Jordan symmetry reduction for conic optimization over the doubly nonnegative cone: Theory and software

Brosch, Daniel; de Klerk, Etienne

Published in:
Optimization Methods and Software

Publication date:
2021

Document Version
Peer reviewed version

Jordan symmetry reduction for conic optimization over the doubly nonnegative cone: theory and software

Daniel Brosch, Etienne de Klerk

Tilburg University

ARTICLE HISTORY
Compiled September 28, 2021

ABSTRACT
A common computational approach for polynomial optimization problems (POPs) is to use (hierarchies of) semidefinite programming (SDP) relaxations. When the variables in the POP are required to be nonnegative — as is the case for combinatorial optimization problems, for example — these SDP problems typically involve nonnegative matrices, i.e. they are conic optimization problems over the doubly nonnegative cone. The Jordan reduction, a symmetry reduction method for conic optimization, was recently introduced for symmetric cones by Parrilo and Permenter [Mathematical Programming 181(1), 2020]. We extend this method to the doubly nonnegative cone, and investigate its application to known relaxations of the quadratic assignment and maximum stable set problems. We also introduce new Julia software where the symmetry reduction is implemented.

KEYWORDS
Quadratic assignment problem; Maximum stable set; Semidefinite programming; Symmetry reduction

AMS CLASSIFICATION
90C22; 20B40

1. Introduction

This paper studies symmetry reduction of semidefinite programs (SDPs) where the matrix variable is also entry-wise nonnegative, i.e. symmetry reduction of conic linear programming over the doubly nonnegative cone. Such problems appear naturally, among others, in the study of convex relaxations of combinatorial problems. In particular, we are interested in such relaxations of the independence number of a graph, and of the quadratic assignment problem (QAP).

The independence number α of a graph G is the maximum number of nodes of G we can choose, such that there is no edge between any of them. The Theta-Prime function (ϑ'-function) [21] is a semidefinite programming relaxation of α, and as such gives an upper bound to it. Given the adjacency matrix A of G, the function is defined by

$$\begin{aligned} \vartheta'(G) = \sup\ & \langle J, X\rangle \\ \text{s.t.}\ & \operatorname{trace}(X) = 1, \\ & \langle A, X\rangle = 0, \\ & X \geq 0, \\ & X \succcurlyeq 0. \end{aligned} \tag{1.1}$$

The second family of problems we are interested are quadratic assignment problems, which are of the form

$$QAP(A, B) = \min_{\varphi \in S_n} \sum_{i,j=1}^{n} a_{ij} b_{\varphi(i)\varphi(j)}, \tag{1.2}$$

where $A = (a_{ij})$ and $B = (b_{ij})$ are symmetyric $n \times n$ matrices, and S_n denotes the symmetric group (i.e. all permutations) on n elements. Here we are interested in the SDP relaxation of by Zhao et al. [24], as reformulated by Povh and Rendl [20].

Symmetry reduction for SDP was first introduced by Schrijver in 1979 in [21]; see for example the chapter [2] by Bachoc, Gijswijt, Schrijver and Vallentin for a review of later developments up to 2012. The specific case of SDP relaxations of quadratic assignment problems was investigated by De Klerk and Sotirov in [7, 8].

Parrilo and Permenter [18] recently introduced a new — and more general — form of symmetry reduction, called *Jordan reduction*. We will extend their approach to the doubly-nonnegative cone. The advantage of the Jordan reduction approach is that it requires no knowledge of group symmetries in the data, and therefore is ideal for automated pre-processing. A drawback is that the initial problem size must be small enough to perform basic linear algebra operations. This is not always the case, e.g. when computing the ϑ' function of Hamming graphs [21].

Outline and Contributions of this Paper

In the next section, we recap relevant definitions and results on the Jordan reduction of Parrilo and Permenter [18]. In Section 3 we subsequently extend this approach — which was formulated for symmetric cones — to the doubly nonnegative cone. This allows us to apply the method to the SDP relaxation of the general QAP due to Zhao et al. [24] in Section 4, and to the ϑ'-function of Erdős-Rényi -graphs in Section 5.1. Our extension of the Jordan reduction method of Parrilo and Permenter [18] in Section 3 should lead to additional applications in SDP relaxations of other combinatorial problems. Finally, in Section 6, we describe a Julia software package implementing this method.

This package complements the two existing packages QDimSum [22] and YALMIP [14], of which the first exploits the symmetry of semidefinite programming problems coming from quantum mechanics, and the second allows for symmetry reduction of polynomial optimization problems with sign symmetries.

2. Preliminaries on Jordan Symmetry Reduction

We will study conic optimization problems in the form

$$\left.\begin{array}{lll} \inf & \langle C, X\rangle & \\ \text{s.t.} & \langle A_i, X\rangle = b_i \text{ for } i \in [m] \\ & X \in \mathcal{K} \end{array} \quad \begin{array}{lll} = \inf & \langle C, X\rangle \\ \text{s.t.} & X \in X_0 + \mathcal{L} \\ & X \in \mathcal{K}, \end{array}\right\} \tag{2.1}$$

where $[m] = \{1, \ldots, m\}$, $\mathcal{K} \subseteq \mathcal{V}$ is a closed, convex cone in a real Hilbert space $\mathcal{V}$, $X_0 \in \mathcal{V}$ satisfies $\langle A_i, X_0\rangle = b_i$ for all $i \in [m]$, and $\mathcal{L} \subseteq \mathcal{V}$ is the nullspace of the linear operator A, where $A(X) = (\langle A_i, X\rangle)_{i=1}^m$. The objective function is given using the inner product $\langle\cdot,\cdot\rangle$ of $\mathcal{V}$, with which one defines the *dual cone* as:

$$\mathcal{K}^* := \{Y \in \mathcal{V} \mid \langle X, Y\rangle \geq 0 \ \ \forall X \in \mathcal{K}\}.$$

In this paper, we will mostly deal with the case where $\mathcal{V}$ is the space $\mathbb{S}^n$ of $n \times n$ symmetric matrices equipped with the Euclidean inner product, and where $\mathcal{K}$ is the cone of doubly nonnegative matrices.

2.1. Constraint Set Invariance

Parrilo and Permenter [18] introduced a set of three conditions a subspace has to fulfill, such that it is be possible to use it for symmetry reduction. Here we revisit some of their results.

Definition 2.1. A *projection* is a linear transformation $P\colon \mathcal{V} \to \mathcal{V}$ which is *idempotent*, i.e. $P^2 = P$.

Definition 2.2 (Definition 2.1. in [18])**.** A projection $P\colon \mathcal{V} \to \mathcal{V}$ fulfills the *Constraint Set Invariance Conditions (CSICs)* for $(\mathcal{K}, X_0 + \mathcal{L}, C)$ if

(i) The projection is positive: $P(\mathcal{K}) \subseteq \mathcal{K}$,
(ii) $P(X_0 + \mathcal{L}) \subseteq X_0 + \mathcal{L}$,
(iii) $P^*(C + \mathcal{L}^\perp) \subseteq C + \mathcal{L}^\perp$,

where P^* is the adjoint of P, which satisfies $\langle P(X), Y\rangle = \langle X, P^*(Y)\rangle$ for all $X, Y \in \mathcal{V}$.

