



# Verifying feasibility of degenerate semidefinite programs

Vladimir Kolmogorov, Simone Naldi, Jeferson Zapata

## ► To cite this version:

Vladimir Kolmogorov, Simone Naldi, Jeferson Zapata. Verifying feasibility of degenerate semidefinite programs. 2024. hal-04591610

**HAL Id: hal-04591610**

**<https://hal.science/hal-04591610v1>**

Preprint submitted on 29 May 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Verifying feasibility of degenerate semidefinite programs

Vladimir Kolmogorov<sup>1</sup>

`vnk@ist.ac.at`

Simone Naldi<sup>2,3</sup>

`simone.naldi@unilim.fr`

Jeferson Zapata<sup>1</sup>

`jeferson.zapata@ist.ac.at`

## Abstract

This paper deals with the algorithmic aspects of solving feasibility problems of semidefinite programming (SDP), aka linear matrix inequalities (LMI). Since in some SDP instances all feasible solutions have irrational entries, numerical solvers that work with rational numbers can only find an approximate solution. We study the following question: is it possible to certify feasibility of a given SDP using an approximate solution that is sufficiently close to some exact solution? Existing approaches make the assumption that there exist rational feasible solutions (and use techniques such as rounding and lattice reduction algorithms).

We propose an alternative approach that does not need this assumption. More specifically, we show how to construct a system of polynomial equations whose set of real solutions is guaranteed to have an isolated correct solution (assuming that the target exact solution is maximum-rank). This allows, in particular, to use algorithms from real algebraic geometry for solving systems of polynomial equations, yielding a hybrid (or symbolic-numerical) method for SDPs. We experimentally compare it with a pure symbolic method in [22]; the hybrid method was able to certify feasibility of many SDP instances on which [22] failed. We argue that our approach may have other uses, such as refining an approximate solution using methods of numerical algebraic geometry for systems of polynomial equations.

**Keywords.** Symbolic-numeric algorithms; linear matrix inequalities; weak feasibility; facial reduction; spectrahedra.

## 1 Introduction

Semidefinite programming (SDP) is a nontrivial generalization of linear programming (LP) asking to minimize linear functions defined over the space of real symmetric matrices and

---

<sup>1</sup>Institute of Science and Technology Austria (ISTA)

<sup>2</sup>Université de Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France.

<sup>3</sup>Sorbonne Université, CNRS, LIP6, F-75005 Paris, France.

restricted to the PSD cone (the convex cone of positive semidefinite matrices). Despite its simple definition the complexity analysis of SDP and of its feasibility problem (LMI, linear matrix inequalities) is still a fairly open question, and likewise the development of efficient but reliable algorithms, which is at the heart of research in convex optimization.

Beyond its interest as conic optimization problem, the importance of SDP is enhanced by its numerous applications, the most classical of which is perhaps the semidefinite relaxation of MAX-CUT by Goemans and Williamson [13]. More generally, SDP is used as numerical tool for solving hard non-convex optimization problems through the so-called moment-SOS hierarchy [20], consisting in relaxing polynomial optimization to a sequence of SDP problems of increasing size and with good convergence properties [28]. Specific applications include checking stability of systems in control theory [36] and analyzing the convergence rate of numerical algorithms for convex optimization [10]. In both cases one needs to find a certain Lyapunov function, and the search for such function is cast as a small-scale SDP.

Unfortunately, naively relying on numerical SDP solvers may lead to incorrect conclusions; examples for small dynamical systems can be found in [42]. This motivates the need for algorithms that can certify feasibility of a given SDP. Below we discuss two classes of such algorithms: *symbolic* methods that compute an exact solution in a *Rational Univariate Representation* using techniques from real algebraic geometry, and *symbolic-numerical* (or *hybrid*) methods that certify feasibility using a numerical SDP solver as a subroutine.

**Symbolic methods** Because the constraints in SDP are nonlinear in the entries of the matrix, but still algebraic (defined by polynomial inequalities), the solution of a SDP program is in general not rational but defined over some algebraic extension of the base field, which we usually assume to be  $\mathbb{Q}$ . The degree of this extension, known as the algebraic degree of SDP, represents a measure of the inner complexity of the program, for which exact formulas are known [34].

The underlying algebraic structure of SDP has motivated in the last years the development of computer algebra algorithms whose arithmetic complexity is essentially quadratic in some multilinear bound on the mentioned algebraic degree [22, 31]. The strategy of the series of work [21, 22, 24, 32] relies on reducing the problem of finding a matrix in a section of the PSD cone to the one of finding low rank elements in some affine space of matrices. This can be cast as a real-root finding problem, asking to compute at least one point in every real connected component of an algebraic set, modeled using systems of polynomial equations with finitely-many solutions and hence solved using Gröbner-bases-based algorithms [5]. By its purely symbolic nature, these algorithms are not comparable, in terms of target problem size, with numerical methods for SDP that are mainly based on variants of interior-point method (IPM), and their correctness depend on genericity assumptions on input data.

Concerning algebraic approaches to SDP, let us mention the method in [33] based on homogenization and projective geometry for identifying the feasibility type of a LMI and the certificate in [27] based on the theory of sums of squares polynomial for checking SDP infeasibility.

**Hybrid methods** These methods can potentially handle much larger SDP instances. Most of the existing work has focused on the case when the set of feasible solutions contains rational points [17, 26, 30, 40, 41, 42, 9]. Some of these papers [30, 9] employ a lattice

reduction technique, namely the Lenstra-Lenstra-Lovász (LLL) algorithm. The paper [9] discusses an extension to quadratic fields  $\mathbb{Q}[\sqrt{\ell}]$ , but the method seems to assume that  $\ell$  is known.

Let us finally mention the method by numerical algebraic geometry in [19] which does not strictly fall into the two classes above. It is based on SDP duality and applies homotopy continuation for numerically tracking the central path.

**Our contributions** In this paper we develop an alternative hybrid method which is not restricted to the case when the feasible region contains rational points. The method essentially reduces the problem of certifying feasibility of an SDP to that of solving a system of polynomial equations whose set of real solutions has a zero-dimensional component with a desired solution. Our main result can be stated informally as follows (a precise formulation is given later in Theorems 6 and 7).

**Theorem 1** (informal). *There exists an algorithm that, given a system  $\mathcal{A}(X) = b$ ,  $X \succeq 0$  and an approximate solution  $\tilde{X}$ , constructs a system of polynomial equations over symmetric matrices  $X \in \mathbb{R}^{n \times n}$  and auxiliary variables  $Y \in \mathbb{R}^{n \times (n-r)}$  of the following form:*

$$\mathcal{A}(X) = b, \quad X^\top = X \quad (1a)$$

$$\Pi^\top X \Pi \begin{pmatrix} Y \\ I_{n-r} \end{pmatrix} = 0^{n \times (n-r)} \quad (1b)$$

$$X_{ij} = \tilde{X}_{ij} \quad \forall (i,j) \in K \quad (1c)$$

where  $\Pi$  is a permutation matrix,  $r \in [n]$ , and  $K$  is a subset of  $\{(i,j) \in [n] \times [n] : i \leq j\}$ . (Quantities  $\Pi$ ,  $r$  and  $K$  are computed from  $\mathcal{A}$ ,  $b$  and  $\tilde{X}$ ). If  $\tilde{X}$  is sufficiently close to some maximum-rank solution  $X^* \in P \stackrel{\text{def}}{=} \{X \succeq 0 : \mathcal{A}(X) = b\}$  then the set of real solutions of system (1) has a zero-dimensional component  $\{(X,Y)\}$  with  $X \in P$ .

There may be several ways to exploit this result. One possibility could be to refine the approximate solution  $\tilde{X}$  and the associated matrix  $\tilde{Y}$  (which is easily computable from  $\tilde{X}$ ) by applying the Newton method for system (1). For some polynomial systems this method converges very slowly near the optimum or even diverges; in that case one could try to solve (1) using techniques from numerical algebraic geometry such as those implemented in the software Bertini [4]. The approximate solution  $(\tilde{X}, \tilde{Y})$  could allow the method to focus on the desired zero-dimensional component, and use the regime known as the *endgame*.

In this paper we explore a different direction: we solve system (1) using exact methods from real algebraic geometry, similar to [22]. Accordingly, our experimental results focus on a comparison with [22]. Note that the latter paper also reduces the problem to a system of polynomial equations of the form (1a)-(1b), but with the following differences: (1) the algorithm in [22] has no a priori information about a feasible rank and thus needs to exhaustively search over all (exponentially many) kernel profiles until a solution is found; (2) it provides a guarantee that for one of these kernel profiles the set of real solutions contains a (possibly positive-dimensional) connected component whose points  $(X,Y)$  satisfy  $X \in P$ ; (3) it searches for a minimum-rank solution, whereas we search for a maximum-rank solution.

As in [22], we use a critical point method to find one real solution per component. This method works under certain assumptions (such as complete intersection of the corresponding

variety or finiteness of the number of solutions of some critical equations); on the instances that we tested these assumptions were often violated for the method in [22] but not for our hybrid method, and accordingly the hybrid method was able to certify feasibility of many instances on which [22] failed.

**Outline** The paper is organized as follows. Section 2 contains the main theoretical ingredients and notations. In Section 3 we make a link between determinantal varieties and facial reduction for SDP. Section 4 describes a hybrid algorithm, which is applied to a collection of instances of degenerate SDPs of the literature in Section 5.

## 2 Preliminaries

**Notation for matrices** All norms  $\|\cdot\|$  for vectors and matrices used in this paper are 2-norms. The Frobenius norm for matrices is denoted as  $\|\cdot\|_F$ .

For a matrix  $A \in \mathbb{R}^{p \times q}$  we denote by  $\sigma_i(A) \geq 0$  the  $i$ -th singular value of  $A$ ; if  $i > \min\{p, q\}$  then  $\sigma_i(A) = 0$  by definition. We denote by  $\mathbb{S}_k^p$  the set of  $p \times p$  symmetric matrices with coefficients in a field  $k$  and for  $k = \mathbb{R}$  we simply write  $\mathbb{S}^p$ . If  $A$  is symmetric of size  $p$  ( $A \in \mathbb{S}^p$ ) then  $\lambda_i(A)$  denotes the  $i$ -th largest eigenvalue of  $A$ . A matrix  $A \in \mathbb{S}^p$  is positive semidefinite ( $A \succeq 0$ ) if  $\lambda_p(A) \geq 0$  and positive definite ( $A \succ 0$ ) if  $\lambda_p(A) > 0$ . Recall that if  $A \succeq 0$  then  $\lambda_i(A) = \sigma_i(A)$  for all  $i$ .

The following inequalities are well known (see e.g. [14, Section 8]):

$$|\sigma_i(A) - \sigma_i(B)| \leq \|A - B\| \quad \forall A, B \in \mathbb{R}^{p \times q}, \forall i \quad (2a)$$

$$|\lambda_i(A) - \lambda_i(B)| \leq \|A - B\| \quad \forall A, B \in \mathbb{S}^p, \forall i \quad (2b)$$

We will also denote  $\rho(A) = \sigma_r(A)$  where  $r = \text{rank}(A)$ . Clearly, for every non-zero matrix  $A$  we have  $\rho(A) > 0$ .

For a set of rows  $I \subseteq [p]$  let  $A_I \in \mathbb{R}^{|I| \times q}$  be the submatrix of  $A$  formed by the rows in  $I$ . Similarly, for a set of columns  $J \subseteq [q]$  let  $A^J \in \mathbb{R}^{p \times |J|}$  be the submatrix of  $A$  formed by the columns in  $J$ .

**Semidefinite programming** Let us equip the vector space  $\mathbb{S}^n$  of real symmetric matrices of size  $n$  with the Frobenius inner product:

$$\langle M, N \rangle_{\mathbb{S}} := \text{Trace}(MN) = \sum_{i,j} M_{ij}N_{ij}.$$

for  $M = (M_{ij})$ ,  $N = (N_{ij}) \in \mathbb{S}^n$ . The cone of positive semidefinite matrices (PSD cone) is  $\mathbb{S}_+^n = \{M \in \mathbb{S}^n : M \succeq 0\}$  is a convex closed basic semialgebraic set in  $\mathbb{S}^n$ , and its interior is the open convex cone of positive definite matrices:  $\mathbb{S}_{++}^n := \text{int}(\mathbb{S}_+^n) \subseteq \mathbb{S}^n$ . We denote by  $\langle u, v \rangle_{\mathbb{R}} = \sum u_i v_i$  the Euclidean inner product on  $\mathbb{R}^m$ .

