desktop with 16 GBs of RAM. Crucially, executing Algorithm 1 is also inexpensive relative to solve time, taking less than a second on each example. Finally, these SDPs can also be simplified using *facial reduction* techniques, as demonstrated in [22]. Combining the proposed method with these techniques simplifies these SDPs even further.

	Before	After	T_s Before	T_s After
$B(x;1)$	35	5×5, 1×10	.5	0.2
$B(x;2)$	120	8×8, 1×56	10.8	.74
$B(x;3)$	286	11×11, 1×165	3589	0.3
$B(x;4)$	560	14×14, 1×364	OOM	.76
$B(x;5)$	969	17×17, 1×680	OOM	2.3

TABLE I: Sizes of psd constraints and solve times before and after simplifications (Section V-A). The notation $n \times m$ indicates m semidefinite constraints on matrices in $\mathbb{S}^n$. Solve time T_s is shown in seconds for SeDuMi [24] and OOM indicates an out-of-memory error.

B. SOSTOOLS and SOSOPT demonstrations

We consider SDPs constructed by demo scripts packaged with SOSTOOLS [19] and the SOS analysis tools available at

$$\texttt{http://www.aem.umn.edu/\~{}AerospaceControl/},$$

which include SOSOPT [23]. Many of these demos solve SDPs demonstrating stability of nonlinear dynamical systems. Table II illustrates the potentially-broad applicability of our method, showing reductions for several of these SDPs. Note many of these scripts construct several SDPs; with the exception of IOGainDemo_(1|3), reported results are for the *first* SDP constructed. For this exception, SDPs come from the so-called V-s iteration of the script. For some examples, it was also necessary to eliminate free variables to put SDPs in the form (1).

C. Comparison with method of Dai and Xia

Our final example compares performance of Algorithm 1 to a simplification strategy described in [5]. Table III illustrates improvement over this strategy for SOS-based proofs of the *monotone column permanent conjecture* ([13], Conjecture 2). These proofs show particular polynomials $p_{i,j}$ are sums-of-squares, where definitions of $p_{i,j}$ can be found in [5] and references therein. Table III compares block-diagonalizations by reporting sizes of the largest blocks.

VI. Implementation

An implementation of the proposed method is integrated into `frlib`, a set of MATLAB tools for SDP pre-processing introduced by the authors in [22]. Both `frlib` and SeDuMi files for the examples of this paper are available at `www.mit.edu/~fperment`.

Script Name	Before	After
sosdemo2	$13, 3$	$3, 2 \times 3, 1 \times 7$
sosdemo4	35	$5 \times 5,\ 1 \times 10$
sosdemo6	$21, 6$	$16,\ 5 \times 2,\ 1$
sosoptdemo2	$13, 3$	$3,\ 2 \times 3,\ 1 \times 7$
sosoptdemo4	35	$5 \times 5,\ 1 \times 10$
gsosoptdemo1	$9, 5$	$6,\ 3 \times 2,\ 2$
IOGainDemo_1	$8, 3$	$5,\ 3 \times 2$
IOGainDemo_3	$15, 8$	$10, 5 \times 2, 3$
Chesi(1|2)_IterationWithVlin	$9, 5$	$6, 3 \times 2, 2$
Chesi3_GlobalStability	$14, 5$	$8, 6, 3, 2$
Chesi(3|4)_IterationWithVlin	$9, 5$	$6, 3 \times 2, 2$
Chesi(5|6)_Bootstrap	$19, 9$	$13, 6 \times 2, 3$
Chesi(5|6)_IterationWithVlin	$19, 9$	$13, 6 \times 2, 3$
Coutinho3_IterationWithVlin	$9, 5$	$6, 3 \times 2, 2$
HachichoTibken_Bootstrap	$19, 9$	$12, 7, 6, 3$
HachichoTibken_IterationWithVlin	$19, 9$	$12, 7, 6, 3$
Hahn_IterationWithVlin	$9, 5$	$6, 3, 3, 2$
KuChen_IterationWithVlin	$19, 9$	$13, 6 \times 2, 3$
Parrilo1_GlobalStabilityWithVec	$3, 2$	$2, 1 \times 3$
Parrilo2_GlobalStabilityWithMat	$3, 2$	$2, 1 \times 3$
Pendubot_IterationWithVlin	$14, 4$	$10, 4 \times 2$
VDP_IterationWithVball	$5, 4$	$3 \times 2, 2, 1$
VDP_IterationWithVlin	$9, 5$	$6, 3 \times 2, 2$
VDP_LinearizedLyap	$9, 5$	$6, 3 \times 2, 2$
VDP_MultiplierExample	$5, 2$	$3, 2, 1 \times 2$
VannelliVidyasagar2_Bootstrap	$19, 9$	$13, 6 \times 2, 3$
VannelliVidyasagar2_IterationWithVlin	$19, 9$	$13, 6 \times 2, 3$
VincentGrantham_IterationWithVlin	$9, 5$	$6, 3 \times 2, 2$
WTBenchmark_IterationWithVlin	$19, 9$	$13, 6 \times 2, 3$

TABLE II: Sizes of psd constraints before and after simplifications (Section V-B). The notation $n \times m$ indicates m semidefinite constraints on matrices in $\mathbb{S}^n$.

