

Exact duality in semidefinite programming based on elementary reformulations ^{*}

Minghui Liu
minghui@unc.edu

Gábor Pataki
gabor@unc.edu

Department of Statistics and Operations Research
University of North Carolina at Chapel Hill

December 8, 2014

Abstract

In semidefinite programming (SDP), unlike in linear programming, Farkas' lemma may fail to prove infeasibility. Here we obtain an exact, short certificate of infeasibility in SDP by an elementary approach: we reformulate any semidefinite system of the form

$$\begin{aligned} A_i \bullet X &= b_i \quad (i = 1, \dots, m) \\ X &\succeq 0 \end{aligned} \tag{P}$$

using only elementary row operations, and rotations. When (P) is infeasible, the infeasibility of the reformulated system is trivial to verify by inspection.

For feasible systems we obtain a similar reformulation, which trivially has strong duality with its Lagrange dual for all objective functions.

As a corollary, we obtain algorithms to systematically generate the data of all infeasible semidefinite programs (SDPs), and the data of all feasible SDPs whose maximum rank feasible solution has a prescribed rank.

We call our technique *elementary semidefinite (ESD-) reformulation*, or elementary reformulation. We give two methods to construct elementary reformulations. One is direct, and based on a simplified facial reduction algorithm, and the second is obtained by adapting the facial reduction algorithm of Waki and Muramatsu.

Key words: semidefinite programming; duality; infeasibility certificates; strong duality

MSC 2010 subject classification: Primary: 90C46, 49N15; secondary: 52A40

OR/MS subject classification: Primary: convexity; secondary: programming-nonlinear-theory

1 Introduction. The certificate of infeasibility and its proof

Semidefinite programs (SDPs) naturally generalize linear programs and share some of the duality theory of linear programming. However, the value of an SDP may not be attained, it may differ from the value of its

^{*}The paper's previous title was "A short proof of infeasibility and generating all infeasible semidefinite programs"

Lagrange dual, and the simplest version of Farkas' lemma is not always sufficient to prove infeasibility in semidefinite programming.

Several alternatives of the traditional Lagrange dual, and Farkas' lemma of SDP are known, which we will review in detail below: see Borwein and Wolkowicz [5, 4]; Ramana [19]; Ramana, Tunçel, and Wolkowicz [20]; Klep and Schweighofer [10]; Waki and Muramatsu [28], and the second author [16].

We will consider semidefinite systems of the form (P), where the A_i are n by n symmetric matrices, the b_i scalars, $X \succeq 0$ means that X is symmetric, positive semidefinite (psd), and the \bullet dot product of symmetric matrices is the trace of their regular product. To motivate our results on infeasibility, we consider the instance

$$\begin{aligned} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet X &= 0 \\ \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \bullet X &= -1 \\ X &\succeq 0, \end{aligned} \tag{1.1}$$

which is trivially infeasible; to see this, suppose that $X = (x_{ij})_{i,j=1}^3$ is feasible in it. Then $x_{11} = 0$, hence the first row and column of X are zero by psdness, so the second constraint implies $x_{22} = -1$, which is a contradiction.

Thus, the internal structure of the system itself proves its infeasibility.

The goal of this short note is twofold. In Theorem 1 we show that a basic transformation reveals such a simple structure – which proves infeasibility – in *every* infeasible semidefinite system. For feasible systems we give a similar transformation – in Theorem 2 – so the resulting system trivially has strong duality with its Lagrange dual for all objective functions.

Definition 1. *We obtain an elementary semidefinite (ESD-)reformulation, or elementary reformulation of (P) by applying a sequence of the following operations:*

- (1) Replace (A_j, b_j) by $(\sum_{i=1}^m y_i A_i, \sum_{i=1}^m y_i b_i)$, where $y \in \mathbb{R}^m$, $y_j \neq 0$.
- (2) Exchange two equations.
- (3) Replace A_i by $V^T A_i V$ for all i , where V is an invertible matrix.

ESD-reformulations clearly preserve feasibility. Note that operations (1) and (2) are also used in Gaussian elimination: we call them elementary row operations (eros). We call operation (3) a rotation. Clearly, we can assume that a rotation is applied only once, when reformulating (P); then X is feasible for (P) if and only if $V^{-1} X V^{-T}$ is feasible for the reformulation.

Theorem 1. *The system (P) is infeasible, if and only if it has an ESD-reformulation of the form*

$$\begin{aligned} A'_i \bullet X &= 0 \quad (i = 1, \dots, k) \\ A'_{k+1} \bullet X &= -1 \\ A'_i \bullet X &= b'_i \quad (i = k + 2, \dots, m) \\ X &\succeq 0 \end{aligned} \tag{P}_{\text{ref}}$$

where $k \geq 0$, and the A'_i are of the form

$$A'_i = \begin{pmatrix} \overbrace{\quad r_1 + \dots + r_{i-1} \quad} & \overbrace{\quad r_i \quad} & \overbrace{\quad n - r_1 - \dots - r_i \quad} \\ \times & \times & \times \\ \times & I & 0 \\ \times & 0 & 0 \end{pmatrix}$$

for $i = 1, \dots, k+1$, with $r_1, \dots, r_k > 0, r_{k+1} \geq 0$, the \times symbols correspond to blocks with arbitrary elements, and matrices A'_{k+2}, \dots, A'_m and scalars b'_{k+2}, \dots, b'_m are arbitrary.