A semidefinite program in standard primal form is given by

$$\begin{aligned} p^* &:= \inf_{X \in \mathbb{S}^n} \langle C, X \rangle_{\mathbb{S}} \\ &\text{s.t.} \quad \mathcal{A}(X) = b \\ &\quad X \in \mathbb{S}_+^n \end{aligned} \quad (3)$$

where  $\mathcal{A} : \mathbb{S}^n \rightarrow \mathbb{R}^m$  is a linear map defined by  $\mathcal{A}(X) = (\langle A_1, X \rangle, \dots, \langle A_n, X \rangle)$  for some real symmetric matrices  $A_1, \dots, A_n \in \mathbb{S}^n$ ,  $b \in \mathbb{R}^m$ , and  $C \in \mathbb{S}^n$ . The feasible sets of the program (3) is denoted by

$$P := \{X \in \mathbb{S}^n : \mathcal{A}(X) = b, X \succeq 0\}. \quad (4)$$

Problem (3) with  $P \neq \emptyset$  is called *strongly feasible* if  $P$  contains a matrix  $X$  with  $X \succ 0$ , and *weakly feasible* otherwise. We will be mainly interested in weakly feasible SDPs. (Note that if (3) is strongly feasible then  $P$  contains rational-valued matrices.)

**Algebraic and semialgebraic geometry** We refer to [8] for basics of (computational) algebraic geometry, and we recall here the main definitions needed in this work. A (complex) algebraic variety is a set of the form

$$V = \{x \in \mathbb{C}^n : f_1(x) = 0, \dots, f_s(x) = 0\} \quad (5)$$

for polynomials  $f_1, \dots, f_s \in K[x]$  with coefficients in some subfield  $K \subseteq \mathbb{C}$  (we usually fix  $K = \mathbb{Q}$  or  $K = \mathbb{R}$ ). Of course defining  $I = \langle f_1, \dots, f_s \rangle \subseteq K[x]$  the ideal generated by  $f_1, \dots, f_s$ , remark that every  $f \in I$  vanishes over  $V$ , in other words  $V = V(I) := \{x \in \mathbb{C}^n : f(x) = 0, \forall f \in I\}$ . The vanishing ideal of a set  $W \subseteq \mathbb{C}^n$  is defined as

$$I(W) = \{f \in \mathbb{C}[x] : f(x) = 0, \forall x \in W\}$$

and by Hilbert's Nullstellensatz [8, §4.1], one has the correspondence  $I(V(J)) = \sqrt{J}$ , where  $\sqrt{J} := \{f \in \mathbb{C}[x] : f^m \in J, \exists m \in \mathbb{N}\}$  is the radical ideal of  $J$ . An ideal  $I$  is called radical if  $I = \sqrt{I}$ . The smallest algebraic variety containing a set  $W$  is denoted by  $\overline{W}$  and called its Zariski-closure.

A variety  $V \subseteq \mathbb{C}^n$  is called irreducible if it is not the union of two proper subvarieties, and every variety is finite union of irreducible varieties called its irreducible components. The dimension of an algebraic variety  $V \subseteq \mathbb{C}^n$  is the integer  $d = \dim(V)$  such that the intersection  $V \cap H_1 \cap \dots \cap H_d$  of  $V$  with  $d$  generic hyperplanes  $H_i$  is finite. The degree of an irreducible variety  $V \subseteq \mathbb{C}^n$  is the cardinality of the intersection of  $V$  with  $\dim(V)$  generic hyperplanes, and the degree of any variety is the sum of the degrees of its irreducible components. A variety whose irreducible components have the same dimension is called equidimensional. An ideal  $I$  with  $V(I) \subseteq \mathbb{C}^n$  is in complete intersection if it can be generated by  $\text{codim}(V(I)) = n - \dim(V(I))$  many polynomials, and with abuse of notation  $V$  is said in complete intersection if this is the case for  $I(V)$ .

Let  $V \subseteq \mathbb{C}^n$  be a variety with  $I(V) = \langle f_1, \dots, f_s \rangle$ . For a points  $p \in V$  the tangent space  $T_p V$  at  $p$  to  $V$  is the kernel of the Jacobian matrix  $J(f_1, \dots, f_s) := (\partial f_i / \partial x_j)_{ij}$  evaluated at  $p$ . A point  $p \in V$  is said to be a *regular point* if the dimension of the tangent space  $T_p V$  at  $p$  coincides with the local dimension of  $V$  at  $p$  (the maximum of the dimensions of the irreducible components containing  $p$ ). When  $V$  is equidimensional of dimension  $d$ , the Jacobian  $J(f_1, \dots, f_s)$  has rank  $\leq n - d$  and equality holds exactly at regular points. We denote the set of regular points as  $\text{Reg}(V)$ , and its complement,  $\text{Sing}(V) := V \setminus \text{Reg}(V)$ , is referred to as the set of singular points of  $V$ .

For  $K = \mathbb{R}$  and  $V$  as in (5), the set  $\mathbb{R}V := V \cap \mathbb{R}^n$  is called a real algebraic variety, and it is defined by the vanishing of finitely many real polynomials. Given  $g_1, \dots, g_t \in \mathbb{R}[x]$ , the set

$$S = \{x \in \mathbb{R}^n : g_1(x) \geq 0, \dots, g_t(x) \geq 0\}$$

is called a basic closed semialgebraic set.

**Determinantal and incidence varieties** We refer to [12, 16] for the general theory of determinantal varieties and we recall below the main definitions in the context of LMI.

Let  $A_1, \dots, A_m \in \mathbb{S}^n$ ,  $b \in \mathbb{R}^m$  be the matrices and the vector defining the map  $\mathcal{A}$  in Equation (3). The set

$$\mathcal{D}_r = \{X \in \mathbb{S}_{\mathbb{C}}^n : \text{rank}(X) \leq r, \mathcal{A}(X) = b\}$$

is called the determinantal variety associated with  $\mathcal{A}$ . It is the algebraic variety defined by the vanishing of the  $(r+1) \times (r+1)$  minors of the matrix  $X$  (plus linear constraints  $\mathcal{A}(X) = b$ ). For some  $r \in \{0, 1, \dots, m\}$ , the real variety  $\mathbb{R}\mathcal{D}_r := \mathcal{D}_r \cap \mathbb{S}^n$  intersects the feasible set  $P$ .

From a computational point of view, minors are not easy to handle. Instead, the rank constraint is equivalent to the existence of a matrix of high rank whose columns generate the kernel of the original matrix. The following set

$$\mathcal{V}_r = \{(X, Y) \in \mathbb{S}_{\mathbb{C}}^n \times \mathbb{C}^{n \times (n-r)} : \mathcal{A}(X) = b, XY = 0, \text{rank}(Y) = n - r\}$$

is associated with the variety  $\mathcal{D}_r$ ; indeed,  $\mathcal{D}_r$  is the projection of  $\mathcal{V}_r$  to  $\mathbb{S}_{\mathbb{C}}^n$ . Remark that  $\mathcal{V}_r$  is not an algebraic variety, since the constraint on the rank of  $Y$  is not closed in the Zariski topology, but instead a constructible set (an open subset of a Zariski closed set). We use the following view of  $\mathcal{V}_r$  from the coordinate charts (see [22]):

$$\mathcal{V}_{r,\iota} = \{(X, Y) \in \mathbb{S}_{\mathbb{C}}^n \times \mathbb{C}^{n \times (n-r)} : \mathcal{A}(X) = b, XY = 0, Y_{[n]-\iota} = I_{n-r}\} \quad (6)$$

for  $\iota \subseteq [n] = \{1, \dots, n\}$ , of cardinality  $|\iota| = r$ , and  $Y_{\iota'}$  is the matrix obtained by selecting the rows of  $Y$  in  $\iota' = [n] - \iota$ . We define  $\mathcal{W}_{r,\iota} = \{X \in \mathbb{S}_{\mathbb{C}}^n : \exists Y \in \mathbb{C}^{n \times (n-r)} \text{ s.t. } (X, Y) \in \mathcal{V}_{r,\iota}\}$  to be the projection of  $\mathcal{V}_{r,\iota}$  to  $\mathbb{S}_{\mathbb{C}}^n$ . Note that  $\mathcal{V}_{r,\iota}$  is an algebraic set. Set  $\mathbb{R}\mathcal{W}_{r,\iota}$  may not be algebraic; it is a constructible set. Let  $\mathbb{R}\mathcal{V}_{r,\iota} := (\mathbb{S}^n \times \mathbb{R}^{n \times (n-r)}) \cap \mathcal{V}_{r,\iota}$  and  $\mathbb{R}\mathcal{W}_{r,\iota} := \mathbb{S}^n \cap \mathcal{W}_{r,\iota}$  be the real traces of these two complex varieties. It can be seen that  $\mathbb{R}\mathcal{W}_{r,\iota}$  is the projection of  $\mathbb{R}\mathcal{V}_{r,\iota}$  to  $\mathbb{S}^n$ . (This follows from the following observation: if  $(X, Y) \in \mathcal{V}_{r,\iota}$  with  $X \in \mathbb{S}^n$  then  $(X, \text{Re } Y) \in \mathbb{R}\mathcal{V}_{r,\iota}$ ).

Incidence varieties are used in [22] to solve feasibility problems of generic instances of LMI, and the following Theorem is a refinement of results in [22].

**Theorem 2.** *Let  $X^* \in \mathbb{S}_+^n$  be a minimum-rank solution of (4), and let  $\iota \subseteq [n]$  be a maximal set of linearly independent columns of  $X^*$ , with  $r = |\iota|$ .*

- (a) *Real variety  $\mathbb{R}\mathcal{W}_{r,\iota}$  has a connected component  $C$  with  $X^* \in C \subseteq P$ .*
- (b) *Real variety  $\mathbb{R}\mathcal{V}_{r,\iota}$  has a connected component  $C'$  with  $X^* \in \pi_{\mathbb{S}^n}(C') \subseteq P$  where  $\pi_{\mathbb{S}^n}(C') = \{X : \exists Y \text{ s.t. } (X, Y) \in C'\}$  is the projection of  $C'$  to  $\mathbb{S}^n$ .*

*Proof.* The definition of  $\iota$  implies that there exists  $Y^* \in \mathbb{R}^{n \times (n-r)}$  with  $X^*Y^* = 0$  and  $Y_{[n]-\iota}^* = I_{n-r}$ . Clearly, we have  $(X^*, Y^*) \in \mathbb{R}\mathcal{V}_{r,\iota}$  and hence  $X^* \in \mathcal{W}_{r,\iota}$ . Let  $C$  be the component of  $\mathcal{W}_{r,\iota}$  containing  $X^*$ .

(a) We need to show that  $C \subseteq P$ . If  $r = 0$  then all points  $(X, Y) \in \mathbb{R}\mathcal{V}_{r,\emptyset}$  satisfy  $X = 0$ , hence  $C = \{X^*\} = \{0\}$  and the claim is straightforward. Suppose that  $r > 0$  and the claim



is false, then there exists continuous curve  $\{X^{(t)}\}_{t \in [0,1]} \subseteq C \subseteq \mathcal{W}_{r,t}$  with  $X^{(0)} = X^*$  and  $X^{(1)} = X$ . For each  $t \in [0,1]$  there must exist matrix  $Y^{(t)}$  with  $(X^{(t)}, Y^{(t)}) \in \mathbb{R}\mathcal{V}_{r,t}$ . Denote  $\lambda_i^{(t)} = \lambda_i(X^{(t)})$  for  $t \in [0,1]$ , so that  $\lambda_1^{(t)} \geq \dots \geq \lambda_n^{(t)}$ . These values satisfy the following properties for each  $t \in [0,1]$ :

- (i) At least  $n - r$  values in  $\lambda_1^{(t)}, \dots, \lambda_n^{(t)}$  are zeros, or equivalently  $\text{rank}(X^{(t)}) \leq r$ . This follows from conditions  $X^{(t)}Y^{(t)} = 0$  and  $\text{rank}(Y^{(t)}) = n - r$ , which imply that  $\text{rank}(X^{(t)}) \leq n - \text{rank}(Y^{(t)}) \leq n - (n - r) = r$ .
- (ii) If  $\lambda_n^{(t)} \geq 0$  then  $\lambda_{r+1}^{(t)} = \dots = \lambda_n^{(t)} = 0$  and  $\lambda_r^{(t)} > 0$ . The first claim follows from (i); if  $\lambda_r^{(t)} = 0$  then  $\text{rank}(X^{(t)}) \leq r - 1$  contradicting the minimality of  $r$  (note that  $X^{(t)} \succeq 0$  and hence  $X^{(t)} \in P$ ).
- (iii) For each  $i \in [n]$ ,  $\lambda_i^{(t)}$  is a continuous function of  $t$ . This holds by continuity of the curve and by (2b).