Note that this definition is symmetric going from primal to dual, since

$$P(\mathcal{K}) \subseteq \mathcal{K} \quad \Leftrightarrow \quad P^*(\mathcal{K}^*) \subseteq \mathcal{K}^*.$$

While it is clear that such a projection has to send feasible solutions to feasible solutions in both the primal and dual program, it seems less obvious that it does not change the objective value of feasible elements (Proposition 1.4.1 in [19]). We give an alternative, more compact proof of this fact. For an $X \in X_0 + \mathcal{L}$ we have $X - P(X) \in \mathcal{L}$ by *(ii)*, and *(iii)* tells us $C - P^*(C) \in \mathcal{L}^\perp$. Thus

$$\begin{aligned} \langle C, X\rangle - \langle C, P(X)\rangle &= \langle C, X - P(X)\rangle - \langle C, P(X) - P(P(X))\rangle \\ &= \langle C, X - P(X)\rangle - \langle P^*(C), X - P(X)\rangle \\ &= \langle C - P^*(C), X - P(X)\rangle = 0. \end{aligned}$$

To make things easier, we restrict ourselves to *orthogonal* projections P_S to a subspace $S \subseteq \mathcal{V}$, which are exactly the projections of which the range and kernel are orthogonal, or equivalently the projections which are self-adjoint, i.e. $P_S = P_S^*$. If the projection P_S fulfills the CSICs we call the subspace S *admissible*, following [18]. In this case, the CSICs may be rewritten as follows, exploiting the fact that orthogonal projections have the property that $P_A(B) \subseteq B \Leftrightarrow P_B(A) \subseteq A$.

Theorem 2.3 (Theorem 5.2.1 in [19])**.** *Consider the conic optimization problem* (2.1) *and let $S \subseteq \mathcal{V}$ be the range of an orthogonal projection $P_S : \mathcal{V} \to \mathcal{V}$. Let $P_{\mathcal{L}}$ denote the orthogonal projection onto $\mathcal{L}$, and define $C_{\mathcal{L}} = P_{\mathcal{L}}(C)$ and $X_{0,\mathcal{L}^\perp} = P_{\mathcal{L}^\perp}(X_0)$. Then S is an admissible subspace if, and only if,*

(a) $C_{\mathcal{L}}, X_{0,\mathcal{L}^\perp} \in S$,
(b) $P_{\mathcal{L}}(S) \subseteq S$,
(c) $P_S(\mathcal{K}) \subseteq \mathcal{K}$.

Restricting the conic program to an admissible subspace S thus results in another, potentially significantly smaller program, with the same optimal value.

$$\begin{aligned} \inf\ & \langle P_S(C), X\rangle \\ \text{s.t.}\ & X \in P_S(X_0) + \mathcal{L} \cap S, \\ & X \in \mathcal{K} \cap S. \end{aligned}$$

2.2. The Reduction for Jordan-Algebras

Next we review some results from [19] for the situation where the space $\mathcal{V}$ is an *Euclidian Jordan algebra* $\mathcal{J}$, that is a commutative algebra (with product denoted by '$\circ$') over $\mathbb{R}$ satisfying the *Jordan identity*

$$(x \circ y) \circ x^2 = x \circ (y \circ x^2),$$

and an inner product with $\langle x \circ y, z\rangle = \langle y, x \circ z\rangle$. For every such algebra we can define $\mathcal{K}$ as the cone of squares of $\mathcal{J}$ given by $\mathcal{K} = \{x \circ x \mid x \in \mathcal{J}\}$, which always is a symmetric cone, i.e. a self-dual and homogenous convex cone (see for example [9]).

The only example relevant for us is the case $\mathcal{J} = \mathbb{S}^n$, the symmetric $n \times n$-matrices with real entries, with product defined by

$$X \circ Y := \frac{1}{2}(XY + YX),$$

and the inner product the Euclidean (trace) inner product $\langle X, Y\rangle = \operatorname{trace}(XY)$. It is easy to see (e.g. from the spectral decomposition) that its cone of squares is exactly the positive semidefinite cone $\mathbb{S}^n_+$.

Since the product of a Jordan algebra is commutative, we have

$$2x \circ y = x \circ y + y \circ x = (x + y)^2 - x^2 - y^2,$$

which means that subspaces are closed under multiplication, if and only if they include all squares. Similarly isomorphisms between (euclidian) Jordan algebras are exactly the bijective linear maps satisfying $\phi(x^2) = (\phi(x))^2$.

Definition 2.4. A Jordan algebra $\mathcal{J}$ is called *special*, if it is isomorphic to the algebra one gets from a real associative algebra by equipping the latter with the product $x \circ y = \frac{1}{2}(xy + yx)$.

There is only a single (up to isomorphisms) simple Jordan algebra which is not special, the algebra of Hermitian 3×3-matrices of Octonions $H_3(\mathbb{O})$. The for us relevant case $\mathcal{J} = \mathbb{S}^n$ is special.

Definition 2.5. A subspace (not necessarily a subalgebra) S of a Jordan algebra is called *unital*, if there is an an element $e \in S$ such that $e \circ a = a \circ e = a$ for all $a \in S$.

An important fact for us is that every Euclidean Jordan algebra is unital.

One main result of [19, Theorem 5.2.3] is an alternative description of the CSICs when the ambient space is a special Euclidean Jordan algebra. In this case the condition $P_S(\mathcal{K}) \subseteq \mathcal{K}$ in Theorem 2.3 — with $\mathcal{K}$ the cone of squares in $\mathcal{J}$ — is equivalent to S being closed under taking squares, i.e. to S being a Jordan sub-algebra of $\mathcal{J}$, if S is unital.

This gives an algorithm for finding the minimal admissible subspace, which is defined as follows.

Definition 2.6. The unique minimal admissible subspace is

$$S_{\min} := \bigcap_{S \text{ is admissible}} S.$$

As mentioned before, we may now formulate an algorithm for $S_{\min}$.

Theorem 2.7 (Theorem 3.2 in [18])**.** *If $\mathcal{V} = \mathcal{J}$ is an Euclidian, special Jordan algebra, and $\mathcal{K}$ its cone of squares, then $S_{\min}$ is the output of Algorithm 1.*

Algorithm 1: Finding $S_{\min}$

```
1 S ← span{C_L, X_{0,L⊥}}
2 repeat
3 |  S ← S + P_L(S)
4 |  S ← S + span{X^2 | X ∈ S}
5 until converged;
```

2.3. A combinatorial Reduction Algorithm

The fourth step of Algorithm 1 is not linear, and hard to implement. But, conveniently, Permenter does introduce three combinatorial algorithms in his PhD thesis ([19], Chapter 7) for the cone $\mathbb{S}^n_+$, which all find orthogonal 0/1-bases for an optimal unital admissible subspace with certain additional properties. A 0/1-basis is a basis where each element has entries solely in $\{0, 1\}$. If the basis is orthogonal, this implies that no two basis elements have nonzero entries in the same position. Here we will only mention one of the algorithms, since the other ones cannot give us better reductions for our special case.