To motivate the reader we now give an entirely elementary, full proof of the “if” direction. It suffices to prove that (P_{ref}) is infeasible, so assume to the contrary that X is feasible in it. The constraint $A'_1 \bullet X = 0$ and $X \succeq 0$ implies that the upper left r_1 by r_1 block of X is zero, and $X \succeq 0$ proves that the first r_1 rows and columns of X are zero. Inductively, from the first k constraints we deduce that the first $\sum_{i=1}^k r_i$ rows and columns of X are zero.

Deleting the first $\sum_{i=1}^k r_i$ rows and columns from A'_{k+1} we obtain a psd matrix, hence

$$A'_{k+1} \bullet X \geq 0,$$

contradicting the $(k+1)^{\text{st}}$ constraint in (P_{ref}) . (Note that in the $k=0$ case $A'_1 \bullet X = -1$ immediately gives a contradiction.) ■

Note that Theorem 1 allows us to systematically generate *all* infeasible semidefinite systems, as we can choose the data matrices and right hand side with structure required in (P_{ref}) ; then all infeasible systems with fixed n and m are a reformulation of such a system. We comment more on this in Section 3.

We now review relevant literature in detail, and its connection to our results. For surveys and textbooks on SDP, we refer to [25, 2, 24, 6]; for treatments of their duality theory see Bonnans and Shapiro [3]; Renegar [21] and Güler [9].

The fundamental facial reduction algorithm of Borwein and Wolkowicz [5, 4] ensures strong duality in a possibly nonlinear conic system by replacing the underlying cone by a suitable face. Ramana in [19] constructed an extended strong dual for SDPs, with polynomially many extra variables, which leads to an exact Farkas’ lemma. Though these approaches seem at first quite different, Ramana, Tunçel, and Wolkowicz in [20] proved the correctness of Ramana’s dual from the algorithm in [5, 4].

More recently, Klep and Schweighofer in [10] constructed a strong dual and exact Farkas’ lemma for SDPs. Their dual resembles Ramana’s; however, it is based on ideas from algebraic geometry, namely sums of squares representations, not convex analysis.

While the algorithms in [5, 4] assume that the system is feasible, Waki and Muramatsu in [28] presented a simplified facial reduction algorithm for conic linear systems, which dispenses with this assumption, and allows one to prove infeasibility. We state here that our reformulations can be obtained by suitably modifying the facial reduction algorithm in [28]; we describe the connection in detail in Section 3. At the same time we provide a direct, and entirely elementary construction.

The second author in [16] described a simplified facial reduction algorithm, and generalized Ramana’s dual to conic linear systems over *nice* cones (for literature on nice cones see [7], [23], [15]). The recent book [26] also has a nice coverage of facial reduction. We refer to Pólik and Terlaky [17] for a generalization of Ramana’s dual for conic LPs over homogeneous cones. We note that the idea of using elementary reformulations of semidefinite systems first appears in [14]. There the second author uses it to bring a semidefinite system into a form, in which it is easy to check whether it has strong duality with its dual for all objective functions.

Several papers, see for instance Pólik and Terlaky [18] on stopping criteria for conic optimization, point to the need of having access to infeasible problems, and we hope that our results will be useful in this respect. In more recent related work, Alfakih [1] gave a certificate of the maximum rank in a feasible semidefinite system, using a sequence of matrices, somewhat similar to the constructions in the duals of [19, 10], and used it in an SDP based proof of a result of Connelly and Gortler on rigidity [8]. Our Theorem 2 gives such a certificate using ESD-reformulations.

We say that an infeasible SDP is weakly infeasible, if the traditional version of Farkas’ lemma fails to prove its infeasibility. We refer to Waki [27] for a systematic method to generate weakly infeasible SDPs from Lasserre’s relaxation of polynomial optimization problems. Lourenco et al. [11] recently presented an error-bound based reduction procedure to simplify weakly infeasible SDPs.

We organize the rest of the paper as follows. After introducing notation, we describe an algorithm to find the reformulation (P_{ref}), and a constructive proof of the “only if” part of Theorem 1. The algorithm is based on facial reduction; however, it is simplified so we do not need to explicitly refer to faces of the semidefinite cone. The algorithm needs a subroutine to solve a primal-dual pair of SDPs. In the SDP pair the primal will always be strictly feasible, but the dual possibly not, and we need to solve them in exact arithmetic. Hence our algorithm may not run in polynomial time. At the same time it is quite simple, and we believe that it will be useful to verify the infeasibility of small instances. We then illustrate the algorithm with Example 1.

In Section 2 we present our reformulation of feasible systems. Here we modify our algorithm to construct the reformulation (P_{ref}) (and hence detect infeasibility); or to construct a reformulation that is easily seen to have strong duality with its Lagrange dual for all objective functions.

In Section 3 we discuss our results in more detail, and in Section 4 we conclude the paper.