Conditions  $X^{(0)} \in P$  and  $X^{(1)} \notin P$  give  $\lambda_n^{(0)} \geq 0$  and  $\lambda_n^{(1)} < 0$ . Denote  $s = \sup\{t \in [0,1] : \lambda_n^{(t)} \geq 0\}$ , then  $\lambda_n^{(s)} \geq 0$  by continuity and hence  $s < 1$ . By (ii) we have  $\mu \stackrel{\text{def}}{=} \lambda_r^{(s)} > 0$ . By continuity, there exists  $u \in (s, 1]$  such that  $\lambda_r^{(t)} > \mu/2$  for all  $t \in [s, u]$ . Condition (i) thus gives  $\lambda_{r+1}^{(t)} = \dots = \lambda_n^{(t)} = 0$  for all  $t \in [s, u]$ , and thus  $\sup\{t \in [0,1] : \lambda_n^{(t)} \geq 0\} \geq u > s$ , which is a contradiction.

(b) Let  $C'$  be the connected component of  $\mathcal{V}_{r,t}$  containing  $(X^*, Y^*)$ . Clearly, we have  $\pi_{\mathbb{S}^n}(C') \subseteq C$ , and so the claim holds by part (a).  $\square$

### 3 Incidence varieties and facial reduction

We consider the feasibility problem of the primal SDP in (3), that is the system:

$$\mathcal{A}(X) = b, \quad X \succeq 0. \quad (7)$$

Let  $P = \{X \in \mathbb{S}^n : \mathcal{A}(X) = b, X \succeq 0\}$  be the set of feasible solutions of (4), and let  $r$  be the maximum rank of a matrix in  $P$ . From now on, we assume that  $P \neq \emptyset$  and  $r > 0$ . A matrix  $X \in P$  of rank  $r$  will be called a *maximum-rank solution*.

Below we recall some basic facts about facial reduction for SDPs, see e.g. [11]. Let  $\mathcal{F}$  be the minimal face of the SDP cone  $\mathbb{S}_+^n$  containing  $P$ . It can be represented by a full-rank matrix  $U \in \mathbb{R}^{n \times r}$  as follows:

$$\mathcal{F} = \{X \in \mathbb{S}_+^n : \text{range}(X) \subseteq \text{range}(U)\} = \{UZU^\top : Z \in \mathbb{S}_+^r\}. \quad (8)$$

In other words,  $\mathcal{F}$  is linearly isomorphic (through  $UZU^\top \rightarrow Z$ ) to a copy of the cone  $\mathbb{S}_+^r$ , and in particular it has dimension  $\binom{r+1}{2}$ . Any matrix in the relative interior of  $P$  is in the relative interior of  $\mathcal{F}$  and hence has rank  $r$ ; every other matrix in  $\mathcal{F}$  has rank  $\leq r$ .

Let  $\iota \subseteq [n]$  be a maximal set of linearly independent rows of  $U$ , with  $|\iota| = r$ . We assume for notational convenience that  $\iota = [r]$  (this can be achieved by renaming rows and columns of



$X$ ). Since  $U$  is full-rank, we can apply Gaussian elimination to compute the column reduced echelon form of  $U$ , without changing its range. Accordingly, we can assume w.l.o.g. that

$$U = \begin{pmatrix} I_r \\ U_0 \end{pmatrix}, \quad \text{with } U_0 \in \mathbb{R}^{n-r \times r}. \quad (9)$$

Then for the matrix  $Y_U := -U_0^\top$  we have

$$\mathcal{F} = \{X \in \mathbb{S}_+^n : \text{range}(X) \subseteq \text{range}(U)\} = \{X \in \mathbb{S}_+^n : (Y_U^\top \ I_{n-r}) X = 0^{(n-r) \times n}\} \quad (10a)$$

$$= \{X \in \mathbb{S}_+^n : X \begin{pmatrix} Y_U \\ I_{n-r} \end{pmatrix} = 0^{n \times (n-r)}\} \quad (10b)$$

Now consider the following system of equations (in variables  $X \in \mathbb{S}_\mathbb{C}^n$  and  $Y \in \mathbb{C}^{n \times (n-r)}$ ):

$$\mathcal{A}(X) = b \quad (11a)$$

$$X \begin{pmatrix} Y \\ I_{n-r} \end{pmatrix} = 0^{n \times (n-r)} \quad (11b)$$

Let  $\mathcal{V}_{r,\iota} \subseteq \mathbb{S}_\mathbb{C}^n \times \mathbb{C}^{n \times (n-r)}$  be the set of complex solutions of (11). Note that it is essentially the same incidence variety defined in the previous section (for  $\iota = [r]$ ), modulo notation: matrix  $Y$  in eq. (6) has a larger size, but all additional entries are fixed to constants. We define  $\mathcal{W}_{r,\iota}$ ,  $\mathbb{R}\mathcal{V}_{r,\iota}$  and  $\mathbb{R}\mathcal{W}_{r,\iota}$  in the same way as before:

$$\mathcal{W}_{r,\iota} = \pi_{\mathbb{S}_\mathbb{C}^n}(\mathcal{V}_{r,\iota}) = \{X \in \mathbb{S}_\mathbb{C}^n : \exists Y \in \mathbb{C}^{n \times (n-r)} \text{ s.t. } (X, Y) \in \mathcal{V}_{r,\iota}\} \quad (12a)$$

$$= \{X \in \mathbb{S}_\mathbb{C}^n : \exists Y \in \mathbb{C}^{n \times (n-r)} \text{ s.t. } \mathcal{A}(X) = b, X \begin{pmatrix} Y \\ I_{n-r} \end{pmatrix} = 0^{n \times (n-r)}\} \quad (12b)$$

$$\mathbb{R}\mathcal{V}_{r,\iota} = (\mathbb{S}^n \times \mathbb{R}^{n \times (n-r)}) \cap \mathcal{V}_{r,\iota} \quad \mathbb{R}\mathcal{W}_{r,\iota} = \mathbb{S}^n \cap \mathcal{W}_{r,\iota} = \pi_{\mathbb{S}^n}(\mathbb{R}\mathcal{V}_{r,\iota}) \quad (13)$$

By construction,  $P \subseteq \mathcal{F} \cap \{X \in \mathbb{S}^n : \mathcal{A}(X) = b\} \subseteq \mathbb{R}\mathcal{W}_{r,\iota}$ .

**Lemma 3.** *Let  $X^* = UZ^*U^\top \in P$  be a maximum-rank solution, with  $U$  as in (9) and  $Z^* \in \mathbb{S}_{++}^r$  positive definite. Define open neighborhood  $\Omega^*$  of  $X^*$  via  $\Omega^* = \{X \in \mathbb{S}^n : \|X - X^*\| < \rho(X^*)\}$ . The following holds:*

$$(a) \ P \cap \Omega^* = \mathbb{R}\mathcal{W}_{r,\iota} \cap \Omega^*.$$

$$(b) \ \text{For every } X \in \mathbb{R}\mathcal{W}_{r,\iota} \cap \Omega^* \text{ there exists a unique matrix } Y \text{ such that } (X, Y) \in \mathcal{V}_{r,\iota}. \text{ This matrix is } Y = Y_U.$$

*Proof. Part (a).* If  $X \in P$  then  $\mathcal{A}(X) = b$  and  $(X, Y_U) \in \mathcal{V}_{r,\iota}$  (since  $P \subseteq \mathcal{F}$ ), and thus  $X \in \mathbb{R}\mathcal{W}_{r,\iota}$ . This shows that  $P \subseteq \mathbb{R}\mathcal{W}_{r,\iota}$  and hence  $P \cap \Omega^* \subseteq \mathbb{R}\mathcal{W}_{r,\iota} \cap \Omega^*$ .

Now suppose that  $X \in \mathbb{R}\mathcal{W}_{r,\iota} \cap \Omega^*$ . Then there exists  $Y$  such that  $(X, Y) \in \mathcal{V}_{r,\iota}$ , i.e.  $(X, Y)$  satisfies system (11). Eq. (11b) implies that  $X$  has at least  $n - r$  zero eigenvalues. Since  $\text{rank}(X^*) = r$ , we have  $\lambda_r(X^*) = \rho(X^*) > 0$ . By (2b), for all  $X \in \Omega^*$  we have  $|\lambda_r(X) - \lambda_r(X^*)| \leq \|X - X^*\| < \rho(X^*) = \lambda_r(X^*)$  and thus  $\lambda_r(X) > 0$ , i.e.  $X$  has at least  $r$  strictly positive eigenvalues. We conclude that  $X$  has exactly  $n - r$  zero eigenvalues and exactly  $r$  strictly positive eigenvalues, therefore  $X \succeq 0$  and  $\text{rank}(X) = r$ . Combined with

the fact that  $\mathcal{A}(X) = b$ , this implies that  $X \in P$ . We showed that converse inclusion  $\mathbb{R}\mathcal{W}_{r,\iota} \cap \Omega^* \subseteq P \cap \Omega^*$  is also true.

*Part (b).* As shown in (a), we have  $X \in P \subseteq \mathcal{F}$ . Therefore,  $(X, Y_U) \in \mathcal{V}_{r,\iota}$ , i.e. it satisfies (11). Suppose that there exists  $Y \neq Y'$  such that  $(X, Y)$  and  $(X, Y')$  satisfy (11). Denote  $C = \begin{pmatrix} Y \\ I_{n-r} \end{pmatrix}$ , then  $XC = 0^{n \times 2(n-r)}$ . Clearly, we have  $\text{rank}(C) \geq n - r + 1$ : indeed, the first  $n - r$  columns of  $C$  are linearly independent, and there exists a column among the remaining ones which is not in the range of the first  $n - r$  columns, by the assumption  $Y \neq Y'$ . This implies that  $\text{rank}(X) \leq n - \text{rank}(C) \leq n - (n - r + 1) = r - 1$ , which contradicts condition  $\text{rank}(X) = r$  shown in part (a).

□

## 4 Description of the method

Motivated by Lemma 3, we propose the following hybrid method for certifying the feasibility of system (7) (see Algorithm 1).

---

**Algorithm 1:** Hybrid algorithm for system  $\mathcal{A}(X) = b$ ,  $X \succeq 0$ .

---

- 1 Solve system (7) numerically to get solution  $\tilde{X} \in \mathbb{S}^n$  which is close to some maximum-rank solution  $X^*$ .
- 2 Find numerically a maximal set  $\iota \subseteq [n]$  of linearly independent columns of  $\tilde{X}$ , let  $r = |\iota|$ . Assume for notational convenience that  $\iota = [r]$  (this can be achieved by renaming variables). Write  $\tilde{X} = \begin{pmatrix} \tilde{S} & \tilde{R}^\top \\ \tilde{R} & \tilde{W} \end{pmatrix}$  where  $\tilde{S} = \tilde{X}_\iota^\top \in \mathbb{S}^r$ . If  $\tilde{S}$  is singular then terminate with failure, otherwise define  $\tilde{Y} = -\tilde{S}^{-1}\tilde{R}^\top \in \mathbb{R}^{r \times (n-r)}$ . As we show later,  $\tilde{X} \begin{pmatrix} \tilde{Y} \\ I_{n-r} \end{pmatrix} \approx 0^{n \times (n-r)}$ .
- 3 Write system (11) as

$$\mathcal{Q}(Y)X_{\text{hvec}} = q \tag{14}$$

where  $X_{\text{hvec}}$  is a vector of dimension  $k = \frac{n(n+1)}{2}$  obtained by vectorizing the lower triangular part of  $X$ ,  $\mathcal{Q}(Y)$  is a matrix that depends linearly on  $Y$ , and  $q$  is fixed vector.