Partition Subspaces

The second combinatorial algorithm by Permenter [19] finds an optimal unital *partition subspace*, which is a subspace with 0/1-basis, the elements of which sum to the all-one matrix. We call these basis elements *characteristic matrices* of the partition. We can describe the basis uniquely with a partition of the coordinates of $\mathbb{R}^{n\times n}$, i.e. of $[n]\times[n]$, simply by having one part in the partition for every basis element with ones in the corresponding coordinates. For example the following spaces are partition spaces:

$$P_1 = \begin{pmatrix} a & a & b \\ a & a & b \\ b & b & c \end{pmatrix}, \qquad P_2 = \begin{pmatrix} a & b & b \\ b & a & b \\ b & b & c \end{pmatrix}, \qquad P_3 = P_1 \wedge P_2 = \begin{pmatrix} a & b & c \\ b & a & c \\ c & c & d \end{pmatrix},$$

where $P_1 \wedge P_2$ is the coarsest (i.e. smallest dimensional) partition space *refining* both P_1 and P_2. A partition space A *refines* the partition space B, if B is a subspace of A.

For our purposes an important special case is a so-called Jordan configuration, defined as follows.

Definition 2.8. A partition P of $A \times A$, where A is a finite set, is called a *Jordan configuration*, if its characteristic matrices $\mathcal{B}_P$ satisfy

- $X = X^T$ for all $X \in \mathcal{B}_P$,
- $XY + YX \in \text{span } \mathcal{B}_P$ for all $X, Y \in \mathcal{B}_P$,
- $I \in \text{span } \mathcal{B}_P$.

In words, a Jordan configuration is a basis of a unital partition space that is also a Jordan subalgebra of $\mathbb{S}^n$.

A more general example of a partition space, also of interest to us, is a so-called coherent algebra (see e.g. [6]).

Definition 2.9. A partition P of $A \times A$, where A is a finite set, is called a *coherent configuration*, if its characteristic matrices $\mathcal{B}_P$ satisfy

- If $X \in \mathcal{B}_P$ then also $X^T \in \mathcal{B}_P$,
- $XY \in \text{span } \mathcal{B}_P$ for all $X, Y \in \mathcal{B}_P$,
- $I \in \text{span } \mathcal{B}_P$.

Thus, a coherent configuration gives a 0/1 basis of a partition subspace that is also a matrix $*$-algebra, namely the associated coherent algebra. Note that the symmetric part of a coherent configuration is a Jordan configuration. It was recently shown in [17] that the converse is not true, there are infinite Jordan configurations that are not the symmetric part of a coherent configuration.

To restrict the algorithm 1 to partition subspaces, we need more notation: $\text{part}(A)$ is the smallest partition space containing the matrix (or subspace) A, which is simply the partition space given by the unique entries of A.

There are two basic ways to implement this algorithm: One can use polynomial matrices, or randomization. For the first variant one introduces (commuting) variables t_i for each element of a basis $B_1, \ldots, B_k$ of P, and then refines the partition with $\text{part}(P_{\mathcal{L}}(\sum_{i=1}^k t_i B_i)) = \text{part}(\sum_{i=1}^k t_i P_{\mathcal{L}}(B_i))$ and $\text{part}((\sum_{i=1}^k t_i B_i)^2)$. If we for example

Algorithm 2: Partition algorithm ([19])

$P \leftarrow \mathrm{part}(C_{\mathcal{L}}) \wedge \mathrm{part}(X_{0,\mathcal{L}^\perp})$
repeat
  $P \leftarrow P \wedge \mathrm{part}(P_{\mathcal{L}}(P))$
  $P \leftarrow P \wedge \mathrm{part}(\mathrm{span}\{X^2 \mid X \in P\})$
until *converged*;

take P_2 from the example above, one has

$$\begin{pmatrix} t_a & t_b & t_b \\ t_b & t_a & t_b \\ t_b & t_b & t_c \end{pmatrix}^2 = \begin{pmatrix} t_a^2 + 2t_b^2 & 2t_a t_b + t_b^2 & t_a t_b + t_b^2 + t_b t_c \\ 2t_a t_b + t_b^2 & t_a^2 + 2t_b^2 & t_a t_b + t_b^2 + t_b t_c \\ t_a t_b + t_b^2 + t_b t_c & t_a t_b + t_b^2 + t_b t_c & 2t_b^2 + t_c^2 \end{pmatrix},$$

of which the unique polynomials induce the partition P_3.

The second variant refines the partition with a random element in the partition space after projecting it to $\mathcal{L}$ and after squaring it. While one has to be more careful about rounding errors here, it is both easier to implement and much faster.

Algorithm 3: Partition algorithm, randomized ([19])

$P \leftarrow \mathrm{part}(C_{\mathcal{L}}) \wedge \mathrm{part}(X_{0,\mathcal{L}^\perp})$
repeat
  $X \leftarrow$ random element of P
  $P \leftarrow P \wedge \mathrm{part}(P_{\mathcal{L}}(X))$
  $P \leftarrow P \wedge \mathrm{part}(X^2)$
until *converged*;

Remark 2.10. We note that the first variant of the partition algorithm presented here is very similar to the Weisfeiler-Leman (WL) algorithm [23], that finds the coarsest coherent configuration refining a given partition of $[n] \times [n]$. The only difference is that the WL algorithm uses non-commuting variables t_i, as opposed to commuting ones; see [1] on details of the implementation of the WL algorithm.

3. Extension to the doubly nonnegative Cone

We will now fix the cone $\mathcal{K}$ in (2.1) to be the *doubly nonnegative cone* $\mathcal{D}^n := \mathbb{S}^n_+ \cap \mathbb{R}^{n\times n}_+$. Since we will refer to nonnegative, symmetric matrices frequently, we also introduce the notation $\mathcal{N}^n = \mathbb{S}^n \cap \mathbb{R}^{n\times n}_+$. Even though $\mathcal{D}^n$ is not a cone of squares in a Euclidean Jordan algebra, one may readily adapt some of the results of the last section to this setting.

We start with an elementary, but important observation.

Proposition 3.1. *Assume that a subspace $S \subset \mathbb{S}^n$ has a basis of nonnegative matrices with pairwise disjoint supports. Then the orthogonal projection P_S onto S satisfies $P_S(\mathcal{D}^n) \subseteq \mathcal{D}^n$ if it satisfies $P_S(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$.*

Proof. If S has a basis of nonnegative matrices with disjoint supports, then it has an orthonormal basis with this property, say A_i ($i \in [d]$), and the orthogonal projection is of the form

$$P_S(X) = \sum_{i=1}^{d} \langle A_i, X \rangle A_i.$$

Since the Euclidean inner product of two nonnegative matrices is nonnegative, we have

$$P_S(\mathcal{N}^n) \subseteq \mathcal{N}^n,$$

and, since $\mathcal{D}^n \subset \mathbb{S}^n_+$, and $P_S(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$ by assumption,

$$P_S(\mathcal{D}^n) \subseteq \mathbb{S}^n_+ \cap \mathcal{N}^n = \mathcal{D}^n.$$

□

If we consider partition subspaces, we may therefore use results on admissible partition subspaces for the case $\mathcal{K} = \mathbb{S}^n_+$, as follows.