We denote by \mathcal{S}^n , \mathcal{S}_+^n , and \mathcal{S}_{++}^n the set of symmetric, symmetric psd, and symmetric positive definite (pd) matrices of order n , respectively. For a closed, convex cone K we write $x \geq_K y$ to denote $x - y \in K$, and denote the relative interior of K by $\text{ri } K$, and its dual cone by K^* , i.e.,

$$K^* = \{ y \mid \langle x, y \rangle \geq 0 \forall x \in K \}.$$

For some $p < n$ we denote by $0 \oplus \mathcal{S}_+^p$ the set of n by n matrices with the lower right p by p corner psd, and the rest of the components zero. If $K = 0 \oplus \mathcal{S}_+^p$, then $\text{ri } K = 0 \oplus \mathcal{S}_{++}^p$, and

$$K^* = \left\{ \begin{pmatrix} Z_{11} & Z_{12} \\ Z_{12}^T & Z_{22} \end{pmatrix} : Z_{22} \in \mathcal{S}_+^p \right\}.$$

For a matrix $Z \in K^*$ partitioned as above, and $Q \in \mathbb{R}^{p \times p}$ we will use the formula

$$\begin{pmatrix} I_{n-p} & 0 \\ 0 & Q \end{pmatrix}^T Z \begin{pmatrix} I_{n-p} & 0 \\ 0 & Q \end{pmatrix} = \begin{pmatrix} Z_{11} & Z_{12}Q \\ Q^T Z_{12}^T & Q^T Z_{22}Q \end{pmatrix} \quad (1.2)$$

in the reduction step of our algorithm that converts (P) into (P_{ref}): we will choose Q to be full rank, so that $Q^T Z_{22}Q$ is diagonal.

We will rely on the following general conic linear system:

$$\begin{aligned} \mathcal{A}(x) &= b \\ \mathcal{B}(x) &\leq_K d \end{aligned} \quad (1.3)$$

where K is a closed, convex cone, and \mathcal{A} and \mathcal{B} are linear operators, and consider the primal-dual pair of conic LPs

$$\begin{aligned} \sup \quad & \langle c, x \rangle \\ (P_{\text{gen}}) \quad & \text{s.t. } x \text{ is feasible in (1.3)} \end{aligned} \quad \begin{aligned} \inf \quad & \langle b, y \rangle + \langle d, z \rangle \\ \text{s.t. } \quad & \mathcal{A}^*(y) + \mathcal{B}^*(z) = c \quad (D_{\text{gen}}) \\ & z \in K^* \end{aligned}$$

where \mathcal{A}^* and \mathcal{B}^* are the adjoints of \mathcal{A} and \mathcal{B} , respectively.

Definition 2. We say that

- (1) strong duality holds between (P_{gen}) and (D_{gen}), if their optimal values agree, and the latter value is attained, when finite;
- (2) (1.3) is well behaved, if strong duality holds between (P_{gen}) and (D_{gen}) for all c objective functions.
- (3) (1.3) is strictly feasible, if $d - \mathcal{B}(x) \in \text{ri } K$ for some feasible x .

We will use the following

Lemma 1. *If (1.3) is strictly feasible, or K is polyhedral, then (1.3) is well behaved.*

When $K = 0 \oplus \mathcal{S}_+^p$ for some $p \geq 0$, then (P_{gen}) - (D_{gen}) are a primal-dual pair of SDPs. In order to solve them efficiently, one must assume that both are strictly feasible; strict feasibility of the latter means that there is a feasible (y, z) with $z \in \text{ri } K^*$.

The system (P) is trivially infeasible, if the alternative system below is feasible:

$$\begin{aligned} y &\in \mathbb{R}^m \\ \sum_{i=1}^m y_i A_i &\succeq 0 \\ \sum_{i=1}^m y_i b_i &= -1; \end{aligned} \tag{1.4}$$

in this case we say that (P) is strongly infeasible. Note that system (1.4) generalizes Farkas' lemma from linear programming. However, (P) may be infeasible, even without (1.4) being feasible, in which case we say that (P) is weakly infeasible. It is easy to verify, for instance, that the system (1.1) is weakly infeasible.

Proof of "only if" in Theorem 1 The proof relies only on Lemma 1. We start with the system (P), which we assume to be infeasible.

In a general step we have a system

$$\begin{aligned} A'_i \bullet X &= b'_i \quad (i = 1, \dots, m) \\ X &\succeq 0, \end{aligned} \tag{P'}$$

where for some $\ell \geq 0$ and $r_1 > 0, \dots, r_\ell > 0$ the A'_i matrices are of the form required by Theorem 1, and $b'_1 = \dots = b'_\ell = 0$. At the start $\ell = 0$, and in a general step we have $0 \leq \ell < \min\{n, m\}$.

Let us define

$$r := r_1 + \dots + r_\ell, \quad K := 0 \oplus \mathcal{S}_+^{n-r},$$

and note that if $X \succeq 0$ satisfies the first ℓ constraints of (P'), then $X \in 0 \oplus \mathcal{S}_+^{n-r}$ (this follows as in the proof of the "if" direction in Theorem 1).

Consider the homogenized SDP and its dual

$$\begin{array}{ll} \sup & x_0 \\ (P_{\text{hom}}) \quad \text{s.t.} & A'_i \bullet X - b'_i x_0 = 0 \quad \forall i \\ & -X \leq_K 0 \end{array} \qquad \begin{array}{ll} \inf & 0 \\ \text{s.t.} & \sum_i y_i A'_i \in K^* \\ & \sum_i y_i b'_i = -1 \end{array} \quad (D_{\text{hom}})$$

The optimal value of (P_{hom}) is 0, since if (X, x_0) were feasible in it with $x_0 > 0$, then $(1/x_0)X$ would be feasible in (P').