- 4 Find numerically a maximal set  $J \subseteq [k]$  of linearly independent columns of  $\mathcal{Q}(\tilde{Y})$ .
- 5 Augment system (11) by fixing variables in  $J' := [n] - J$ :

$$\mathcal{A}(X) = b \tag{15a}$$

$$X \begin{pmatrix} Y \\ I_{n-r} \end{pmatrix} = 0^{n \times (n-r)} \tag{15b}$$

$$(X_{\text{hvec}})_j = (\tilde{X}_{\text{hvec}})_j \quad \forall j \in J' \tag{15c}$$

Compute one point per connected component of the real algebraic variety defined by Equation (15) as in [22] (cf. Section 4.2). By [22, Lemma 3.2], the equations in (15b) at positions  $i, j$  with  $i - j > r$  are implied by other equations and thus can be omitted.

---

The next observation justifies Step 2.

**Lemma 4.** *Consider a PSD matrix  $X = \begin{pmatrix} S & R^\top \\ R & W \end{pmatrix} \in \mathbb{S}_+^n$  with  $S \in \mathbb{S}^r$ , and suppose that  $\iota = [r]$  is a maximal set of linearly independent columns (so that the rank of  $X$  is  $r$ ). Then  $S$  is positive definite, and matrix  $Y = -S^{-1}R^\top$  satisfies  $X \begin{pmatrix} Y \\ I_{n-r} \end{pmatrix} = 0^{n \times (n-r)}$ .*

*Proof.* The first claim follows from Lemma 8 given later. Since the last  $n - r$  columns of  $X$  linearly depend on the first  $r$  columns, we must have

$$\begin{pmatrix} R^\top \\ W \end{pmatrix} = \begin{pmatrix} S \\ R \end{pmatrix} Z$$

for some matrix  $Z \in \mathbb{R}^{r \times (n-r)}$ , i.e.  $R^\top = SZ$  and  $W = RZ$ . Hence  $Z = S^{-1}R^\top$  and  $W = RS^{-1}R^\top$ . Using these equations, one can now check that  $SY + R^\top = 0^{r \times (n-r)}$  and  $RY + W = 0^{(n-r) \times (n-r)}$ , which yields the second claim.  $\square$

**Definition 5.** *We say that sets  $\iota \subseteq [n]$  and  $J \subseteq [k]$  are valid for a maximum-rank solution  $X^*$  if they can be outputs of steps 2 and 4, respectively, assuming that step 1 computes the vector  $X^*$ , and the computations in steps 2 and 4 are performed exactly (i.e. a maximal set of linearly independent columns is computed exactly rather than numerically).*

**Theorem 6.** *Each maximum-rank solution  $X^*$  has an open neighborhood  $\tilde{\Omega} \subset \mathbb{S}^n$  with the following property. Suppose that step 1 produces point  $\tilde{X} \in \tilde{\Omega}$ , and lines 2 and 4 output sets  $\iota$  and  $J$ , respectively, which are valid for  $X^*$ . Then the set of real solutions of system (15) has a zero-dimensional component with a solution  $(X, Y)$  where  $X \in \mathbb{S}_+^n$ .*

*Proof.* By assumption, set  $\iota$  is valid for  $X^*$ , i.e.  $\iota$  is a maximal set of linearly independent columns in  $X^*$  (and so  $|\iota| = r$ ). Assume for notational convenience that  $\iota = [r]$ . By Lemma 4,  $\iota$  is also a maximal set of linearly independent rows in  $X^*$ . Choose matrix  $U \in \mathbb{R}^{n \times r}$  characterizing face  $\mathcal{F}$  as in (8). Since  $\text{range}(X^*) \subseteq \text{range}(U)$  and  $\text{rank}(X^*) = r$ , set  $\iota$  is a maximal set of linearly independent rows in  $U$ . Thus, we can choose  $U$  as in (9).

Denote  $\Omega^* = \{X \in \mathbb{S}^n : \|X - X^*\| < \rho(X^*)\}$ . By Lemma 3(a,b), for any  $(X, Y)$  we have

$$\begin{aligned} \begin{aligned} \mathcal{Q}(Y)X_{\text{hvec}} &= q \\ X &\in \Omega^* \end{aligned} &\iff \begin{aligned} \mathcal{Q}(Y_U)X_{\text{hvec}} &= q \\ X &\in \Omega^* \\ Y &= Y_U \end{aligned} \implies X \in \mathbb{S}_+^n \end{aligned} \quad (16)$$

To simplify notation, let us write  $X_{\text{hvec}} =: \begin{pmatrix} X^J \\ X^{J'} \end{pmatrix}$  and  $\mathcal{Q}(Y_U) =: \mathcal{Q} =: \begin{pmatrix} Q^J & Q^{J'} \end{pmatrix}$ . By assumption,  $J$  is a maximal set of linearly independent columns for matrix  $\mathcal{Q}$ . Let  $I$  be a maximal set of linearly independent rows of  $\mathcal{Q}$ , then  $|I| = |J| = \text{rank}(\mathcal{Q})$ . The linear system  $\mathcal{Q}X_{\text{hvec}} = q$  has at least one solution, namely  $X_{\text{hvec}} = X_{\text{hvec}}^*$ . Thus, removing equations corresponding to rows  $i \notin I$  will not affect the set of feasible solutions. This implies that system  $\mathcal{Q}X_{\text{hvec}} = q = \mathcal{Q}X_{\text{hvec}}^*$  is equivalent to the system

$$\begin{pmatrix} Q_I^J & Q_I^{J'} \end{pmatrix} \begin{pmatrix} X^J - (X^*)^J \\ X^{J'} - (X^*)^{J'} \end{pmatrix} = 0,$$

which is in turn equivalent to  $X^J - (X^*)^J = -(\mathcal{Q}_I^J)^{-1} \mathcal{Q}_I^{J'} (X^{J'} - (X^*)^{J'})$  since matrix  $\mathcal{Q}_I^J$  is nonsingular. Let us denote  $\mathcal{R}_I^J = -(\mathcal{Q}_I^J)^{-1} \mathcal{Q}_I^{J'}$ .

By adding constraint  $X^{J'} = \tilde{X}^{J'}$  to (16) we can now conclude the following for any  $(X, Y)$ :

$$\begin{array}{ccc} (X, Y) \text{ satisfies (15)} & & X^J - (X^*)^J = \mathcal{R}_I^J(\tilde{X}^{J'} - (X^*)^{J'}) \\ X \in \Omega^* & \iff & \begin{array}{l} X^{J'} = \tilde{X}^{J'} \\ X \in \Omega^* \\ Y = Y_U \end{array} \implies X \in \mathbb{S}_+^n \end{array} \quad (17)$$

Let us define  $\delta_{\iota, J, I} = \frac{\rho(X^*)}{4 \max\{\|\mathcal{R}_I^J\|, 1\}}$ . We claim that for any  $\tilde{X} \in \mathbb{S}^n$  with  $\|\tilde{X} - X^*\|_F < \delta_{\iota, J, I}$  the system in the middle of (17) has exactly one feasible solution  $(X, Y)$ . Indeed, we only need to verify that  $X$  defined by the equalities in this system satisfies  $X \in \Omega^*$ . This holds since  $\|X^J - (X^*)^J\| \leq \|\mathcal{R}_I^J\| \cdot \|\tilde{X}^{J'} - (X^*)^{J'}\| < \frac{1}{4}\rho(X^*)$  and  $\|X^{J'} - (X^*)^{J'}\| < \frac{1}{4}\rho(X^*)$ , implying  $\|X - X^*\| \leq \|X - X^*\|_F \leq 2\|X^J - (X^*)^J\| + 2\|X^{J'} - (X^*)^{J'}\| < \rho(X^*)$ .

We can now define set  $\tilde{\Omega}$  in Theorem 6 as follows:  $\tilde{\Omega} = \{\tilde{X} \in \mathbb{S}^n : \|\tilde{X} - X^*\|_F < \delta\}$  where  $\delta = \min_{\iota, J, I} \delta_{\iota, J, I}$  and the minimum is taken over (finitely many) valid choices of  $\iota, J, I$ .  $\square$

## 4.1 Numerical algorithms: implementing steps 2 and 4

To implement Algorithm 1, we need a procedure for numerically computing a maximal set of linearly independent columns of a given matrix. This can be done, for example, by a rank-revealing Gaussian elimination algorithm [35, 44]. Such algorithm takes matrix  $A \in \mathbb{R}^{p \times q}$  and tolerance value  $\varepsilon$  as an input, and produces an integer  $r$  and subset  $J \subseteq [q]$  of size  $r$  with the following guarantees:<sup>1</sup>

$$\sigma_r(A) \geq \varepsilon \quad (18a)$$

$$\sigma_{r+1}(A) \leq c_{pq}\varepsilon \quad (18b)$$

$$\sigma_r(A^J) \geq \sigma_r(A)/c_{pq} \quad (18c)$$

where constant  $c_{pq} > 1$  depends polynomially on dimensions  $p, q$ . Below we analyze Algorithm 1 assuming that steps 2 and 4 use the method above with tolerance values  $\varepsilon_1$  and  $\varepsilon_2$ , respectively. The next result shows that if  $\|\tilde{X} - X^*\|$  is sufficiently small then from  $\tilde{X}$  it is possible to compute sets  $\iota, J$  which are valid for  $X^*$ .

**Theorem 7.** (a) Suppose that  $\delta < \varepsilon_1 < (\rho^* - \delta)/c_{nn}$  where  $\delta = \|\tilde{X} - X^*\|$  and  $\rho^* = \rho(X^*)$ . Then set  $\iota$  computed in step 2 is a maximal set of linearly independent columns for  $X^*$ . Furthermore,

$$\lambda_r(\tilde{S}) \geq \phi(\delta) := \frac{((\rho^* - \delta)/c_{nn} - \delta)^2}{n\|X^*\|} - \delta \quad (19)$$

(b) Suppose the precondition in (a) holds and  $\phi(\delta) > \delta$ . Write  $X^* = \begin{pmatrix} S^* & (R^*)^\top \\ R^* & W^* \end{pmatrix}$  where  $S^* = (X^*)_\iota$ , and let  $Y^* = -(S^*)^{-1}(R^*)^\top$ . Let  $\|\mathcal{Q}\|$  be the 2-norm of linear operator  $\mathcal{Q} : \mathbb{R}^{r \times (n-r)} \rightarrow \mathbb{R}^{p \times q}$ , and denote

$$\psi(\delta) = \|\mathcal{Q}\| \cdot \left[ \frac{\|X^*\|}{\phi(\delta) - \delta} + 1 \right] \cdot \frac{\delta}{\phi(\delta)} \quad (20)$$

<sup>1</sup>The algorithm actually produces subsets  $I \subseteq [p]$  and  $J \subseteq [q]$  of size  $r$  such that  $\sigma_r(A_I^J) \geq \sigma_r(A)/c_{pq}$ . (This follows by combining Lemma 3.1 and Theorem 4.1 in [44].) This implies (18c) since  $\sigma_r(A^J) \geq \sigma_r(A_I^J)$  by a well-known property of singular values, see e.g. [14, Corollary 8.6.3].

If  $\psi(\delta) < \varepsilon_2 < (\rho(\mathcal{Q}(Y^*)) - \psi(\delta))/c_{pq}$  then set  $J$  computed in step 4 is a maximal set of linearly independent columns for  $\mathcal{Q}(Y^*)$ .

Note that  $\lim_{\delta \rightarrow 0} \phi(\delta) = \text{const} > 0$  and  $\lim_{\delta \rightarrow 0} \psi(\delta) = 0$ . Thus, the preconditions in (a,b) can be satisfied for sufficiently small  $\delta > 0$  as long as  $\rho(\mathcal{Q}(Y^*)) > 0$ . (The case when  $\rho(\mathcal{Q}(Y^*)) = 0$  is not interesting: we then have  $\mathcal{Q}(Y^*) = 0$  and thus  $\mathcal{A}(X) = 0$  for all  $X$ ). We remark that linear operator  $\mathcal{Q}$ , function  $\psi(\cdot)$  and value  $\rho(\mathcal{Q}(Y^*)) - \psi(\delta)$  depend on  $\iota$ , but there are only finitely many subsets  $\iota \subseteq [n]$  which give maximal linearly independent columns for  $X^*$ , so we could take the minimum value over such  $\iota$ 's when formulating the final condition.