Corollary 3.2. *Consider a conic optimization problem of the form* (2.1)*, with $\mathcal{V} = \mathbb{S}^n$, and $\mathcal{K} = \mathbb{S}^n_+$, and let S be a admissible partition subspace for this problem. Then, S is also an admissible partition subspace for the related problem where we replace $\mathcal{K} = \mathbb{S}^n_+$ by $\mathcal{K} = \mathcal{D}^n$.*

The important practical implication is that we may use Algorithm 3 to find an admissible Jordan configuration for conic optimization problems on the cone $\mathcal{D}^n$ (but we do not know if it is optimal in general). In the next section we will do precisely this for an SDP relaxation of the quadratic assignment problem.

It it instructive though to ask how restrictive it is to only consider admissible partition subspaces. In what follows, we show that the partition subspace structure is actually imposed by some relatively weak assumptions.

To this end, we first recall a result on nonnegative projection matrices; recall that a matrix P is a nonnegative projection matrix if $P^2 = P$ and P maps nonnegative vectors to nonnegative vectors. If, in addition, $P = P^T$, then it is called a nonnegative, *orthogonal* projection matrix. The following characterization of nonnegative projection matrices is taken from [10, Theorem 2.38], but originally due to Belitskii and Lyubich (cf. [3, p. 108]).

Proposition 3.3 (Theorem 2.1.11 in [3])**.** *The general form of a nonnegative projection matrix is*

$$P = (A + B)C^T \tag{3.1}$$

where $r = rank(P)$, $A, B, C \in \mathbb{R}^{n\times r}_+$, $A^TA = I$, $C^TA = I$, $B^TA = 0$ and $B^TC = 0$.

As a consequence, a nonnegative, orthogonal projection matrix has the following structure.

Corollary 3.4. *Any $n \times n$ symmetric nonnegative orthogonal projection matrix P with r-dimensional range takes the form $P = CC^T$ for some $C \in \mathbb{R}^{n\times r}_+$ such that*

$C^TC = I$. In particular, the columns of C form a nonnegative, orthonormal basis of the range of P, and these basis vectors therefore have disjoint supports.

Proof. With reference to (3.1), one has

$$P = P^T \Longrightarrow PA = P^TA \Longleftrightarrow (A+B)C^TA = C(A^T+B^T)A \Longleftrightarrow A+B = C.$$

Thus by (3.1) one has $P = CC^T$, and $C^TC = I$. Since nonnegative vectors can only be orthogonal if they have disjoint supports, the columns of C have this property.

Finally, recall that a projection matrix is symmetric if and only if it corresponds to an orthogonal projection. □

One may easily extend this to orthogonal projection operators, as follows.

Proposition 3.5. *Assume that a given orthogonal projection P_S with range $S \subseteq \mathbb{S}^n$ satisfies $P_S(\mathcal{N}^n) \subseteq \mathcal{N}^n$. Then:*

(1) S has a basis of nonnegative matrices with disjoint supports.
(2) If, in addition, S contains the all ones matrix J, then it is a partition subspace.
(3) If, in addition to the condition in item 2), $P_S(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$ and S contains the identity matrix, then S is a Jordan configuration.

Proof. Since P_S is self-adjoint, we may write it as a symmetric matrix, say M_{P_S}, with respect to the standard orthonormal basis of $\mathbb{S}^n$. For a $X \in \mathbb{S}^n$, we define the vector $\mathbf{svec}(X) \in \mathbb{R}^{\frac{1}{2}n(n+1)}$ as

$$\mathbf{svec}(X) = \Big(X_{11}, \sqrt{2}X_{21}, \ldots, \sqrt{2}X_{n1}, X_{22}, \sqrt{2}X_{32}, \ldots, \sqrt{2}X_{n2}, \ldots, X_{nn}\Big)^T.$$

Thus $\mathbf{svec}(X)$ gives the coordinates of X in the standard orthonormal basis of $\mathbb{S}^n$. One therefore has

$$\mathbf{svec}(P_S(X)) = M_{P_S} \cdot \mathbf{svec}(X) \qquad \forall X \in \mathbb{S}^n.$$

Choosing $\mathbf{svec}(X)$ as the standard unit vectors in $\mathbb{R}^{\frac{1}{2}n(n+1)}$ makes it clear that $M_{P_S} \in \mathcal{N}^{\frac{1}{2}n(n+1)}$. Thus the first claim now follows from Corollary 3.4, namely that S has a basis of nonnegative matrices with pairwise disjoint supports. If S contains the all-ones matrix J, then it must hold that these basis matrices are $0/1$, proving the second claim.

Finally, to prove the third claim, we recall that S unital and $P_S(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$ implies that S is a Euclidean Jordan algebra, as mentioned in Section 2.2. Since it has a $0/1$ basis, it is in fact a Jordan configuration if we also assume $I \in S$.

□

The last proposition shows that partition subspaces are closely related to nonnegative projections.

The question remains if there exists an orthogonal projection $P_S : \mathbb{S}^n \to \mathbb{S}^n$ with range $S \subseteq \mathbb{S}^n$ that satisfies $P_S(\mathcal{D}^n) \subseteq \mathcal{D}^n$, but <u>not</u> $P_S(\mathcal{N}^n) \subseteq \mathcal{N}^n$.

Proposition 3.6. *Let P_S be an orthogonal projection with range S that satisfies $P_S(\mathcal{D}^n) \subseteq \mathcal{D}^n$. If $I \in S$, then $P_S(\mathcal{N}^n) \subseteq \mathcal{N}^n$.*

Proof. Let $X \in \mathcal{N}^n$. Since $X \in (\mathcal{D}^n)^* = \mathcal{N}^n + \mathbb{S}^n_+$ and $P_S((\mathcal{D}^n)^*) \subseteq (\mathcal{D}^n)^*$, we know that the diagonal entries of $P_S(X)$ are nonnegative. To see the same for the off-diagonal entries, let r be the spectral radius of X. Then $X + rI \in \mathcal{D}^n$, which implies that $P_S(X + rI) \in \mathcal{D}^n$ has nonnegative off-diagonal entries. Since $P_S(I) = I$ we have $P_S(X) = P_S(X + rI) - rI$, thus showing that $P_S(X)$ has nonnegative off-diagonal entries and therefore $P_S(X) \in \mathcal{N}^n$. □

Hence all admissible subspaces that contain J and I are automatically Jordan configurations for conic problems over the doubly nonnegative cone, if they are $\mathbb{S}^n_+$-positive, by the last two propositions. It is still an open problem whether $\mathcal{D}^n$-positive implies $\mathbb{S}^n_+$-positive.

4. Reducing the semidefinite Relaxation of the quadratic assignment Problem

A semidefinite programming relaxation for QAP(A, B) (see (1.2)), due to Zhao, Karisch, Rendl and Wolkowicz [24], is

$$\begin{aligned} \min\ & \langle B \otimes A, Y\rangle \\ \text{s.t.}\ & \langle I_n \otimes E_{jj}, Y\rangle = 1 \text{ for } j \in [n], \\ & \langle E_{jj} \otimes I_n, Y\rangle = 1 \text{ for } j \in [n], \\ & \langle I_n \otimes (J_n - I_n) + (J_n - I_n) \otimes I_n, Y\rangle = 0, \\ & \langle J_{n^2}, Y\rangle = n^2, \\ & Y \in \mathcal{D}^{n^2}, \end{aligned} \tag{4.1}$$

where $A, B \in \mathbb{S}^n$. We refer to [20] for more details of this relaxation.