We first check whether (P_{hom}) is strictly feasible, by solving the primal-dual pair of auxiliary SDPs

$$\begin{array}{ll} \sup & t \\ (P_{\text{aux}}) \quad \text{s.t.} & A'_i \bullet X - b'_i x_0 = 0 \quad \forall i \\ & -X + t \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \leq_K 0 \end{array} \qquad \begin{array}{ll} \inf & 0 \\ \text{s.t.} & \sum_i y_i A'_i \in K^* \\ & \sum_i y_i b'_i = 0 \\ & (\sum_i y_i A'_i) \bullet \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} = 1. \end{array} \quad (D_{\text{aux}})$$

Clearly, (P_{aux}) is strictly feasible, with $(X, x_0, t) = (0, 0, -1)$ so it has strong duality with (D_{aux}) . Therefore

$$\begin{aligned} (P_{\text{hom}}) \text{ is not strictly feasible} &\Leftrightarrow \text{the value of } (P_{\text{aux}}) \text{ is } 0 \\ &\Leftrightarrow (P_{\text{aux}}) \text{ is bounded} \\ &\Leftrightarrow (D_{\text{aux}}) \text{ is feasible.} \end{aligned}$$

We distinguish two cases:

Case 1: $n - r \geq 2$ and (P_{hom}) is not strictly feasible.

Let y be a feasible solution of (D_{aux}) and apply the reduction step in Figure 1. Now the lower $(n - r)$ by $(n - r)$ block of $\sum_i y_i A'_i$ is nonzero, hence after Step 3 we have $r_{\ell+1} > 0$. We then set $\ell = \ell + 1$, and continue.

REDUCTION STEP

Step 1: Using eros (1) and (2), replace $(A'_{\ell+1}, b'_{\ell+1})$ by $(\sum_{i=1}^m y_i A'_i, \sum_{i=1}^m y_i b'_i)$, while keeping the first ℓ equations in (P') the same.

Step 2: Find a full rank matrix Q s.t.

$$\begin{pmatrix} I_r & 0 \\ 0 & Q \end{pmatrix}^T A'_{\ell+1} \begin{pmatrix} I_r & 0 \\ 0 & Q \end{pmatrix}$$

is of the form expected from $A'_{\ell+1}$ in Theorem 1.

Step 3: Set $A'_i := \begin{pmatrix} I_r & 0 \\ 0 & Q \end{pmatrix}^T A'_i \begin{pmatrix} I_r & 0 \\ 0 & Q \end{pmatrix}$ ($i = 1, \dots, m$).

Figure 1: The reduction step used to obtain the reformulated system

Case 2: $n - r \leq 1$ or (P_{hom}) is strictly feasible.

Now strong duality holds between (P_{hom}) and (D_{hom}) ; when $n - r \leq 1$, this is true because then K is polyhedral. Hence (D_{hom}) is feasible. Let y be feasible in (D_{hom}) and apply the same reduction step in Figure 1. Then we set $k = \ell$, and stop with the reformulation (P_{ref}) .

We now complete the correctness proof of the algorithm. First, we note that the choice of the rotation matrix in Step 2 of the reduction steps implies that A'_1, \dots, A'_ℓ remain in the required form: cf. equation (1.2).

Second, we prove that after finitely many steps our algorithm ends in Case 2. In each iteration both ℓ and $r = r_1 + \dots + r_\ell$ increase. If $n - r$ becomes less than or equal to 1, then our claim is obviously true. Otherwise, at some point during the algorithm we find $\ell = m - 1$. Then $b'_m \neq 0$, since (P') is infeasible. Hence for any $X \in 0 \oplus \mathcal{S}_{++}^{n-r}$ we can choose x_0 to satisfy the last equality constraint of (P_{hom}) , hence at this point we are in Case 2. ■

We next illustrate our algorithm:

Example 1. Consider the semidefinite system with $m = 6$, and data

$$\begin{aligned}
 A_1 &= \begin{pmatrix} 2 & 0 & 0 & 1 \\ 0 & 3 & 0 & -1 \\ 0 & 0 & 4 & 2 \\ 1 & -1 & 2 & 0 \end{pmatrix}, & A_2 &= \begin{pmatrix} -1 & 2 & 1 & -2 \\ 2 & 3 & 3 & 1 \\ 1 & 3 & 4 & -3 \\ -2 & 1 & -3 & 3 \end{pmatrix}, & A_3 &= \begin{pmatrix} -1 & 1 & -2 & 0 \\ 1 & -2 & 0 & 2 \\ -2 & 0 & -3 & -2 \\ 0 & 2 & -2 & -1 \end{pmatrix}, \\
 A_4 &= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, & A_5 &= \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & -1 & 1 & 0 \end{pmatrix}, & A_6 &= \begin{pmatrix} -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & -1 & 1 \end{pmatrix}, \\
 b &= (0, 6, -3, 2, 1, 3).
 \end{aligned}$$

In the first iteration we are in Case 1, and find

$$\begin{aligned}
 y &= (1, -1, -1, -1, -4, 3), \\
 \sum_i y_i A'_i &= \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\
 \sum_i y_i b_i &= 0.
 \end{aligned}$$

We choose

$$Q = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

to diagonalize $\sum_i y_i A'_i$, and after the reduction step we have a reformulation with data

$$\begin{aligned}
 A'_1 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, & A'_2 &= \begin{pmatrix} -1 & 3 & 1 & -2 \\ 3 & -2 & 2 & 3 \\ 1 & 2 & 4 & -3 \\ -2 & 3 & -3 & 3 \end{pmatrix}, & A'_3 &= \begin{pmatrix} -1 & 2 & -2 & 0 \\ 2 & -5 & 2 & 2 \\ -2 & 2 & -3 & -2 \\ 0 & 2 & -2 & -1 \end{pmatrix}, \\
 A'_4 &= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, & A'_5 &= \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 2 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & -1 & 1 & 0 \end{pmatrix}, & A'_6 &= \begin{pmatrix} -1 & 1 & 0 & -1 \\ 1 & -1 & 1 & 1 \\ 0 & 1 & 0 & -1 \\ -1 & 1 & -1 & 1 \end{pmatrix}, \\
 b' &= (0, 6, -3, 2, 1, 3).
 \end{aligned}$$