In the remainder of this section we prove Theorem 7. We start with part (a). Let  $r^* = \text{rank}(X^*)$  be the true rank and  $r$  be the rank computed by the numerical procedure in step 2. Condition (18c) gives  $\sigma_r(\tilde{X}^\iota) \geq \sigma_r(\tilde{X})/c_{nn} \geq (\sigma_r(X^*) - \|\tilde{X} - X^*\|)/c_{nn} = (\rho^* - \delta)/c_{nn}$ . Therefore,  $\sigma_r((X^*)^\iota) \geq \sigma_r(\tilde{X}^\iota) - \|X^* - \tilde{X}\| \geq (\rho^* - \delta)/c_{nn} - \delta > 0$ , implying that columns in  $\iota$  are linearly independent in  $X^*$  (and so  $r \leq r^*$ ). If  $r < r^*$  then  $\sigma_{r+1}(\tilde{X}) \leq c_{nn}\varepsilon_1$  by condition (18b) and  $\sigma_{r+1}(\tilde{X}) \geq \sigma_{r^*}(\tilde{X}) \geq \sigma_{r^*}(X^*) - \|\tilde{X} - X^*\| = \rho^* - \delta$  by (2a), which is a contradiction. This shows that  $r = r^*$ .

The last claim (19) will follow from the result below for matrix  $X = \tilde{X}$  and value  $\tau = \sigma_r(\tilde{X}^\iota) \geq (\rho^* - \delta)/c_{nn} > \delta$ .

**Lemma 8.** Consider symmetric matrices  $X^* \in \mathbb{S}_+^n$ ,  $X \in \mathbb{S}^n$  with  $\|X^* - X\| = \delta$ . Let  $\iota \subseteq [n]$  be a subset of size  $r$  with  $\sigma_r(X^\iota) = \tau \geq \delta$ . Then  $\lambda_r(X^\iota) \geq \frac{(\tau - \delta)^2}{n\|X^*\|} - \delta$ .

Note that this lemma also implies the first part of Lemma 4 (by setting  $X = X^*$ ).

*Proof.* We will use a well-known fact that for any matrix  $A \in \mathbb{R}^{p \times q}$  with  $p \geq q$  we have  $\sigma_q(A) = \min\{\|Au\| : u \in \mathbb{R}^q, \|u\| = 1\}$ .

Consider vector  $u \in \mathbb{R}^\iota$  with  $\|u\| = 1$ , then  $\|X^\iota u\| \geq \tau$ . Let  $v \in \mathbb{R}^n$  be the vector with  $v_i = u_i$  for  $i \in \iota$  and  $v_i = 0$  for  $i \in [n] - \iota$ . We have  $\|Xv\| = \|X^\iota u\| \geq \tau$ , and hence

$$\|X^*v\| \geq \|Xv\| - \|(X^* - X)v\| \geq \tau - \|X^* - X\| \cdot 1 = \tau - \delta$$

Let us write  $X^* = \sum_{i=1}^n \lambda_i v_i v_i^\top = \sum_{i=1}^n w_i w_i^\top$  where  $\|X^*\| = \lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$ ,  $v_1, \dots, v_n$  are orthonormal vectors, and  $w_i = \sqrt{\lambda_i} v_i$  (then  $\|w_i\| \leq \sqrt{\lambda_1}$ ). Denote  $\alpha_i = |v^\top w_i|$ , then  $v^\top X^* v = \sum_{i=1}^n \alpha_i^2$ . We can write

$$\begin{aligned} \tau - \delta &\leq \|X^*v\| = \left\| \sum_{i=1}^n w_i (v^\top w_i) \right\| \leq \sum_{i=1}^n \alpha_i \|w_i\| \\ &\leq \sqrt{\lambda_1} \cdot \sum_{i=1}^n \alpha_i \leq \sqrt{\lambda_1} \cdot \sqrt{n \sum_{i=1}^n \alpha_i^2} = \sqrt{\lambda_1 n (v^\top X^* v)} \end{aligned}$$

where the last inequality is tight if and only if  $\alpha_1 = \dots = \alpha_n$ . Therefore,

$$u^\top X^\iota u = v^\top X v = v^\top X^* v + v^\top (X - X^*) v \geq \frac{(\tau - \delta)^2}{n\lambda_1} - \|v\|^2 \cdot \|X - X^*\| = \frac{(\tau - \delta)^2}{n\lambda_1} - \delta$$

□

Next, we proof part (b) of Theorem 7. We will need the following result.

**Theorem 9** ([1, Theorem 7.12]). *Let  $A, B$  be square matrices of the same size such that  $A$  is nonsingular and  $\|A - B\| \leq \frac{1}{\|A^{-1}\|}$ . Then  $B$  is also nonsingular and*

$$\|B^{-1}\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\| \cdot \|A - B\|}$$

$$\|A^{-1} - B^{-1}\| \leq \frac{\|A^{-1}\|^2 \cdot \|A - B\|}{1 - \|A^{-1}\| \cdot \|A - B\|}$$

For brevity, let us denote  $S = S^*$  and  $R = R^*$ . We can write

$$\tilde{Y} - Y = S^{-1}R^\top - \tilde{S}^{-1}\tilde{R}^\top = (S^{-1} - \tilde{S}^{-1})R^\top + \tilde{S}^{-1}(R^\top - \tilde{R}^\top)$$

From part (a) we know that  $\|\tilde{S}^{-1}\| \leq \frac{1}{\phi(\delta)}$ , and so  $\|\tilde{S}^{-1} - S^{-1}\| \leq \|\tilde{X} - X^*\| = \delta < \phi(\delta) \leq \frac{1}{\|\tilde{S}^{-1}\|}$ . Using Theorem 9 with  $A = \tilde{S}$ ,  $B = S$  we get

$$\begin{aligned} \|\tilde{Y} - Y^*\| &\leq \|S^{-1} - \tilde{S}^{-1}\| \cdot \|R^\top\| + \|\tilde{S}^{-1}\| \cdot \|R^\top - \tilde{R}^\top\| \\ &\leq \frac{(\frac{1}{\phi(\delta)})^2 \cdot \delta}{1 - \frac{1}{\phi(\delta)} \cdot \delta} \cdot \|X^*\| + \frac{1}{\phi(\delta)} \cdot \delta = \left[ \frac{\|X^*\|}{\phi(\delta) - \delta} + 1 \right] \cdot \frac{\delta}{\phi(\delta)} \end{aligned}$$

Denote  $\mathcal{Q}^* = \mathcal{Q}(X^*)$  and  $\tilde{\mathcal{Q}} = \mathcal{Q}(\tilde{X})$ , then  $\|\tilde{\mathcal{Q}} - \mathcal{Q}^*\| \leq \|\mathcal{Q}\| \cdot \|\tilde{Y} - Y^*\| \leq \psi(\delta)$ . We now conclude that  $J$  is a maximal set of linearly independent columns of  $\mathcal{Q}^*$  by the same argument as in part (a).

## 4.2 Symbolic algorithms: implementing step 5

Step 5 of Algorithm 1 and Theorem 6, similarly to [22], reduce the original problem of validating the feasibility of a LMI to a classical question in real algebraic geometry, the computation of one point per connected component of the real trace  $\mathbb{R}V = V \cap \mathbb{R}^n$  of a complex variety

$$V = \{x \in \mathbb{C}^n : f_1(x) = 0, \dots, f_c(x) = 0\},$$

see e.g. [3, § 12.6]. A standard technique consists in computing the critical points of a well-chosen polynomial map  $\varphi : \mathbb{C}^n \rightarrow \mathbb{C}$  (usually of low degree) restricted to  $V$ , often called *critical point method* (CPM). The expected output of a CPM is a finite set  $T \subset \mathbb{R}^n$  such that  $T$  intersects every connected component of  $\mathbb{R}V$ .

Usual choices for  $\varphi$  are linear functions: when  $\varphi$  is a generic linear function, and since  $\mathbb{R}V$  has finitely many connected components, there are finitely many critical points of its restriction to  $\mathbb{R}V$ . In case  $\mathbb{R}V$  is non-compact,  $\varphi$  might not admit critical points on the (unbounded) connected components  $C \subseteq V$  satisfying  $\varphi(C) = \mathbb{R}$ . In this case, the recursive method in [43] allows to compute such components. However, our situation is special inasmuch as our target component is an isolated real point, so that the mentioned recursion is not needed on Step 5 of Algorithm 1.

The CPM method for polynomials  $f : \mathbb{C}^n \rightarrow \mathbb{C}^c$  and polynomial map  $\varphi : \mathbb{C}^n \rightarrow \mathbb{C}$  can be described as follows. Recall that  $J(f) = (\partial f_i / \partial x_j)_{ij}$  is the Jacobian matrix of  $f$ . Define the extended Jacobian of  $(\varphi, f)$  via

$$J(\varphi, f) := \begin{pmatrix} \frac{\partial \varphi}{\partial x_1} & \cdots & \frac{\partial \varphi}{\partial x_n} \\ & J(f) & \end{pmatrix}.$$

Now consider the following Lagrange system associated with  $\varphi$  and  $V$ :

$$\mathcal{L}_u : \begin{cases} f_1 = \cdots = f_c = 0 \\ z^T J(\varphi, f) = 0 \\ z^T u - 1 = 0 \end{cases} \quad (21)$$

where  $z = (z_0, z_1, \dots, z_c) \in \mathbb{C}^{c+1}$  are new variables and  $u \in \mathbb{C}^{c+1}$  is a fixed vector. Intuitively, constraint  $z^T u = 1$  for a generic  $u$  is essentially equivalent to the constraint  $z \neq 0$ .

Let  $V(\mathcal{L}_u) \subseteq \mathbb{C}^{n+c+1}$  be the set of solutions of system (21), and let  $\Pi_x(x, z) : \mathbb{C}^{n+c+1} \rightarrow \mathbb{C}^n$  be the projection to the first  $n$  variables (with  $\Pi_x(x, z) = x$ ). Under certain conditions (e.g. those described below) the set  $\Pi_x(V(\mathcal{L}_u))$  will be finite. In this case this set can be computed, for example, via the software `msolve` based on Gröbner basis computations, and represented through a so-called *rational univariate representation*. This is given by a sequence

$$(q, q_0, q_1, \dots, q_n) \in \mathbb{Q}[t]^{n+2},$$

with  $q_0, q$  coprime, and such that

$$T = \left\{ \left( \frac{q_1(t)}{q_0(t)}, \dots, \frac{q_n(t)}{q_0(t)} \right) \in \mathbb{R}^n : q(t) = 0 \right\}.$$

In other words, the coordinates of vectors  $x \in \Pi_x(V(\mathcal{L}_u))$  are represented by the evaluation of  $n$  univariate rational functions to the roots of a univariate polynomial  $q$ .

**Properties of CPM** With some abuse of notation, let us denote

$$\text{Reg}(f) = \{x \in V : \text{rank } J(f) = c\} \quad \text{Sing}(f) = \{x \in V : \text{rank } J(f) < c\}$$

Note that  $\text{Reg}(f) = \text{Reg}(V)$  and  $\text{Sing}(f) = \text{Sing}(V)$  (respectively the set of regular and singular points of  $V = V(f)$ ), assuming that the following condition holds.

**Assumption 1.**  *$V$  is equidimensional of codimension  $c$  and  $I(V) = \langle f_1, \dots, f_c \rangle$ . (Then  $V$  is a complete intersection of dimension  $d = n - c$ , as defined in Section 2).*

A critical point of the restriction of  $\varphi$  to  $\text{Reg}(f)$  is a point  $x \in \text{Reg}(f)$  such that the differential of the restriction of  $\varphi$  to  $\text{Reg}(f)$  at  $x$  is not surjective, that is an element of the constructible set

$$\begin{aligned} \text{Crit}(\varphi, f) &= \{x \in \text{Reg}(f) : \text{rank } J(\varphi, f) < c + 1\} \\ &= \{x \in \text{Reg}(f) : \text{rank } J(\varphi, f) = c\} \end{aligned}$$

We also denote  $\text{Crit}(\varphi, V) = \{x \in \text{Reg}(V) : \text{rank } J(\varphi, f) = c\}$  under Assumption 1.