First, we have to transform this program into the form on the right side of equation (2.1). We get a feasible solution X_0 by forming the outer product of a vectorized permutation-matrix, for example we can set

$$X_0 = \text{vec}(I_n)\text{vec}(I_n)^T.$$

We get the space $\mathcal{L}$, as seen earlier, by

$$\mathcal{L} = \{X \in \mathbb{S}^{n^2} \mid \langle A_i, X\rangle = 0 \quad \forall i \in [m]\},$$

where

$$\{A_i\}_{i\in[m]} = \{J_{n^2}, I_n \otimes (J_n - I_n) + (J_n - I_n) \otimes I_n, I_n \otimes E_{jj} \text{ and } E_{jj} \otimes I_n \ (j \in [n])\}$$

are the data-matrices of the constraints of the SDP relaxation (4.1). Accordingly the orthogonal complement is exactly $\mathcal{L}^\perp = \text{span}\{A_1, \ldots, A_m\}$.

Theorem 4.1. *Consider an admissible subspace, say $S \subset \mathbb{S}^{n^2}$, for the QAP relaxation* (4.1) *with $n > 2$. Assume that $I \in S$ or $P_S(\mathcal{N}^n) \subseteq \mathcal{N}^n$. Then:*

(1) S has a basis of nonnegative matrices with disjoint supports.
(2) If S is a Jordan sub-algebra of $\mathbb{S}^{n^2}$, i.e. closed under taking squares, then S is a Jordan configuration.

(3) If $P_S(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$, and S is unital, then S is a Jordan configuration.

Proof. Let S be an admissible subspace for the QAP relaxation (4.1) with $n > 2$. The first claim of the theorem is an immediate consequence of Propositions 3.6 and 3.5.

To show the second claim, note that, by assumption, S contains $X_{0,\mathcal{L}^\perp}$ and its square, which we will now calculate. In [4] the authors show that:

$$X_{0,\mathcal{L}^\perp} = \frac{1}{n^2-n}(J_{n^2} - I_n \otimes J_n - J_n \otimes I_n) + \frac{1}{n-1} I_{n^2}.$$

Straightforward calculation now yields

$$\begin{aligned} X^2_{0,\mathcal{L}^\perp} &= \frac{n^2-2n+2}{n^2(n-1)^2} J_{n^2} - \frac{1}{n(n-1)^2}(I_n \otimes J_n + J_n \otimes I_n) + \frac{1}{(n-1)^2} I, \\ X^4_{0,\mathcal{L}^\perp} &= \frac{1}{(n-1)^2} X^2_{0,\mathcal{L}^\perp} + \frac{n-2}{n(n-1)^2} J_{n^2}. \end{aligned}$$

Thus S contains the all-ones matrix if $n > 2$, since

$$\frac{n-2}{n(n-1)^2} J_{n^2} = X^4_{0,\mathcal{L}^\perp} - \frac{1}{(n-1)^2} X^2_{0,\mathcal{L}^\perp},$$

and the right-hand-side terms both belong to S. Thus, S must therefore have a 0/1 basis, i.e. it must be a partition subspace, since it has a basis of nonnegative matrices with disjoint supports. To show that it is in fact a Jordan configuration, we only need to show still that it contains the identity matrix. To this end, it suffices to note that all the diagonal entries of $X^2_{0,\mathcal{L}^\perp}$ are the same, and different from the off-diagonal entries. Since S has a 0/1 basis, it must therefore contain the identity.

We now prove the third claim in the theorem. If we assume $P_S(\mathbb{S}^n_+) \subseteq \mathbb{S}^n_+$, and that S unital, then S is closed under taking squares, i.e. it is a Jordan sub-algebra of $\mathbb{S}^n$ [19, Lemma 5.2.2]. Now the result follows from part 2 of this theorem. □

The important practical implication of this theorem is that the optimal admissible Jordan configuration S of the QAP relaxation (4.1) may be computed using Algorithm 3. The resulting reduction is at least as good as the known ones from the literature, as we now show. "At least as good" means here that the other algorithms return partition spaces which refine the partition space returned by Algorithm 3. Thus the better reduction results in an optimization problem in fewer variables.

Corollary 4.2. *This symmetry reduction of the QAP relaxation* (1.2) *via Algorithm 3 is at least as good as both the group symmetry reduction (see [7, 8]) and the reduction to the smallest coherent algebra containing the data matrices of the program (via the Weisfeiler-Leman algorithm [23]).*

Proof. The symmetric part of a coherent configuration is a Jordan configuration, and the partition given by the orbitals of a group leaving the program invariant is a coherent configuration. Both the partition spaces given by the coherent configuration and the partition given by orbitals are admissible since they adhere to stricter conditions allowing for symmetry reduction than required for admissible subspaces. Since

Algorithm 3 returns the coarsest admissible Jordan configuration, both coherent configurations returned by the other algorithms refine it. □

4.1. Results of Reductions of QAPLib Problems

In practice the (partition) Jordan reduction is not much stronger than group symmetry reduction, and reduction to the smallest coherent algebra containing the data matrices. When comparing reductions for data from QAPLib [5], only a single reduction (esc16f), of the ones that were symmetry reduced before, was stronger, the others were exactly the same as reported in [7], where the reduction was done using group symmetry. But we managed to reduce some larger instances for the first time. We also do gain a large speed up in determining the reduction, since we avoid having to determine the automorphism groups of matrices. In Table 1 we give the dimension of the smallest admissible partition subspace for each problem (for which we determined a reduction), the original number of variables of the problem, and the time needed for the reduction. In Table 2 we show the time needed to block-diagonalize (using the algorithm described in [16]) and solve these problems afterwards (if the dimension of the admissible subspace was at most 3000), as well as the resulting bounds. The optimal value of most of these is known exactly, which we give in the last row, taken from `http://anjos.mgi.polymtl.ca/qaplib/inst.html`.

QAP	Dim.	Red. dim.	Jordan red. (s)
chr18b	52650	14742	0.358
esc16a	32896	150	0.162
esc16b	32896	155	0.192
esc16c	32896	405	0.194
esc16d	32896	405	0.171
esc16e	32896	135	0.166
esc16f	32896	3	0.100
esc16g	32896	230	0.207
esc16h	32896	90	0.130
esc16i	32896	280	0.254
esc16j	32896	150	0.214
esc32a	524800	2112	3.826
esc32b	524800	96	3.306
esc32c	524800	366	3.228
esc32d	524800	342	3.097
esc32e	524800	120	2.885
esc32g	524800	180	2.858
esc32h	524800	666	3.051
esc64a	8390656	679	57.581
kra32	524800	28752	3.099
nug12	10440	2952	0.077
nug15	25425	7425	0.152
nug16b	32896	4704	0.147
nug20	80200	21000	0.819
nug21	97461	27783	0.474
nug22	117370	29766	0.757
nug24	166176	41760	1.010
nug25	195625	28675	1.132
nug27	266085	75087	1.489
nug28	307720	78792	1.865
scr12	10440	2952	0.057
scr15	25425	13275	0.147
tai64c	8390656	75	55.839
tho30	405450	112950	2.927
tho40	1280800	333600	9.477
wil50	3126250	813750	27.123

Table 1. Results for numerical symmetry reduction of QAPLib problems using Algorithm 3.