We start the next iteration with this data, and $\ell = 1$, $r_1 = r = 1$. We are again in Case 1, and find

$$\begin{aligned}
 y &= (0, 1, 1, 0, 3, -2), \\
 \sum_i y_i A'_i &= \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 2 & 0 \\ -1 & 2 & 4 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\
 \sum_i y_i b'_i &= 0.
 \end{aligned}$$

Now the lower right 3 by 3 block of $\sum_i y_i A'_i$ is psd, and rank 1. We choose

$$Q = \begin{pmatrix} 1 & -2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

to diagonalize this block, and after the reduction step we have a reformulation with data

$$\begin{aligned} A'_1 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, & A'_2 &= \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, & A'_3 &= \begin{pmatrix} -1 & 2 & -6 & 0 \\ 2 & -5 & 12 & 2 \\ -6 & 12 & -31 & -6 \\ 0 & 2 & -6 & -1 \end{pmatrix}, \\ A'_4 &= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & -3 & 0 \\ 1 & -3 & 7 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, & A'_5 &= \begin{pmatrix} 0 & -1 & 2 & 0 \\ -1 & 2 & -4 & -1 \\ 2 & -4 & 9 & 3 \\ 0 & -1 & 3 & 0 \end{pmatrix}, & A'_6 &= \begin{pmatrix} -1 & 1 & -2 & -1 \\ 1 & -1 & 3 & 1 \\ -2 & 3 & -8 & -3 \\ -1 & 1 & -3 & 1 \end{pmatrix}, \\ b' &= (0, 0, -3, 2, 1, 3). \end{aligned}$$

We start the last iteration with $\ell = 2$, $r_1 = r_2 = 1$, $r = 2$. We end up in Case 2, with

$$\begin{aligned} y &= (0, 0, 1, 2, 1, -1), \\ \sum_i y_i A'_i &= \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \\ \sum_i y_i b'_i &= -1. \end{aligned}$$

Now the lower right 2 by 2 submatrix of $\sum_i y_i A'_i$ is zero, so we don't need to rotate. After the reduction step the data of the final reformulation is

$$\begin{aligned} A'_1 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, & A'_2 &= \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, & A'_3 &= \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \\ A'_4 &= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & -3 & 0 \\ 1 & -3 & 7 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, & A'_5 &= \begin{pmatrix} 0 & -1 & 2 & 0 \\ -1 & 2 & -4 & -1 \\ 2 & -4 & 9 & 3 \\ 0 & -1 & 3 & 0 \end{pmatrix}, & A'_6 &= \begin{pmatrix} -1 & 1 & -2 & -1 \\ 1 & -1 & 3 & 1 \\ -2 & 3 & -8 & -3 \\ -1 & 1 & -3 & 1 \end{pmatrix}, \\ b' &= (0, 0, -1, 2, 1, 3). \end{aligned}$$

2 The elementary reformulation of feasible systems

For feasible systems we have the following result:

Theorem 2. *Let $p \geq 0$ be an integer. Then the following hold:*

(1) The system (P) is feasible with a solution with maximum rank $p \geq 0$, if and only if it has a feasible solution with rank p and an ESD-reformulation

$$\begin{aligned} A'_i \bullet X &= 0 \quad (i = 1, \dots, k) \\ A'_i \bullet X &= b'_i \quad (i = k+1, \dots, m) \\ X &\succeq 0 \end{aligned} \quad (\text{P}_{\text{ref,feas}})$$

where A'_1, \dots, A'_k are of the form as in Theorem 1 with $k \geq 0$, $r_1 > 0, \dots, r_k > 0$, $r_1 + \dots + r_k = n - p$ and matrices A'_{k+1}, \dots, A'_m and scalars b'_{k+1}, \dots, b'_m are arbitrary.

(2) Let (P) and $(P_{\text{ref,feas}})$ be as above and $(P_{\text{ref,feas,red}})$ be the system obtained from $(P_{\text{ref,feas}})$ by replacing the constraint $X \succeq 0$ by $X \in 0 \oplus \mathcal{S}_+^p$. Then $(P_{\text{ref,feas,red}})$ is well-behaved, i.e., for all $C \in \mathcal{S}^n$ the SDP

$$\sup \{ C \bullet X \mid X \text{ is feasible in } (P_{\text{ref,feas,red}}) \} \quad (2.5)$$

has strong duality with its Lagrange dual

$$\inf \left\{ \sum_{i=1}^m y_i b_i : \sum_{i=1}^m y_i A'_i - C \in (0 \oplus \mathcal{S}_+^p)^* \right\}. \quad (2.6)$$

Before the proof we remark that the case $k = 0$ corresponds to (P) being strictly feasible.

Proof of the “if” part in (1) This implication follows similarly as in Theorem 1.

Proof of (2) This implication follows, since $(P_{\text{ref,feas,red}})$ is trivially strictly feasible.

Proof of “only if” in part (1) We modify the algorithm that we used to prove Theorem 1. We now do not assume that (P) is infeasible, nor that the optimal value of (P_{hom}) is zero. As before, we keep iterating in Case 1, until we end up in Case 2, with strong duality between (P_{hom}) and (D_{hom}) . We distinguish two subcases:

Case 2(a): The optimal value of (P_{hom}) is 0. We proceed as before to construct the $(k+1)^{\text{st}}$ equation in (P_{ref}) , which proves infeasibility of (P') .