**Proposition 10.** *Define*

$$\begin{aligned}\text{Crit}_u(\varphi, f) &= \{x \in \text{Crit}(\varphi, f) : \exists z \in \mathbb{C}^{c+1} \text{ s.t. } z^T J(\varphi, f) = 0, z^T u = 1\} \\ \text{Sing}_u(\varphi, f) &= \{x \in \text{Sing}(f) : \exists z \in \mathbb{C}^{c+1} \text{ s.t. } z^T J(\varphi, f) = 0, z^T u = 1\}\end{aligned}\tag{22}$$

- (a) *There holds  $\Pi_x(V(\mathcal{L}_u)) = \text{Crit}_u(\varphi, f) \cup \text{Sing}_u(\varphi, f)$ .*
- (b) *If Assumption 1 holds then for a generic linear form  $\varphi \in \mathbb{C}[x]_1$ , the set  $\text{Crit}(\varphi, V)$  is finite.*
- (c) *If  $\text{Crit}(\varphi, f)$  is finite then  $\text{Crit}_u(\varphi, f) = \text{Crit}(\varphi, f)$  for a generic  $u$ .*
- (d) *If  $\text{Sing}(f)$  is finite then  $\text{Sing}_u(\varphi, f) = \text{Sing}(f)$  for a generic  $u$ .*

*Consequently, if Assumption 1 holds and  $\text{Sing}(V)$  is finite then for generic  $\varphi, u$  set  $\Pi_x(V(\mathcal{L}_u))$  is finite and satisfies  $\Pi_x(V(\mathcal{L}_u)) = \text{Crit}(\varphi, V) \cup \text{Sing}(V)$ .*

*Proof.* (a) The following can be checked using simple linear algebra:

- Suppose that  $x \in \text{Reg}(f)$ . Then  $x \in \text{Crit}_u(\varphi, f)$  if and only if there exists  $z \in \mathbb{C}^{c+1}$  s.t.  $z^T J(\varphi, f) = 0$  and  $z^T u = 1$ , or equivalently s.t.  $(x, z)$  satisfies (21).
- Suppose that  $x \in \text{Sing}(f)$ . Then  $x \in \text{Sing}_u(\varphi, f)$  if and only if there exists  $z \in \mathbb{C}^{c+1}$  s.t.  $z^T J(\varphi, f) = 0$  and  $z^T u = 1$ , or equivalently s.t.  $(x, z)$  satisfies (21).

These facts imply that  $\Pi_x(V(\mathcal{L}_u)) \cap \text{Reg}(f) = \text{Crit}_u(\varphi, f)$  and  $\Pi_x(V(\mathcal{L}_u)) \cap \text{Sing}(f) = \text{Sing}_u(\varphi, f)$ , which yields the desired claim.

(b) The claim follows from [2, Lem.7].

(c, d) For  $\mathcal{X} \in \{\text{Crit}(\varphi, f), \text{Sing}(f)\}$  denote by  $\mathcal{X}_u$  the corresponding set in (22). We claim that if  $\mathcal{X}$  is finite then  $\mathcal{X}_u = \mathcal{X}$  for a generic  $u$ . Indeed, assume  $\mathcal{X} = \{x_1, \dots, x_k\}$ . By definition of  $\mathcal{X}$ , for each  $x_i \in \mathcal{X}$  there exists  $z_i \in \mathbb{C}^{c+1} - \{0\}$  with  $z_i^T \cdot J(\varphi, f)|_{x_i} = 0$ . Define  $\mathcal{U}_i = \{u \in \mathbb{C}^{c+1} : z_i^T u \neq 0\}$ , and let  $\mathcal{U} = \mathcal{U}_1 \cap \dots \cap \mathcal{U}_k$ . Clearly,  $\mathcal{U}$  is a non-empty Zariski-open set, and for each  $u \in \mathcal{U}$  we have  $\mathcal{X}_u = \mathcal{X}$ .  $\square$

**Remark 1.** *The precondition of the last statement in Proposition 10 can be shown to hold for input system  $f$  in certain circumstances. For example, this is the case for the method of [22] (described in Section 2) applied to the LMI problem  $\mathcal{A}(X) = b, X \in \mathbb{S}_+^n$  with generic input data  $(\mathcal{A}, b)$ : if  $r_{\min}$  is the minimum rank then for at least one subset  $\iota \subseteq [n]$  of size  $|\iota| = r_{\min}$  system (11) will satisfy Assumption 1, and the variety  $V$  will be smooth (i.e.  $\text{Sing}(V) = \emptyset$ ), by [22, Thm. 2.1]. Accordingly, in this case the CPM method will solve the problem with generic  $\varphi, u$ , i.e. it will find a rational univariate representation of at least one minimum-rank solution.*

*However, for non-generic instances  $(\mathcal{A}, b)$  there are no such guarantees. It may happen, for example, that  $\text{Sing}(f)$  is positive-dimensional, in which case the Zariski closure of  $\Pi_x(V(\mathcal{L}_u))$  is expected to be also positive-dimensional causing the CPM method to fail.*

**Radical ideals** Instead of directly applying CPM to  $f$ , one can first compute the radical ideal of  $V(f)$  and attempt to derive a minimal set of polynomials  $g$  such that  $\langle g \rangle = I(V(f)) = \sqrt{\langle f \rangle}$ . Subsequently, CPM is applied to  $g$ . This approach is referred to as  $\text{CPM}_{\text{rad}}$ .

In our experiments, we utilized `Macaulay2` software [15] to compute the radical and then employed the command `mingens` that attempts to find a smaller set of its generators. Note that it is not guaranteed to produce a minimal generating set (unless the radical ideal is homogeneous). We observed that  $\text{CPM}_{\text{rad}}$  was capable of solving some systems that CPM could not resolve. In the latter case, the ideal  $\langle f \rangle$  was not radical, while the former was able to construct a polynomial system whose ideal was radical and complete intersection.

## 5 Numerical Results

This section compares our method, denoted as “Hybrid”, with the method in [22] that we call “HNS”.<sup>2</sup> Their details are described below.

**Hybrid** For the implementation Algorithm 1 requires a numerical solution  $\tilde{X}$  in the neighborhood  $\tilde{\Omega} = \{\tilde{X} \in \mathbb{S}^n : \|\tilde{X} - X^*\|_F < \delta\}$  of an exact solution  $X^*$  as in theorem 6. Since we are interested in weakly feasible SDPs, we cannot use standard interior-point methods that are designed to work for instances in which both primal and dual problems are strongly feasible. Fortunately, weakly feasible SDPs can be tackled via facial reduction approaches [6, 7, 37, 45, 38, 39, 46, 29]. We chose to use the method in [18] which combines facial reduction with a numerical algebraic geometry algorithm based on the Bertini software [4]. The resulting polynomial system (15) was solved with a critical point method (CPM or  $\text{CPM}_{\text{rad}}$ ) as described in Section 4.2.

**HNS** This method constructs system (11) for every subset  $\iota \subseteq [n]$  of increasing cardinality  $r = |\iota|$ , ranging from 0 to  $n-1$ , and applies the critical point method to each such variety. It stops if it finds a PSD matrix (with entries in a rational univariate representation). By Theorem 2, we can expect to obtain a minimal-rank solution in  $P$  for at least one  $\iota$  of size  $|\iota| = r_{\min}$ . Note that in [22] only CPM was used, which was shown to be sufficient for generic instances  $(\mathcal{A}, b)$ . Since we deal with non-generic LMI instances, we tested both CPM and  $\text{CPM}_{\text{rad}}$ .

The Lagrange system (21) was solved using library `msolve` [5]. We say that this computation *succeeds* if the projection to variables  $X$  is zero-dimensional, and at least one of the solutions (represented in a rational univariate representation) is a PSD matrix. Otherwise the computation *fails*.

**Instances** We applied the methods to various instances of weakly feasible SDPs extracted from the existing literature; their description can be found in <https://git.ista.ac.at/jzapata/hybrid-method>. Note that these instances are quite sparse; to test how the meth-

<sup>2</sup>The method of [22] is implemented in software [23], but some details are different, e.g. [23] relies on `FGb` (deprecated) instead of the more recent software `msolve` for solving 0-dimensional polynomial systems. For a more fair comparison we used `msolve` both for the new method and for the algorithm in [22]. Note that the method in [22] has been improved and adapted to the case when the  $P$  is non-generic in [25], but this variant is not implemented and it can be considered computationally more demanding, inasmuch as it would involve computation of bivariate rational representations of algebraic curves, instead of univariate representations.

ods cope with non-sparse instances, for each SDP linear map  $\mathcal{A}(X) = (\langle A_1, X \rangle, \dots, \langle A_n, X \rangle)$  we created two input instances denoted as *clean* (I) and *rotated* (T). The latter uses the map

$$\mathcal{A}_T(X) = (\langle T^\top A_1 T, X \rangle, \dots, \langle T^\top A_n T, X \rangle) = \mathcal{A}(T X T^\top)$$

for some random matrix  $T \in GL(n, \mathbb{R})$ . This transformation does not affect feasibility of the system, but it allows us to generate more weakly feasible SDPs that are less sparse and may become more challenging for numerical methods.

**Results** The outcomes of our experiments are shown in Table 1. Values  $r_{\min} / r_{\max}$  given in the second column are the ranks of solutions found by HNS / Hybrid (recall that these methods search for minimum- and maximum-rank solutions, respectively). The results are given in the format “ $t / t_{\text{rad}}$ ” where  $t$  is the runtime of CPM and  $t_{\text{rad}}$  is the runtime of CPM<sub>rad</sub> (in seconds). For Hybrid we report additionally the time for computing approximate solution; it is given in square brackets. Note that this time is included in both  $t$  and  $t_{\text{rad}}$ . If the method fails then the corresponding runtime is crossed; for example, entry “~~5.8~~ / 3.1” in the first row for HNS means that CPM failed for all considered polynomial systems (corresponding to different subsets  $\iota \subseteq [n]$ ) while CPM<sub>rad</sub> succeeded for at least one system.

In cases where the method reaches the predefined time limit of 10 minutes, the computation is terminated, and the respective output is labeled with  $\infty$ .

**Discussion of results** Hybrid was able to solve most of our instances both with CPM and CPM<sub>rad</sub>. In contrast, HNS failed more often, especially when CPM was used. Interestingly, in several cases HNS with CPM<sub>rad</sub> solved the clean version of the problem but failed on the rotated version. We conjecture that in these cases Macaulay2 did not find *minimal* generators of the radical ideal.

We can conclude that in many cases polynomial systems constructed by Hybrid appear to be easier to solve compared to systems in HNS, and also Hybrid is usually faster than HNS. (Note that the latter needs to solve many more systems). However, there are several exceptions: HNS with CPM<sub>rad</sub> could solve clean versions of the last four rows in Table 1, whereas Hybrid failed or did not terminate on these instances.

In Appendix A we expand the first example from Table 1 (the clean version of DruWo2017). We show, in particular, that in the failed cases varieties  $\text{Sing}(f)$  for given polynomial systems  $f$  are positive-dimensional.

We also confirm that system (15) after fixing variables in Hybrid for this example indeed has a zero-dimensional real component, as predicted by Theorem 6. We observe that the system also has two other real components, which are positive-dimensional. After applying CPM, the system becomes 0-dimensional, and so Hybrid with CPM succeeds.

## 6 Conclusions and future work

We presented a hybrid method for certifying weakly feasible SDP problems that uses an approximate numerical SDP solver and an exact solver for polynomial systems over reals.