QAP	block-diag. (s)	solve (s)	blocks (size × mult)	optimal value (4.1)	QAP optimum
esc16a	0.884	0.229	6 × 5,3 × 5,1 × 15,	63.285	68
esc16b	0.749	0.285	7 × 5,1 × 15,	289.999	292
esc16c	2.766	0.759	12 × 5,1 × 15,	153.999	160
esc16d	2.714	0.373	12 × 5,1 × 15,	13.000	16
esc16e	0.753	0.159	6 × 5,2 × 5,1 × 15,	26.337	28
esc16f	0.040	0.048	1 × 3,	0.000	0
esc16g	1.109	0.217	9 × 5,1 × 5,	24.740	26
esc16h	0.414	0.125	5 × 5,1 × 15,	976.228	996
esc16i	1.572	0.296	10 × 5,1 × 5,	11.375	14
esc16j	0.690	0.167	7 × 5,1 × 10,	7.794	8
esc32a	482.027	23.958	26 × 6,1 × 6,	103.320	130
esc32b	11.502	0.041	2 × 24,1 × 24,	131.883	168
esc32c	51.146	0.296	10 × 6,1 × 36,	615.178	642
esc32d	56.076	0.213	9 × 6,2 × 12,1 × 36,	190.227	200
esc32e	11.436	0.054	5 × 6,1 × 30,	1.900	2
esc32g	14.941	0.096	7 × 6,1 × 12,	5.833	6
esc32h	114.943	1.135	14 × 6,1 × 36,	424.398	438
esc64a	1 985.917	0.885	13 × 7,2 × 7,1 × 21,	97.750	116
nug12	12.884	80.019	48 × 2,24 × 2,	567.970	578
scr12	12.894	83.330	48 × 2,24 × 2,	31 409.997	31410
tai64c	182.909	0.153	2 × 15,1 × 30,	1 811 366.481	≥1855928

Table 2. Details on solving (4.1) for QAPLib instances via block-diagonalization.

5. Reducing the ϑ'-Function

The ϑ'-function of a Graph $G = (V, E)$, as given in (1.1), is a doubly nonnegative semidefinite program of size $n := |V|$. Here we can say a bit less about admissible subspaces in the general case. As seen in Theorem 2.3, every admissible subspace needs to contain $C_{\mathcal{L}}$ and $X_{0,\mathcal{L}^\perp}$. Here it is straightforward to see that $C_{\mathcal{L}}$ is exactly the adjacency matrix of the complementary graph $\overline{G}$, and $X_{0,\mathcal{L}^\perp} = \frac{1}{n} I_n$. Thus we know at least that S contains the identity. This implies that every admissible subspace for the ϑ'-function has a basis of nonnegative matrices with disjoint supports, by Propositions 3.5 and 3.6.

But an admissible subspace here does not necessarily need to contain the all-one matrix J_n. One obtains an easy example by $G = ([n], \{\{i, j\} \text{ if } i > m \text{ or } j > m\}$ for $n > m > 0$. It is easy to check that an admissible subspace for this problem is of the form

$$\begin{pmatrix} a & b & b & & & & \\ b & a & b & & & & \\ b & b & a & & & & \\ & & & c & & & \\ & & & & c & & \\ & & & & & c & \\ & & & & & & c \end{pmatrix},$$

here shown for $n = 7$ and $m = 3$. While this case is not too interesting, this shows that the Jordan reduction can be better than a group-symmetry reduction, which would result in the five-dimensional subspace given by

$$\begin{pmatrix} a & b & b & d & d & d & d \\ b & a & b & d & d & d & d \\ b & b & a & d & d & d & d \\ d & d & d & c & e & e & e \\ d & d & d & e & c & e & e \\ d & d & d & e & e & c & e \\ d & d & d & e & e & e & c \end{pmatrix}.$$

Do note though that in this case the three additional variables will be eliminated by the constraints of the SDP soon after.

5.1. The ϑ'-Function of Erdős-Rényi Graphs

Let q be an odd prime, and let $V = \mathrm{GF}(q)^3$ be a three dimensional vector space over the finite field of order q. The set of one dimensional subspaces, i.e. the projective plane, of V is denoted by $\mathrm{PG}(2, q)$. There are $q^2 + q + 1$ such subspaces, which form the vertices of the Erdős-Rényi graph $\mathrm{ER}(q)$. Two vertices are adjacent if they are distinct and orthogonal, i.e. for two representing vectors x and y we have $x^T y = 0$. The interested reader is referred to the papers [11, 12], and the references therein, for more details on these graphs.

We are interested in the size of a maximum stable set of these graphs, specifically upper bounds for this value.

In [11] the authors derive the upper bound

$$\frac{\sqrt{q} + \sqrt{q + 4(q+1)\frac{q+\sqrt{q}+1}{q^2+q+1}}}{2\frac{q+\sqrt{q}+1}{q^2+q+1}}, \tag{5.1}$$

which was shown to be at most as good as the ϑ'-function in [12].

The ϑ'-function of $\mathrm{ER}(q)$ is a doubly nonnegative semidefinite program of size $q^2 + q + 1$. Without further reductions one can practically solve this for up to $q = 17$. In [12] the authors reduced the problem size enough to solve it for up to $q = 31$, and in this paper we managed to solve it for up to $q = 97$.

We applied the reduction algorithm, numerically block-diagonalized (more on that next section) and solved the resulting problems for all primes from $q = 3$ to 97, as shown in Tables 3 and 4. Interestingly, the reduced block sizes always are one block of size 3×3, and $\lceil \frac{q}{2} \rceil$ blocks of size 2×2, i.e. the problem nearly reduces to a second order cone problem. By comparison, the problem was reduced to SDPs of matrix size $2q + 11$ in [12].

6. The Julia Package

We provide a package "SDPSymmetryReduction.jl" as part of the Julia registry, available at `https://github.com/DanielBrosch/SDPSymmetryReduction.jl`. We provide functions to both find an optimal admissible partition subspace for a given SDP, as well as to consequently block-diagonalize it.