Case 2(b): The optimal value of (P_{hom}) is positive (i.e., it is $+\infty$). We choose

$$(X, x_0) \in K \times \mathbb{R}$$

to be feasible, with $x_0 > 0$. Then $(1/x_0)X$ is feasible in (P') , but it may not have maximum rank. We now construct a maximum rank feasible solution in (P') . If $n - r \leq 1$, then a simple case checking can complete the construction. If $n - r \geq 2$, then we take

$$(X', x'_0) \in \text{ri } K \times \mathbb{R}$$

as a strictly feasible solution of (P_{hom}) . Then for a small $\epsilon > 0$ we have that

$$(X + \epsilon X', x_0 + \epsilon x'_0) \in \text{ri } K \times \mathbb{R}$$

is feasible in (P_{hom}) with $x_0 + \epsilon x'_0 > 0$. Hence

$$\frac{1}{x_0 + \epsilon x'_0} (X + \epsilon X') \in \text{ri } K$$

is feasible in (P') . ■

Example 2. Consider the feasible semidefinite system with $n = m = 4$, and data

$$A_1 = \begin{pmatrix} -2 & 2 & 7 & -3 \\ 2 & -2 & -4 & -6 \\ 7 & -4 & -15 & -7 \\ -3 & -6 & -7 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 2 & 0 & -3 & 2 \\ 0 & 4 & 6 & 4 \\ -3 & 6 & 14 & 5 \\ 2 & 4 & 5 & 0 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 2 & 0 & -3 & -1 \\ 0 & -1 & -3 & 0 \\ -3 & -3 & -3 & 2 \\ -1 & 0 & 2 & 0 \end{pmatrix},$$

$$A_4 = \begin{pmatrix} -1 & 1 & 4 & 2 \\ 1 & 6 & 11 & 2 \\ 4 & 11 & 16 & 1 \\ 2 & 2 & 1 & 0 \end{pmatrix}, \quad b = (-3, 2, 1, 0).$$

The conversion algorithm produces the following y vectors, and rotation matrices in steps 1

$$y = (1, 2, -1, -1), \quad V = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.7)$$

and

$$y = (0, 1, -2, -1), \quad V = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.8)$$

in step 2 (for brevity, we now do not show the $\sum_i y_i A_i$ matrices, and the intermediate data), to obtain a reformulation with data and maximum rank feasible solution

$$A'_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad A'_2 = \begin{pmatrix} -1 & 0 & -1 & 2 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}, \quad A'_3 = \begin{pmatrix} 2 & -2 & 1 & -1 \\ -2 & 1 & -2 & 1 \\ 1 & -2 & 1 & 0 \\ -1 & 1 & 0 & 0 \end{pmatrix},$$

$$A'_4 = \begin{pmatrix} -1 & 2 & 0 & 2 \\ 2 & 3 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 2 & 0 & 1 & 0 \end{pmatrix}, \quad b' = (0, 0, 1, 0), \quad X = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Note that in the final system the first two constraints prove that the rank of any feasible solution is at most 2. Thus the system itself and X are a certificate that X has maximum rank, hence it is easy to convince a “user” that $(P_{\text{ref,feas,red}})$ (with $p = 2$) is strictly feasible, hence well behaved.

3 Discussion

In this section we discuss our results in some more detail.

We first compare our conversion algorithm with facial reduction algorithms, and describe how to adapt the algorithm of Waki and Muramatsu in [28] to obtain our reformulations.

Remark 1. We say that a convex subset F of a convex set C is a *face of C* , if $x, y \in C$, $1/2(x + y) \in F$ implies that x and y are in F . When (P) is feasible, we define its *minimal cone* as the smallest face of \mathcal{S}_+^n that contains the feasible set of (P) .

Borwein and Wolkowicz in [5, 4] presented an algorithm to find the minimal cone of a feasible, but possibly nonlinear conic system. The algorithm of Waki and Muramatsu [28] is a simplified variant which is applicable to conic linear systems, and is also able to detect infeasibility. We now describe their Algorithm 5.1, which specializes their general algorithm to SDPs, and how to modify it to obtain our reformulations. In the first step they find a $y \in \mathbb{R}^m$ with

$$W := - \sum_{i=1}^m y_i A_i \succeq 0, \quad \sum_{i=1}^m y_i b_i \geq 0.$$

If the only such y is $y = 0$, they stop with $F = \mathcal{S}_+^n$; if $\sum_{i=1}^m y_i b_i > 0$, they stop and report that (P) is infeasible. Otherwise they replace \mathcal{S}_+^n by $\mathcal{S}_+^n \cap W^\perp$, apply a rotation step to reduce the order of the SDP to $n - r$, where r is the rank of W , and continue.

Waki and Muramatsu do not apply elementary row operations. We can obtain our reformulations from their algorithm, if after each iteration $\ell = 0, 1, \dots$ we

- choose the rotation matrix to turn the psd part of W into $I_{r_{\ell+1}}$ for some $r_{\ell+1} \geq 0$.
- perform eros to produce an equality constraint like the $(\ell + 1)$ st constraint in (P_{ref}) , or $(P_{\text{ref,feas}})$.

In their reduction step they also rely on Theorem 20.2 from Rockafellar [22], while we use explicit SDP pairs. In the language of facial reduction, the cone $0 \oplus \mathcal{S}_+^p$ is the minimal cone of the system $(P_{\text{ref,feas,red}})$.