SDP	$n/r_{\min}/r_{\max}$	clean instances (I)		rotated instances (T)	
		HNS	Hybrid	HNS	Hybrid
DruWo2017-2.3.2P	3/1/2	<del>5.8</del> / 3.1	1.1/1.6 [0.5]	<del>4.6</del> / <del>12.7</del>	1.2 / 1.6 [0.6]
Gupta2013-12.3P	3/1/2	<del>5.0</del> / 3.1	1.7/1.9 [1.0]	<del>4.8</del> / <del>12.4</del>	1.7 / 2.0 [1.1]
Hauenstein2.6P	3/1/2	<del>5.8</del> / 3.1	1.5/1.9 [0.8]	<del>4.1</del> / <del>12.8</del>	2.0 / 1.9 [1.3]
Helmberg2000-2.2.1P	3/1/2	2.6 / 2.8	1.7/1.9 [1.2]	2.1 / 3.4	1.8 / 1.8 [1.3]
LauVall2020-2.5.1P	2/1/1	1.6 / 1.8	1.1/1.3 [0.7]	1.2 / 2.0	1.1 / 1.5 [0.7]
LauVall2020-2.5.2P	3/1/2	<del>6.2</del> / 3.1	1.6/1.9 [1.0]	<del>4.1</del> / <del>12.5</del>	1.7 / 2.4 [1.0]
Pataki2017-4P	3/1/2	<del>5.0</del> / 3.0	1.5/1.9 [0.9]	<del>4.5</del> / <del>12.6</del>	1.6 / 2.0 [0.9]
deKlerk2002-2.1P	2/1/1	1.4 / 1.8	1.1/1.6 [0.7]	1.5 / 1.9	1.1 / 1.3 [0.7]
DruWo2017-2.3.2D	3/1/2	2.4 / 2.8	1.7/2.0 [1.2]	2.6 / 6.2	2.0 / 2.1 [1.3]
Gupta2013-12.3D	3/1/2	2.4 / 2.8	1.6/2.0 [1.1]	2.2 / 3.2	1.8 / 2.0 [1.3]
HNS2020-4.1D	4/2/2	8.7 / 9.7	2.4/2.9 [1.7]	7.0 / 17.6	2.9 / 3.1 [2.0]
Hauenstein2.6D	3/1/2	2.5 / 2.8	1.8/1.5 [1.2]	2.0 / 6.2	1.9 / 2.2 [1.3]
Helmberg2000-2.2.1D	3/1/2	<del>5.0</del> / 3.0	1.6/1.8 [1.0]	<del>4.1</del> / 12.3	1.8 / 2.1 [1.1]
Pataki2017-4D	3/1/2	2.4 / 2.9	1.7/1.9 [1.2]	2.0 / 3.4	2.5 / 1.8 [1.6]
Permenter2018-4.3.1D	5/0/1	11.5 / 1.0	5.5/8.1 [4.5]	1.0 / 1.5	6.6 / 6.7 [5.6]
Permenter2018-4.3.2D	4/2/2	1.0 / 1.5	2.3/2.7 [1.6]	9.1 / 129.6	2.6 / 3.1 [1.5]
PatakiCleanDim3P	3/1/2	<del>4.7</del> / 3.6	1.6/1.5 [1.0]	<del>4.6</del> / <del>12.5</del>	1.7 / 1.9 [1.0]
PatakiCleanDim4P	4/1/-	<del>49.5</del> / 4.4	<del>2.3</del> / <del>3.3</del> [1.3]	<del>77.3</del> / <del>133.4</del>	<del>27.2</del> / $\infty$ [2.6]
PatakiCleanDim5P	5/1/-	$\infty$ / 6.9	<del>154.6</del> / <del>519.2</del> [3.0]	$\infty$ / $\infty$	$\infty$ / $\infty$ [9.5]
PatakiCleanDim6P	6/1/-	$\infty$ / 12.4	$\infty$ / $\infty$ [11.5]	$\infty$ / $\infty$	$\infty$ / $\infty$ [68.4]
HeNaSa2016-6.2P	6/2/-	33.1 / 36.3	$\infty$ / $\infty$ [9.3]	$\infty$ / $\infty$	$\infty$ / $\infty$ [10.5]

Table 1: Comparison between Hybrid and HNS for clean and rotated instances.

Our numerical results indicate that the hybrid method can outperform a pure exact algorithm from [22] when given a good approximate solution.

In our current experiments scalability was limited both by the numerical SDP solver that we employed (which was the facial reduction algorithm [18] based on Bertini [4]), and by the exact solver for polynomial systems (`msolve`). We believe that our approach can handle larger instances if the method in [18] is replaced with an alternative facial reduction algorithm, and an exact solver for polynomial systems is replaced with a numerical solver; this is left as a future work. Potential candidates for the latter could be algorithms from numerical algebraic geometry such as Bertini. Using such solver could address the issue discussed in the previous section: if a system of polynomial equations has a zero-dimensional component and a positive-dimensional component, a numerical solver would focus on the desired zero-dimensional component if the previous step found a good approximation.

**Acknowledgments** This work has been partially supported by the ANR JCJC project ANR-21-CE48-0006-01 “HYPERSPACE”.

## A Example: DruWo2017

In this section we illustrate how Hybrid and HNS work on (the clean version of) DruWo2017. (Recall that on this example HNS with CPM fails, and all other methods succeed.) Our conclusions are as follows:

- **Hybrid.** The resulting polynomial system  $F$  has  $\text{Sing}(F) = \{(X^*, Y^*)\}$  where  $(X^*, Y^*)$  is the desired solution, and CPM succeeds. We write down the rational univariate representation of the solution obtained with `msolve`.
- **HNS.** In this case there are two polynomial systems to consider,  $F_1$  and  $F_2$ . Sets  $\text{Sing}(F_1)$  and  $\text{Sing}(F_2)$  are both positive-dimensional, which causes CPM to fail for  $F_1$  and  $F_2$ . Furthermore, the ideals  $\langle F_1 \rangle$  and  $\langle F_2 \rangle$  are not radical. We then compute generators of  $\sqrt{\langle F_1 \rangle}$  and  $\sqrt{\langle F_2 \rangle}$  with `Macaulay2`. These generators turn out to be non-minimal in the first case, and minimal in the second case.  $\text{CPM}_{\text{rad}}$  fails on  $F_1$  but succeeds on  $F_2$ .

The example comes from [11, Example 2.3.2], and can be written as follows (over matrices  $X \in \mathbb{S}^4$ ):

$$X_{33} = 0 \tag{23a}$$

$$2X_{13} + X_{22} = 1 \tag{23b}$$

$$X \succeq 0 \tag{23c}$$

Its solution set  $P$  is

$$P = \left\{ \begin{pmatrix} x & y & 0 \\ y & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} : \begin{pmatrix} x & y \\ y & 1 \end{pmatrix} \succeq 0 \right\}$$

## A.1 Hybrid method

The minimal face  $\mathcal{F}$  containing  $P$  is given by

$$\mathcal{F} = \left\{ \begin{pmatrix} x & y & 0 \\ y & z & 0 \\ 0 & 0 & 0 \end{pmatrix} : \begin{pmatrix} x & y \\ y & z \end{pmatrix} \succeq 0 \right\}$$

This face is described by matrix

$$U = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}$$

as in (8),(9) (with  $\iota = \{1,2\}$ ), which corresponds to matrix  $Y_U = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ . Suppose that in step 1 we found approximate solution  $\tilde{X}$  with entries  $\tilde{X}_{11} = 1$  and  $\tilde{X}_{12} = 0$  (and other entries are close to the unique true values). Also suppose that step 2 correctly finds  $\iota = \{1,2\}$ , which would give  $\tilde{Y} \approx \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ . If  $\tilde{X}_{ij} \neq X_{ij}^*$  for all  $i,j$  then the only way to get a system that contains a 0-dimensional real component with a correct solution is to fix variables  $X_{11} = \tilde{X}_{11}$ ,  $X_{12} = \tilde{X}_{12}$ . Suppose that step 4 correctly identifies these variables, then in step 5 we obtain the system

$$\begin{aligned} X_{33} &= 0 \\ 2X_{13} + X_{22} &= 1 \\ X_{11} &= 1 \\ X_{12} &= 0 \end{aligned} \quad \begin{pmatrix} X_{11} & X_{12} & X_{13} \\ X_{12} & X_{22} & X_{23} \\ X_{13} & X_{23} & X_{33} \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \\ 1 \end{pmatrix} = 0^{3 \times 2} \tag{24}$$

Making substitutions  $X_{11} = 1$ ,  $X_{12} = 0$ ,  $X_{33} = 0$ ,  $X_{13} = \frac{1}{2} - \frac{1}{2}X_{22}$  and eliminating these variables yields the system

$$F : \begin{cases} Y_1 - \frac{X_{22}}{2} + \frac{1}{2} = 0 \\ X_{23} + X_{22}Y_2 = 0 \\ X_{23}Y_2 - Y_1 \left( \frac{X_{22}}{2} - \frac{1}{2} \right) = 0 \end{cases} \quad (25)$$

Its set of solutions is

$$V(F) = \left\{ (X_{22}, X_{23}, Y_1, Y_2) = \left( 2t + 1, -2\sqrt{-\frac{1}{2t+1}}t^2 \pm \sqrt{-\frac{1}{2t+1}}t, t, \sqrt{-\frac{1}{2t+1}}t \right) : t \in \mathbb{C} \setminus \left\{ -\frac{1}{2} \right\} \right\}$$

The solutions are real for  $t \in \{0\} \cup (-\infty, -\frac{1}{2})$ . Accordingly, the real variety  $\mathbb{R}V(F)$  contains a zero-dimensional component with the correct solution  $(1, 0, 0, 0)$  (as predicted by Theorem 6) but also two one-dimensional components corresponding to  $t \in (-\infty, -\frac{1}{2})$  with “plus” and “minus” signs.

To compute  $\text{Sing}(F)$ , we need to evaluate the rank of the Jacobian matrix

$$J(F) = \begin{pmatrix} -\frac{1}{2} & 0 & 1 & 0 \\ Y_2 & 1 & 0 & X_{22} \\ -\frac{Y_1}{2} & Y_2 & \frac{1}{2} - \frac{X_{22}}{2} & X_{23} \end{pmatrix}$$

at points in  $V(F)$ . This rank equals 2 at the point corresponding to  $t = 0$ , and is 3 at all other points. Hence, we have

$$\text{Sing}(F) = \{ (X_{22}, X_{23}, Y_1, Y_2) = (1, 0, 0, 0) \}$$

Next, we formed the Lagrangian system  $\mathcal{L}_u$  as in (21) for a randomly chosen linear map  $\varphi$  and vector  $u \in \mathbb{Z}^4$ , and solved it using `msolve`. It returned the following rational parametrization encoding its zero-dimensional solution set:

$$\Pi_{X_{22}, X_{23}}(V(\mathcal{L}_u)) = \left\{ \left( \frac{q_1(t)}{q'(t)}, \frac{q_2(t)}{q'(t)} \right) : q(t) = 0 \right\}$$

where

$$\begin{aligned} q_1(t) &= -129t^4 - 240t^3 - 76t^2 - 16t - 4, \\ q_2(t) &= t(-135t^4 + 880t^3 + 60t^2 + 16t + 4), \\ q(t) &= 27t^5 - 220t^4 - 20t^3 - 8t^2 - 4t. \end{aligned}$$

The floating-point representation (up to 5 decimal places) is

$$\Pi_{X_{22}, X_{23}}(V(L_u)) \approx \left\{ \begin{aligned} &(-10.18376, 17.84481), \\ &(-0.08682, -0.16012), \\ &(1, 0), \\ &(0.04640 - 0.13399i, 0.08358 + 0.16089i), \\ &(0.04640 + 0.13399i, 0.08358 - 0.16089i) \end{aligned} \right\}$$

## A.2 HNS method

Since HNS searches for a minimum rank solution, we need to use  $|\iota| = 1$ . Consequently, three possibilities arise for the kernel matrix:

$$\begin{pmatrix} Y_{11} & Y_{12} \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ Y_{21} & Y_{22} \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ Y_{31} & Y_{32} \end{pmatrix}$$

It can be checked that for the third matrix  $Y$  system  $XY = 0$  does not have solutions with  $X \in P$ . Below we analyze the two remaining systems.