	Proposed	(Dai, Xia '14) [5]
$p_{1,2}$	15	77
$p_{1,3}$	8	15
$p_{2,2}$	12	62
$p_{2,3}$	10	39

TABLE III: Comparison of largest block-size for SDPs described in Section V-C.

VII. Conclusion

We have given a technique for simplifying SDPs and illustrated its effectiveness on examples arising in SOS optimization. The proposed technique is easily combined with other simplification methods, e.g., [20] and [21], to yield even smaller SDPs. Algebraic interpretations of the method and proofs will be given in a full version of this paper. Finally, a common generalization of the presented method and techniques based on coherent configurations [6] is possible. This generalization will be explored in a forthcoming paper.

References

[1] Victor Baston. Extreme copositive quadratic forms. *Acta Arithmetica*, 15(3):319–327, 1969.

[2] Dimitris Bertsimas, Robert M. Freund, and Xu A. Sun. An accelerated first-order method for solving SOS relaxations of unconstrained polynomial optimization problems. *Optimization Methods Software*, 28(3):424–441, June 2013.

[3] Grigoriy Blekherman, Pablo A. Parrilo, and Rekha R. Thomas. *Semidefinite optimization and convex algebraic geometry*. SIAM, 2013.

[4] Nathalie Caspard and Bernard Monjardet. The lattices of closure systems, closure operators, and implicational systems on a finite set: a survey. *Discrete Applied Mathematics*, 127(2):241–269, 2003.

[5] Liyun Dai and Bican Xia. Smaller SDP for SOS decomposition. *arXiv preprint arXiv:1407.2679*, 2014.

[6] Etienne de Klerk. Exploiting special structure in semidefinite programming: A survey of theory and applications. *European Journal of Operational Research*, 201(1):1–10, 2010.

[7] Etienne de Klerk, Dmitrii Pasechnik, and Alexander Schrijver. Reduction of symmetric semidefinite programs using the regular $*$-representation. *Mathematical Programming*, 109(2-3):613–624, 2007.

[8] Dmitriy Drusvyatskiy, Gábor Pataki, and Henry Wolkowicz. Coordinate shadows of semi-definite and euclidean distance matrices. *arXiv preprint arXiv:1405.2037*, 2014.

[9] Igor Dukanovic and Franz Rendl. Semidefinite programming relaxations for graph coloring and maximal clique problems. *Mathematical Programming*, 109(2-3):345–365, 2007.

[10] Mituhiro Fukuda, Masakazu Kojima, Kazuo Murota, and Kazuhide Nakata. Exploiting sparsity in semidefinite programming via matrix completion I: General framework. *SIAM Journal on Optimization*, 11(3):647–674, 2001.

[11] Karin Gatermann and Pablo A. Parrilo. Symmetry groups, semidefinite programs, and sums of squares. *Journal of Pure and Applied Algebra*, 192(1–3):95–128, 2004.

[12] Israel Gohberg, Peter Lancaster, and Leiba Rodman. *Invariant subspaces of matrices with applications*, volume 51. SIAM, 1986.

[13] James Haglund, Ken Ono, and David G Wagner. Theorems and conjectures involving rook polynomials with only real zeros. In *Topics in number theory*, pages 207–221. Springer, 1999.

[14] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[15] Adam N. Letchford and Michael M. Sørensen. Binary positive semidefinite matrices and associated integer polytopes. *Mathematical Programming*, 131(1-2):253–271, 2012.

[16] Johan Löfberg. Pre-and post-processing sum-of-squares programs in practice. *IEEE Transactions on Automatic Control*, 54(5):1007–1011, 2009.

[17] Daniele Mundici. *Advanced Lukasiewicz calculus and MV-algebras*, volume 35. Springer Science & Business Media, 2011.

[18] Jiawang Nie and James Demmel. Sparse sos relaxations for minimizing functions that are summations of small polynomials. *SIAM Journal on Optimization*, 19(4):1534–1558, 2008.

[19] Antonis Papachristodoulou, James Anderson, Giorgio Valmorbida, Stephen Prajna, Pete Seiler, and Pablo Parrilo. SOSTOOLS version 3.00 sum of squares optimization toolbox for MATLAB. *arXiv preprint arXiv:1310.4716*, 2013.

[20] Frank Permenter and Pablo A. Parrilo. Selecting a monomial basis for sums of squares programming over a quotient ring. In *IEEE 51st Annual Conference on Decision and Control (CDC)*, pages 1871–1876. IEEE, 2012.

[21] Frank Permenter and Pablo A. Parrilo. Basis selection for sos programs via facial reduction and polyhedral approximations. In *Proceedings of the IEEE Conference on Decision and Control*, 2014.

[22] Frank Permenter and Pablo A. Parrilo. Partial facial reduction: simplified, equivalent SDPs via inner approximations of the PSD cone. http://arxiv.org/abs/1408.4685, 2014.

[23] Peter Seiler. SOSOPT: A toolbox for polynomial optimization. *arXiv preprint arXiv:1308.1889*, 2013.

[24] Jos F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999.

[25] Frank Vallentin. Symmetry in semidefinite programs. *Linear Algebra and Its Applications*, 430(1):360–369, 2009.

[26] Hayato Waki, Sunyoung Kim, Masakazu Kojima, and Masakazu Muramatsu. Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM Journal on Optimization*, 17(1):218–242, 2006.