Our conversion algorithm is simplified, so we do not actually need to introduce the concept of faces.

For an alternative approach to ensuring strong duality, called *conic expansion*, we refer to Luo et al [12]; and to [28] for a detailed study of the connection of the two approaches.

We next comment on how to find the optimal solution of a linear function over the original system (P), and on duality properties of this system.

Remark 2. Assume that (P) is feasible, and we used the rotation matrix V to obtain $(P_{\text{ref,feas}})$ from (P). Let $C \in \mathcal{S}^n$. Then one easily verifies

$$\begin{aligned} \sup \{ C \bullet X \mid X \text{ is feasible in } (P) \} &= \sup \{ V^T C V \bullet X \mid X \text{ is feasible in } (P_{\text{ref,feas}}) \} \\ &= \sup \{ V^T C V \bullet X \mid X \text{ is feasible in } (P_{\text{ref,feas,red}}) \}, \end{aligned}$$

and by Theorem 2 the last SDP has strong duality with its Lagrange dual.

Clearly, (P) well behaved (cf. Definition 2), if and only if its ESD-reformulations are. The system (P), or equivalently, system $(P_{\text{ref,feas}})$ may not be well behaved, of course. We refer to the paper [14] for an exact characterization of well-behaved semidefinite systems (in an inequality constrained form).

We next comment on algorithms to generate the data of all SDPs which are either infeasible, or have a maximum rank solution with a prescribed rank.

Remark 3. Let us fix an integer $p \geq 0$, and define the sets

$$\begin{aligned} \text{INFEAS} &= \{ (A_i, b_i)_{i=1}^m \subseteq (\mathcal{S}^n \times \mathbb{R})^m : (P) \text{ is infeasible} \}, \\ \text{FEAS}(p) &= \{ (A_i, b_i)_{i=1}^m \subseteq (\mathcal{S}^n \times \mathbb{R})^m : (P) \text{ is feasible, with maximum rank solution of rank } p \}. \end{aligned}$$

These sets – in general – are nonconvex, neither open, nor closed. Despite this, we can systematically generate *all* of their elements. To generate all elements of INFEAS we use Theorem 1, by which we only need to find systems of the form (P_{ref}) , then reformulate them.

To generate all elements of $\text{FEAS}(p)$ we first find constraint matrices in a system like $(P_{\text{ref,feas}})$, then choose $X \in 0 \oplus \mathcal{S}_{++}^p$, and set $b'_i := A'_i \bullet X$ for all i . By Theorem 2 all elements of $\text{FEAS}(p)$ arise as a reformulation of such a system.

Loosely speaking, Theorems 1 and 2 show that there are only finitely many “schemes” to generate an infeasible semidefinite system, or a feasible system with a maximum rank solution having a prescribed value.

The paper [14] describes a systematic method to generate all well behaved semidefinite systems (in an inequality constrained form), in particular, to generate all linear maps under which the image of \mathcal{S}_+^n is closed. Thus, our algorithms to generate INFEAS and FEAS(p) complement the results of [14].

We next comment on strong infeasibility of (P).

Remark 4. Clearly, (P) is strongly infeasible (i.e., (1.4) is feasible), if and only if it has a reformulation of the form (P_{ref}) with $k = 0$. Thus we can easily generate the data of all strongly infeasible SDPs: we only need to find systems of the form (P_{ref}) with $k = 0$, then reformulate them.

We can also easily generate weakly infeasible instances using Theorem 1: we can choose $k + 1 = m$, and suitable blocks of the A'_i to be nonzero to make sure that they do not have a psd linear combination. (For instance, choosing the block of A'_{k+1} that corresponds to rows $r_1 + \dots + r_{k-1} + 1$ through $r_1 + \dots + r_k$ and the last $n - r_1 - \dots - r_{k+1}$ columns will do.) Then (P_{ref}) is weakly infeasible. It is also likely to be weakly infeasible if we choose the A'_i as above, and m only slightly larger than $k + 1$.

Even when (P) is strongly infeasible, our conversion algorithm may only find a reformulation with $k > 0$. To illustrate this point, consider the system with data

$$A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.9)$$

$$b = (0, -1, 1).$$

This system is strongly infeasible ((1.4) is feasible with $y = (4, 2, 1)$), and it is already in the form of (P_{ref}) with $k = 1$. Our conversion algorithm, however, constructs a reformulation with $k = 2$, since it finds (P_{hom}) to be not strictly feasible in the first two steps.

We next discuss complexity implications.

Remark 5. Theorem 1 implies that semidefinite feasibility is in $\text{co-}\mathcal{NP}$ in the real number model of computing. This result was already proved by Ramana [19] and Klep and Schweighofer [10] via their Farkas’ lemma that relies on extra variables. If one wants to check the infeasibility of the system (P) using our methods, he/she needs to verify that (P_{ref}) is a reformulation of (P), using eros, and a rotation matrix V . Alternatively, one can check that

$$A'_i = V^T \left(\sum_{j=1}^m t_{ij} A_j \right) V \quad (i = 1, \dots, m)$$

holds for V and an invertible matrix $T = (t_{ij})_{i,j=1}^m$.

4 Conclusion

Two well-known pathological phenomena in semidefinite programming are that Farkas’ lemma may fail to prove infeasibility, and strong duality does not hold in general. Here we described an exact certificate of infeasibility, and a strong dual for SDPs, which does not assume any constraint qualification. Such certificates and duals have been known before: see [5, 4, 19, 20, 28, 10].