q	q^2+q+1	Jordan red. (s)	block diag. (s)	blocks (size × mult.)
3	13	0.001	0.001	$3\times 1, 2\times 2$
5	31	0.002	0.006	$3\times 1, 2\times 3$
7	57	0.007	0.011	$3\times 1, 2\times 4$
11	133	0.067	0.033	$3\times 1, 2\times 6$
13	183	0.092	0.051	$3\times 1, 2\times 7$
17	307	0.241	0.170	$3\times 1, 2\times 9$
19	381	0.421	0.303	$3\times 1, 2\times 10$
23	553	1.019	0.723	$3\times 1, 2\times 12$
29	871	2.455	2.392	$3\times 1, 2\times 15$
31	993	3.398	3.664	$3\times 1, 2\times 16$
37	1407	7.745	9.068	$3\times 1, 2\times 19$
41	1723	13.039	16.358	$3\times 1, 2\times 21$
43	1893	14.533	21.508	$3\times 1, 2\times 22$
47	2257	19.910	36.711	$3\times 1, 2\times 24$
53	2863	33.271	72.052	$3\times 1, 2\times 27$
59	3541	51.463	140.119	$3\times 1, 2\times 30$
61	3783	54.714	166.267	$3\times 1, 2\times 31$
67	4557	78.579	332.438	$3\times 1, 2\times 34$
71	5113	115.303	487.162	$3\times 1, 2\times 36$
73	5403	118.058	545.498	$3\times 1, 2\times 37$
79	6321	179.084	886.065	$3\times 1, 2\times 40$
83	6973	215.983	1 336.901	$3\times 1, 2\times 42$
89	8011	293.947	1 931.723	$3\times 1, 2\times 45$
97	9507	434.341	2 912.840	$3\times 1, 2\times 49$

Table 3. Results of the numerical symmetry reduction of the Theta' function of Erdos-Renyi graphs.

q	solve time (s)	$\vartheta'(\mathrm{ER}(q))$	EV bound (5.1)
3	0.002	5.000	5.560
5	0.003	10.067	10.556
7	0.002	15.743	16.727
11	0.003	31.088	32.051
13	0.003	40.509	41.025
17	0.004	60.221	61.291
19	0.004	71.301	72.493
23	0.004	96.240	96.858
29	0.006	136.978	137.910
31	0.007	151.702	152.707
37	0.007	199.269	200.203
41	0.009	233.390	234.312
43	0.009	250.917	252.063
47	0.011	287.772	288.907
53	0.013	346.626	347.388
59	0.015	408.548	409.534
61	0.015	430.219	431.030
67	0.020	496.438	497.775
71	0.019	543.128	544.095
73	0.021	566.915	567.787
79	0.945	639.644	640.932
83	1.111	690.583	691.375
89	0.115	768.469	769.481
97	0.108	877.075	878.027

Table 4. The resulting bounds for the stable set number of Erdos-Renyi graphs.

To enter a semidefinite programm one has to provide (potentially sparse) vectors and matrices $C \in \mathbb{R}^{n^2}, A \in \mathbb{R}^{m\times n^2}$ and $b \in \mathbb{R}^m$ as in (2.1) (vectorizing the variable X). We provide examples for how one can approach this for both the ϑ'-function of a given graph and for the SDP-bound (4.1). Since we return a partition based symmetry reduction, it is not necessary to give entry-wise nonnegativity constraints, as long as one remembers to use nonnegative variables in the final, reduced SDP (see also the example in A).

Determining an admissible Subspace

The function 'admPartSubspace' determines an optimal admissible partition subspace for the problem, by Algorithm 3. This is done using a randomized Jordan-reduction algorithm, and it returns a Jordan algebra. SDPs can be restricted to such a subspace without changing their optimal value.

Given C,A and b, admPartSubspace(C, a, b) returns a Partition P with $P.n$ giving the number of parts of the partition, and $P.P$ returning an integer valued matrix with entries $1, \dots, n$ defining the partition.

For example, let C, A and b define the ϑ'-function of the cycle graph C_5. If we label the vertices such that its adjacency matrix is

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

then calling admPartSubspace(C, a, b) returns the partition P with $P.n = 3$ and

$$P.P = \begin{pmatrix} 1 & 2 & 3 & 3 & 2 \\ 2 & 1 & 2 & 3 & 3 \\ 3 & 2 & 1 & 2 & 3 \\ 3 & 3 & 2 & 1 & 2 \\ 2 & 3 & 3 & 2 & 1 \end{pmatrix},$$

i.e. we can restrict the feasible set to the three-dimensional subspace given by P.

Block-diagonalizing a Jordan-Algebra

The function 'blockDiagonalize' determines a block-diagonalization of a (Jordan)-algebra given by a partition P using a randomized algorithm. It implements the Algorithm from [16] (see also [13]). To our knowledge this is the first implementation available to the public.

Remark 6.1. Every matrix $*$-algebra over $\mathbb{C}$ is isomorphic to a direct sum of full matrix $*$-algebras over $\mathbb{C}$. But the situation over the reals has more cases (as detailed in [15]); this is the underlying property of block-diagonalization over the reals. We currently offer two versions of the block diagonalization function: One to attempt to block diagonalize it fully over the reals (which does fail in the cases mentioned in [15]), and one over the complex numbers, which never fails (at the cost of potentially being non-optimal when working with real valued SDPs).

blockDiagonalize(P) returns a real block-diagonalization *blkd*, if it exists, otherwise 'nothing'.

- *blkd.blkSizes* returns an integer array of the sizes of the blocks.
- *blkd.blks* returns an array of length $P.n$ containing arrays of (real) matrices of sizes *blkd.blkSizes*. I.e. *blkd.blks*$[i]$ is the image of the basis element given by the 0/1-matrix with a one in the positions where $P.P$ is i.

blockDiagonalize$(P; complex = true)$ returns the same, but with complex valued matrices, and should be used if no real block-diagonalization was found. To use the complex matrices in practice, remember that a Hermitian matrix Y is positive semidefinite if and only if

$$\begin{pmatrix} \mathrm{real}(Y) & -\mathrm{imag}(Y) \\ \mathrm{imag}(Y) & \mathrm{real}(Y) \end{pmatrix}$$

is positive semidefinite.

Continuing the example of reducing $\vartheta'(C_5)$, blockDiagonalize(P) here returns a block-diagonalization *blkd* with $blkd.blkSizes = [1, 1, 1]$ (i.e. three blocks of size 1×1), and the image of the basis is

$$\begin{aligned} blkd.blks \approx\ & [[1, 1, 1], \\ & [-1.618, 0.618, 2], \\ & [0.618, -1.618, 2]]. \end{aligned}$$

This means that

$$\begin{pmatrix} a & b & c & c & b \\ b & a & b & c & c \\ c & b & a & b & c \\ c & c & b & a & b \\ b & c & c & b & a \end{pmatrix} \succcurlyeq 0 \Leftrightarrow \begin{cases} a - 1.618b + 0.618c \geq 0, \\ a + 0.618b - 1.618 \geq 0, \\ a + 2b + 2c \geq 0, \end{cases}$$

which allows us to rewrite $\vartheta'(C_5)$ as a linear program in the three variables a, b, c.

7. Concluding Remarks

We have extended the Jordan symmetry reduction method to the doubly nonnegative cone, and showed that for this cone that the restriction to admissible subspaces that are partition-spaces is not a strong requirement in Section 3. We have shown that indeed the optimal admissible subspace semidefinite programming relaxation of the quadratic assignment problem is a Jordan configuration in Theorem 4.1 of Section 4 under some weak requirements, and applied this to the symmetric instances of QAPLib. In Section 5 we have seen that the optimal admissible subspace of the ϑ'-function can always be given by a nonnegative basis with disjoint supports, and applied this reduction to Erdős-Rényi -graph instances. Finally we describe the Julia package "SDPSymmetryReduction", available at `https://github.com/DanielBrosch/SDPSymmetryReduction.jl`, implementing the described algorithms in Section 6. In Appendix A we give a complete code example as to how one can use this package to calculate $\vartheta'(\mathrm{ER}(q))$.