(1) Making substitutions  $X_{33} = 0$ ,  $X_{13} = \frac{1}{2} - \frac{1}{2}X_{22}$  and eliminating these variables yields

$$\begin{pmatrix} X_{11} & X_{12} & \frac{1}{2} - \frac{X_{22}}{2} \\ X_{12} & X_{22} & X_{23} \\ \frac{1}{2} - \frac{X_{22}}{2} & X_{23} & 0 \end{pmatrix} \begin{pmatrix} Y_{11} & Y_{12} \\ 1 & 0 \\ 0 & 1 \end{pmatrix} = 0^{3 \times 2} \iff F_1 : \begin{cases} X_{12} + X_{11}Y_{11} = 0 \\ X_{22} + X_{12}Y_{11} = 0 \\ X_{11}Y_{12} - \frac{X_{22}}{2} + \frac{1}{2} = 0 \\ X_{23} + X_{12}Y_{12} = 0 \\ -Y_{12}(\frac{X_{22}}{2} - \frac{1}{2}) = 0 \end{cases}$$

Recall that the equation at position (3,1) on the LHS is implied by other equations (see step 5 of Algorithm 1), and accordingly we removed it on the RHS. This polynomial system has the following parametric description for its set of solutions:

$$V(F_1) = \{(X_{11}, X_{12}, X_{22}, X_{23}, Y_{11}, Y_{12}) = (\frac{1}{t^2}, -\frac{1}{t}, 1, 0, t, 0) : t \in \mathbb{C} \setminus \{0\}\}$$

If we compute the Jacobian and replace the parametric description we can see the matrix has rank 4:

$$J(F_1) = \begin{pmatrix} Y_{11} & 1 & 0 & 0 & X_{11} & 0 \\ 0 & Y_{11} & 1 & 0 & X_{12} & 0 \\ Y_{12} & 0 & -\frac{1}{2} & 0 & 0 & X_{11} \\ 0 & Y_{12} & 0 & 1 & 0 & X_{12} \\ 0 & 0 & -\frac{Y_{12}}{2} & 0 & 0 & \frac{1}{2} - \frac{X_{22}}{2} \end{pmatrix} = \begin{pmatrix} t & 1 & 0 & 0 & \frac{1}{t^2} & 0 \\ 0 & t & 1 & 0 & -\frac{1}{t} & 0 \\ 0 & 0 & -\frac{1}{2} & 0 & 0 & \frac{1}{t^2} \\ 0 & 0 & 0 & 1 & 0 & -\frac{1}{t} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Moreover, we can see the ideal  $I = \langle F_1 \rangle$  is not radical as  $(X_{22} - 1)^2 \in I$  but  $X_{22} - 1 \notin I$ . Using Macaulay2 we can compute its radical ideal and obtain

$$\sqrt{I} = \langle X_{23}, Y_{12}, X_{22} - 1, X_{12}^2 - X_{11}, X_{12}Y_{11} + 1, X_{12} + X_{11}Y_{11} \rangle$$

but in this occasion the set of generators is not minimal since

$$X_{12} + X_{11}Y_{11} = X_{12}(X_{12}Y_{11} + 1) - Y_{11}(X_{12}^2 - X_{11})$$

(2) Similarly, for the second system we have



$$\begin{pmatrix} X_{11} & X_{12} & \frac{1}{2} - \frac{X_{22}}{2} \\ X_{12} & X_{22} & X_{23} \\ \frac{1}{2} - \frac{X_{22}}{2} & X_{23} & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ Y_{21} & Y_{22} \\ 0 & 1 \end{pmatrix} = 0^{3 \times 2} \iff F_2 : \begin{cases} X_{11} + X_{12}Y_{21} = 0 \\ X_{12} + X_{22}Y_{21} = 0 \\ X_{23}Y_{21} - \frac{X_{22}}{2} + \frac{1}{2} = 0 \\ X_{23} + X_{22}Y_{22} = 0 \\ X_{23}Y_{22} = 0 \end{cases}$$

Again, the ideal  $I = \langle F_2 \rangle$  is not radical but when we obtain its radical in `Macaulay2` we get a minimal set of generators:

$$\sqrt{I} = \langle X_{23}, Y_{22}, X_{12} + Y_{21}, X_{22} - 1, X_{11} - Y_{21}^2 \rangle$$

To summarize, CPM fails for  $F_1, F_2$  while  $\text{CPM}_{\text{rad}}$  fails for  $F_1$  but succeeds for  $F_2$ .

**Remark 2.** As Table 1 shows, HNS with both CPM and  $\text{CPM}_{\text{rad}}$  fails for the rotated version of *DruWo2017*. The resulting polynomial systems were too big to analyze by hand; we conjecture that in these cases `Macaulay2` also failed to find minimal generators of radical ideals, as for system  $F_1$ .

## References

- [1] K. Atkinson. *An introduction to numerical analysis*. Wiley, 2nd edition, 1991. [13](#)
- [2] B. Bank, M. Giusti, J. Heintz, and L.-M. Pardo. Generalized polar varieties: geometry and algorithms. *Journal of Complexity*, 21(4):377–412, 2005. [15](#)
- [3] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in real algebraic geometry*, volume 10 of *Algorithms and Computation in Mathematics*. Springer-Verlag, second edition, 2006. [13](#)
- [4] D. Bates, J. Hauenstein, A. Sommese, and C. Wampler. *Numerically Solving Polynomial Systems with Bertini*. SIAM, Philadelphia, PA, USA, 2013. [3](#), [16](#), [18](#)
- [5] J. Berthomieu, C. Eder, and M. Safey El Din. MSOLVE: A library for solving polynomial systems. In *Proceedings of the 2021 International Symposium on Symbolic and Algebraic Computation*, pages 51–58, 2021. [2](#), [16](#)
- [6] J. Borwein and H. Wolkowicz. Facial reduction for a cone-convex programming problem. *J. Aust. Math. Soc. (Ser. A)*, 30:369–380, 1981. [16](#)
- [7] J. Borwein and H. Wolkowicz. Regularizing the abstract convex program. *J. Math. Anal. Appl.*, 83:495–530, 1981. [16](#)
- [8] D. Cox, J. Little, and D. O’Shea. *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*. Springer, 2007. [5](#)
- [9] M. Dostert, D. de Laat, and P. Moustrou. Exact semidefinite programming bounds for packing problems. *SIAM J. Optim.*, 31(2):1433–1458, 2021. [2](#), [3](#)

- [10] Y. Drori and M. Teboulle. Performance of first-order methods for smooth convex minimization: A novel approach. *Mathematical Programming*, 145(1):451–482, 2014. [2](#)
- [11] D. Drusvyatskiy and H. Wolkowicz. The many faces of degeneracy in conic optimization. *Foundations and Trends in Optimization*, 3(2):77–170, 2017. [7](#), [19](#)
- [12] D. Eisenbud. Linear sections of determinantal varieties. *American J. Math.*, 110(3):541–575, 1988. [6](#)
- [13] M. Goemans and D. Williamson. Improved approximation algorithms for maximum cuts and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42:1115–1145, 1995. [2](#)
- [14] G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, 2013. [4](#), [11](#)
- [15] D. Grayson and M. Stillman. Macaulay2, a software system for research in algebraic geometry. Available at <http://www.math.uiuc.edu/Macaulay2/>. [16](#)
- [16] J. Harris and L. W. Tu. On symmetric and skew-symmetric determinantal varieties. *Topology*, 23(1):71–84, 1984. [6](#)
- [17] J. Harrison. Verifying nonlinear real formulas via sums of squares. In *Theorem Proving in Higher Order Logics*, pages 102–118. Springer, 2007. [2](#)
- [18] J. D. Hauenstein, A. C. L. Jr., S. McPherson, and Y. Zhang. Numerical algebraic geometry and semidefinite programming. *Results in Applied Mathematics*, 11, 2021. [16](#), [18](#)
- [19] J. D. Hauenstein, A. C. Liddell Jr, S. McPherson, and Y. Zhang. Numerical algebraic geometry and semidefinite programming. *Results in Applied Mathematics*, 11:100166, 2021. [3](#)
- [20] D. Henrion, M. Korda, and J. B. Lasserre. *The Moment-SOS Hierarchy. Lectures In Probability, Statistics, Computational Geometry, Control And Nonlinear Pdes*, volume 4. World Scientific, 2020. [2](#)
- [21] D. Henrion, S. Naldi, and M. Safey El Din. Real root finding for determinants of linear matrices. *J. of Symbolic Computation*, 74:205–238, 2015. [2](#)
- [22] D. Henrion, S. Naldi, and M. Safey El Din. Exact algorithms for linear matrix inequalities. *SIAM J. Optim.*, 26(4):2512–2539, 2016. [1](#), [2](#), [3](#), [4](#), [6](#), [9](#), [13](#), [15](#), [16](#), [18](#)
- [23] D. Henrion, S. Naldi, and M. Safey El Din. SPECTRA: a Maple library for solving linear matrix inequalities in exact arithmetic. *Optimization Methods and Software*, 34(1):62–78, 2019. [16](#)
- [24] D. Henrion, S. Naldi, and M. Safey El Din. Real root finding for low rank linear matrices. *Appl. Algebra Eng. Comm. Comput.*, 31(2):101–133, 2020. [2](#)

- [25] D. Henrion, S. Naldi, and M. Safey El Din. Exact algorithms for semidefinite programs with degenerate feasible set. *J. Symb. Comput.*, 104:942–959, 2021. [16](#)
- [26] E. L. Kaltofen, B. Li, Z. Yang, and L. Zhi. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. *J. Symbolic Comput.*, 47(1):1–15, 2012. [2](#)
- [27] I. Klep and M. Schweighofer. An exact duality theory for semidefinite programming based on sums of squares. *Mathematics of Operations Research*, 38(3):569–590, 2013. [2](#)
- [28] J.-B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11(3):796–817, 2001. [2](#)
- [29] B. Lourenço, M. Muramatsu, and T. Tsuchiya. Solving SDP completely with an interior point oracle. *Optimization Methods and Software*, pages 1–47, 2021. [16](#)
- [30] D. Monniaux and P. Corbineau. On the generation of Positivstellensatz witnesses in degenerate cases. In *International Conference on Interactive Theorem Proving*, pages 249–264. Springer, 2011. [2](#)
- [31] S. Naldi. *Exact algorithms for determinantal varieties and semidefinite programming*. Theses, Université de Toulouse, Sept. 2015. [2](#)
- [32] S. Naldi. Solving rank-constrained semidefinite programs in exact arithmetic. *J. Symb. Comput.*, 85(C):206–223, 2018. [2](#)
- [33] S. Naldi and R. Sinn. Conic programming: infeasibility certificates and projective geometry. *J. Pure Appl. Algebra*, page 106605, 2020. [2](#)
- [34] J. Nie, K. Ranestad, and B. Sturmfels. The algebraic degree of semidefinite programming. *Mathematical Programming*, 122(2):379–405, 2010. [2](#)
- [35] C.-T. Pan. On the existence and computation of rank-revealing LU factorizations. *Linear Algebra Appl.*, 316(1-3):199–222, 2000. [11](#)
- [36] A. Papachristodoulou and S. Prajna. A tutorial on sum of squares techniques for systems analysis. In *American Control Conference*, pages 2686–2700, 2005. [2](#)
- [37] G. Pataki. Strong duality in conic linear programming: facial reduction and extended duals. In *Computational and Analytical Mathematics*, volume 50, pages 613–634. Springer, New York, 2013. [16](#)
- [38] F. Permenter, H. Friberg, and E. Andersen. Solving conic optimization problems via self-dual embedding and facial reduction: a unified approach. *SIAM. J. Optim.*, 27:1257–1282, 2017. [16](#)
- [39] F. Permenter and P. Parrilo. Partial facial reduction: simplified, equivalent SDPs via approximations of the PSD cone. *Mathematical Programming (Ser. A)*, 171:1–54, 2018. [16](#)

- [40] H. Peyrl and P. Parrilo. Computing sum of squares decompositions with rational coefficients. *Theoretical Computer Science*, 409(2):269 – 281, 2008. Symbolic-Numerical Computations. [2](#)
- [41] A. Platzer, J.-D. Quesel, and P. Rümmer. Real world verification. In *CADE*, pages 485–501, 2009. [2](#)
- [42] P. Roux, Y.-L. Voronin, and S. Sankaranarayanan. Validating numerical semidefinite programming solvers for polynomial invariants. *Formal Methods in System Design*, 53(2):286–312, 2018. [2](#)
- [43] M. Safey El Din and E. Schost. Polar varieties and computation of one point in each connected component of a smooth real algebraic set. In *ISSAC’03*, pages 224–231. ACM, 2003. [13](#)
- [44] L. Schork and J. Gondzio. Rank revealing gaussian elimination by the maximum volume concept. *Linear Algebra and its Applications*, 592:1–19, 2020. [11](#)
- [45] H. Waki and M. Muramatsu. Facial reduction algorithms for conic optimization problems. *J. Optim. Theory. Appl.*, 158:188–215, 2013. [16](#)
- [46] Y. Zhu, G. Pataki, and Q. Tran-Dinh. Sieve-sdp: a simple facial reduction algorithm to preprocess semidefinite programs. *Math. Prog. Comput.*, 11(3):503–586, 2019. [16](#)