Our approach appears to be simpler: in particular, the validity of our infeasibility certificate – the infeasibility of the system (P_{ref}) – is almost a tautology (we borrow this terminology from the paper [13] on semidefinite representations). We can also easily convince a “user” that the system $(P_{\text{ref,feas,red}})$ is strictly

feasible (hence strong duality holds for all objective functions). To do so, we use a maximum rank feasible solution, and the system itself, which proves that this solution has maximum rank.

Acknowledgement We thank Rekha Thomas and the anonymous referees for their careful reading of the paper, and their helpful comments.

References

- [1] Abdo Alfakih. On Farkas lemma and dimensional rigidity of bar frameworks. Technical report, <http://arxiv.org/abs/1405.2301>, 2014.
- [2] Aharon Ben-Tal and Arkadii Nemirovskii. *Lectures on modern convex optimization*. MPS/SIAM Series on Optimization. SIAM, Philadelphia, PA, 2001.
- [3] Frédéric J. Bonnans and Alexander Shapiro. *Perturbation analysis of optimization problems*. Springer Series in Operations Research. Springer-Verlag, 2000.
- [4] Jonathan M. Borwein and Henry Wolkowicz. Facial reduction for a cone-convex programming problem. *J. Aust. Math. Soc.*, 30:369–380, 1981.
- [5] Jonathan M. Borwein and Henry Wolkowicz. Regularizing the abstract convex program. *J. Math. Anal. App.*, 83:495–530, 1981.
- [6] Stephen Boyd and Lieven Vandenbergh. *Convex Optimization*. Cambridge University Press, 2004.
- [7] Check-Beng Chua and Levent Tunçel. Invariance and efficiency of convex representations. *Math. Program. B*, 111:113–140, 2008.
- [8] Robert Connelly and Steven Gortler. Iterative universal rigidity. Technical report, <http://arxiv.org/abs/1401.7029>, 2014.
- [9] Osman Güler. *Foundations of Optimization*. Graduate Texts in Mathematics. Springer, 2010.
- [10] Igor Klep and Markus Schweighofer. An exact duality theory for semidefinite programming based on sums of squares. *Math. Oper. Res.*, 38(3):569–590, 2013.
- [11] Bruno Lourenco, Masakazu Muramatsu, and Takashi Tsuchiya. A structural geometrical analysis of weakly infeasible SDPs. *Optimization Online*.
- [12] Zhi-Quan Luo, Jos Sturm, and Shuzhong Zhang. Duality results for conic convex programming. Technical Report Report 9719/A, Erasmus University Rotterdam, Econometric Institute, The Netherlands, 1997.
- [13] David Papp and Farid Alizadeh. Semidefinite characterization of sum-of-squares cones in algebras. *SIAM J. Opt.*, 23(3):1398–1423.
- [14] Gábor Pataki. Bad semidefinite programs: they all look the same. Technical Report available from Optim. Online and <http://arxiv.org/abs/1112.1436>, University of North Carolina at Chapel Hill, under review, 2010.
- [15] Gábor Pataki. On the connection of facially exposed and nice cones. *J. Math. Anal. App.*, 400:211–221, 2013.
- [16] Gábor Pataki. Strong duality in conic linear programming: facial reduction and extended duals. In David Bailey, Heinz H. Bauschke, Frank Garvan, Michel Théra, Jon D. Vanderwerff, and Henry Wolkowicz, editors, *Proceedings of Jonfest: a conference in honour of the 60th birthday of Jon Borwein*. Springer, also available from <http://arxiv.org/abs/1301.7717>, 2013.
- [17] Imre Pólik and Tamás Terlaky. Exact duality for optimization over symmetric cones. Technical report, Lehigh University, Bethlehem, PA, USA, 2009.

- [18] Imre Pólik and Tamás Terlaky. New stopping criteria for detecting infeasibility in conic optimization. *Optim. Lett.*, 3(2):187–198, 2012.
- [19] Motakuri V. Ramana. An exact duality theory for semidefinite programming and its complexity implications. *Math. Program. Ser. B*, 77:129–162, 1997.
- [20] Motakuri V. Ramana, Levent Tunçel, and Henry Wolkowicz. Strong duality for semidefinite programming. *SIAM J. Opt.*, 7(3):641–662, 1997.
- [21] James Renegar. *A Mathematical View of Interior-Point Methods in Convex Optimization*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, USA, 2001.
- [22] Tyrrel R. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, USA, 1970.
- [23] Vera Roshchina. Facially exposed cones are not nice in general. *SIAM J. Opt.*, FJOURNAL =.
- [24] Romesh Saigal, Lieven Vandenbergh, and Henry Wolkowicz, editors. *Handbook of semidefinite programming*. Kluwer Academic Publishers, 2000.
- [25] Michael J. Todd. Semidefinite optimization. *Acta Numer.*, 10:515–560, 2001.
- [26] Levent Tunçel. *Polyhedral and Semidefinite Programming Methods in Combinatorial Optimization*. Fields Institute Monographs, 2011.
- [27] Hayato Waki. How to generate weakly infeasible semidefinite programs via Lasserre’s relaxations for polynomial optimization. *Optim. Lett.*, 6(8):1883–1896, 2012.
- [28] Hayato Waki and Masakazu Muramatsu. Facial reduction algorithms for conic optimization problems. *J. Optim. Theory Appl.*, 158(1):188–215, 2013.