Acknowledgement

We thank Edwin van Dam for pointing out the reference [17].

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 764759 (MINOA).

Appendix A. Example use of the Software Package

In this appendix we give a complete example how to compute the ϑ'-function of $\mathrm{ER}(q)$, where $q = 31$, using block-diagonalization, and solving the reduced SDP with JuMP and Mosek.

```
using SDPSymmetryReduction
using LinearAlgebra, SparseArrays
using JuMP, MosekTools

## Calculating the Theta'-function of Erdos-Renyi graphs
q = 31

# Generating the adjacency matrix of ER(q)
PG2q = vcat([[0, 0, 1]],
            [[0, 1, b] for b = 0:q-1],
            [[1, a, b] for a = 0:q-1 for b = 0:q-1])
Adj = [x' * y % q == 0 for x in PG2q, y in PG2q]
Adj[diagind(Adj)] .= 0

# Theta' SDP
N = length(PG2q) # = q^2+q+1
C = ones(N^2)
A = hcat(vec(Adj), vec(Matrix(I, N, N)))'
b = [0, 1]

# Find the optimal admissible subspace (= Jordan algebra)
P = admPartSubspace(C, A, b, true)

# Block-diagonalize the algebra
blkD = blockDiagonalize(P, true)

# Calculate the coefficients of the new SDP
PMat = hcat([sparse(vec(P.P .== i)) for i = 1:P.n]...)
newA = A * PMat
newB = b
newC = C' * PMat

# Solve with optimizer of choice
m = Model(Mosek.Optimizer)

# Initialize variables corresponding parts of the partition P
# >= 0 because the original SDP-matrices are entry-wise nonnegative
x = @variable(m, x[1:P.n] >= 0)

@constraint(m, newA * x .== newB)
@objective(m, Max, newC * x)

# Setup the block-diagonalized PSD-constraints
psdBlocks = sum(blkD.blks[i] .* x[i] for i = 1:P.n)
```

```
for blk in psdBlocks
    if size(blk, 1) > 1
        @constraint(m, blk in PSDCone())
    else
        @constraint(m, blk .>= 0)
    end
end

optimize!(m)

@show termination_status(m)
@show value(newC * x)
```

References

[1] L. Babel, I.V. Chuvaeva, M. Klin, and D.V. Pasechnik. Algebraic combinatorics in mathematical chemistry II. Program implementation of the Weisfeiler-Leman algorithm. Technische Universität München, Preprint TUM-M9701, Arxiv.org e-print 1002.1921, 1997.

[2] C. Bachoc, D. C. Gijswijt, A. Schrijver, and F. Vallentin. Invariant semidefinite programs. In *Handbook on semidefinite, conic and polynomial optimization*, M.F. Anjos and J.B. Lasserre (eds.), pages 219–269. Springer, 2012.

[3] G.R. Belitskii and Yu.I. Lyubich. *Matrix Norms and their Applications*. Volume 36 of Operator Theory: Advances and Applications, Birkhäuser, 2013.

[4] D. Brosch, and E. de Klerk. Minimum energy configurations on a toric lattice as a quadratic assignment problem. *Discrete Optimization*, `https://doi.org/10.1016/j.disopt.2020.100612`.

[5] R. E. Burkard, S. E. Karisch, and F. Rendl. Qaplib–a quadratic assignment problem library. *Journal of Global optimization*, 10(4):391–403, 1997.

[6] P.J. Cameron. Coherent configurations, association schemes and permutation groups. Groups, combinatorics and geometry, pages 55–72, World Scientific, 2003.

[7] E. de Klerk and R. Sotirov. Exploiting group symmetry in semidefinite programming relaxations of the quadratic assignment problem. *Mathematical Programming*, 122(2):225, 2010.

[8] E. de Klerk and R. Sotirov. Improved semidefinite programming bounds for quadratic assignment problems with suitable symmetry. *Mathematical programming*, 133(1-2):75–91, 2012.

[9] J. Faraut and A. Korányi. *Analysis on symmetric cones*. Clarendon Press, Oxford, 1994.

[10] A. Galántai. *Projectors and projection methods.* Advances in Mathematics, Volume 6, Springer, 2004.

[11] C.D. Godsil and M.W. Newman. Eigenvalue bounds for independent sets. *Journal of Combinatorial Theory, Series B* 98(4):721–734, 2008, `https://doi.org/10.1016/j.jctb.2007.10.007`.

[12] E. de Klerk, M.W. Newman, D.V. Pasechnik, R. Sotirov. On the Lovász ϑ-number of almost regular graphs with application to Erdős-Rényi graphs. *European Journal of Combinatorics* 30(4):879–888,2009. `https://doi.org/10.1016/j.ejc.2008.07.022`

[13] E., C. Dobre, and D. V. Pasechnik. Numerical block diagonalization of matrix *-algebras with application to semidefinite programming. *Mathematical programming* 129(1),2011.

[14] J. Löfberg. Pre- and post-processing sum-of-squares programs in practice. *IEEE Transactions on Automatic Control* 5, 54, 1007–1011 (2009). Software available via `https://yalmip.github.io/example/moresos/`

[15] T. Maehara, K. Murota, A numerical algorithm for block-diagonal decomposition of

matrix *-algebras with general irreducible components. *Japan J. Indust. Appl. Math.* 27, 263–293 (2010). https://doi.org/10.1007/s13160-010-0007-8

[16] Kazuo Murota, Yoshihiro Kanno, Masakazu Kojima, and Sadayoshi Kojima. A numerical algorithm for block-diagonal decomposition of matrix ∗-algebras with application to semidefinite programming. *Japan Journal of Industrial and Applied Mathematics* 27(1):125–160, 2010.

[17] Mikhail Muzychuk, Sven Reichard, and Mikhail Klin. On Jordan schemes. arXiv:1912.04451, 2020

[18] F. N. Permenter and P. A. Parrilo. Dimension reduction for semidefinite programs via Jordan algebras. *Mathematical Programming*, 181, 51–84, 2020.

[19] F.N. Permenter. *Reduction methods in semidefinite and conic optimization.* PhD thesis, Massachusetts Institute of Technology, 2017.

[20] J. Povh and F. Rendl. Copositive and semidefinite relaxations of the quadratic assignment problem. *Discrete Optimization*, 6(3):231–241, 2009.

[21] A. Schrijver. A comparison of the Delsarte and Lovász bounds. *IEEE Transactions on Information Theory*, 25(4):425–429, 1979.

[22] A. Tavakoli, D. Rosset, and M. Renou. Enabling computation of correlation bounds for finite-dimensional quantum systems via symmetrization. *Physical review letters* 122.7 (2019): 070501. Code available at https://denisrosset.github.io/qdimsum/.

[23] B.Yu. Weisfeiler and A.A. Leman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno - Technicksaya Informatsia*, Seria 2, 9:12–16, 1968, (Russian).

[24] Q. Zhao, S. E. Karisch, F. Rendl, and H. Wolkowicz. Semidefinite programming relaxations for the quadratic assignment problem. *Journal of Combinatorial Optimization*, 2(1):71–109, 1998.