ELEMENTARY REFORMULATION AND SUCCINCT CERTIFICATES IN CONIC LINEAR PROGRAMMING

Minghui Liu

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistics and Operations Research.

Chapel Hill
2015

Approved by:

Gábor Pataki

Shankar Bhamidi

Amarjit Budhiraja

Shu Lu

Scott Provan

ABSTRACT

MINGHUI LIU: Elementary Reformulation and Succinct Certificates in Conic Linear Programming
(Under the direction of Gábor Pataki)

The first part of this thesis deals with infeasibility in semidefinite programs (SDPs). In SDP, unlike in linear programming, Farkas' lemma may fail to prove infeasibility. We obtain an exact, short certificate of infeasibility in SDP by an elementary approach: we reformulate any equality constrained semidefinite system using only elementary row operations, and rotations. When a system is infeasible, the reformulated system is trivially infeasible. When a system is feasible, the reformulated system has strong duality with its Lagrange dual for all objective functions.

The second part is about simple and exact duals, and certificates of infeasibility and weak infeasibility in conic linear programming that do not rely on any constraint qualification and retain most of the simplicity of the Lagrange dual. Some of our infeasibility certificates generalize the row echelon form of a linear system of equations, as they consist of a small, trivially infeasible subsystem. The "easy" proofs – as sufficiency of a certificate to prove infeasibility – are elementary.

We also derived some fundamental geometric corollaries: 1) an exact characterization of when the linear image of a closed convex cone is closed, 2) an exact characterization of nice cones, and 3) bounds on the number of constraints that can be dropped from, or added to a (weakly) infeasible conic LP while keeping it (weakly) infeasible.

Using our infeasibility certificates we generated a public domain library of infeasible and weakly infeasible SDPs. The status of our instances is easy to verify by inspection in exact arithmetic, but they turn out to be challenging for commercial and research codes.

ACKNOWLEDGEMENTS

First, I would like to thank my advisor, Gábor Pataki. He is a great person. I learned a lot from him and the most important thing is that it is fun to work with him on solving math problems.

Also, I would like to thank all faculty and staff members in my department. I really enjoy the time I spent in UNC and STOR department. I think that this department is the best fit for me.

Moreover, I don't know how to express my feelings to my friends: Taoran Cui, Qing Feng, Siliang Gong, Hansi Jiang, Michael Lamm, Gen Li, Siying Li, Bill Shi, Dong Wang, Tao Wang, Jie Xiong, Tan Xu, Dan Yang, Danny Yeung, Liang Yin and Yu Zhang. I just want to say no matter where we go, we will always be friends.

Special thanks to Dongqing Yu, I feel really lucky to have you on my side, and also this paragraph would indicate that the acknowledgement is written in November.

Last but not the least, I would like to thank my parents. You are the best! Hardly can I find words to express my love to you, and there would be no words to describe your love to me.

Let's forget about the grammar and fancy words here. I enjoy five years I spent here at UNC and my heart tells me that I really love you guys!

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Lack of Strong duality in Conic Linear Programming

Conic linear programming (conic LP) is an important subclass of convex optimization. A conic linear programming problem minimizes a linear function over the intersection of an affine subspace and a closed convex cone. It generalizes a lot of useful problems, such as linear programming (LP), semidefinite programming (SDP) and second order cone programming (SOCP). These problems have great modeling power and efficient algorithms have been designed to solve them, so, they are subjects of intensive research. Unlike in LP, there are many challenging open theoretical questions for other conic-LP's. A conic-LP problem is usually stated as:

$$\begin{array}{llll} \sup & \langle c, x \rangle & & \\ s.t. & Ax & \leq_K & b, \end{array} \tag{P}$$

where A is a linear operator, K is a closed convex cone, and we write $s \leq_K t$ to denote $t - s \in K$, and $s <_K t$ to denote $t - s \in \mathrm{ri}K$, where $\mathrm{ri}K$ is the relative interior of K.

A conic linear system is defined as the constraints of a conic LP problem. Duality theory plays an important role in conic LP, and we have the following dual problem associated with the problem (P) defined above:

$$\begin{array}{rrcl} \inf & \langle b, y\rangle & & \\ s.t. & A^*y & = & c \\ & y & \geq_{K^*} & 0, \end{array} \tag{D}$$

where A^* is the adjoint operator of A, and K^* is the dual cone of K.

Similarly to LP, weak duality always holds for conic LP, that is the primal optimal value is always less than or equal to the dual optimal value. Therefore, any feasible solution of primal/dual problem gives an lower/upper bound for the dual/primal optimal value.

However, strong duality doesn't always hold for conic LP. We say that strong duality holds between the primal-dual pair, if their optimal values agree, and the latter optimal value is attained, when it is finite. Strong duality holds in conic LP, when suitable constraint qualifications (CQs) are specified. One of the CQs is Slater condition. When the primal problem (P) satisfies Slater condition, i.e. $\exists x,$ s.t. $Ax <_K b$, the primal and dual optimal values are the same and the dual optimal value is attained when they are finite. However, strong duality doesn't always hold when the appropriate CQ is not specified and the cone is not polyhedral. Given the assumption that the primal problem is bounded and optimal value is attained, the dual problem can be infeasible, have a different optimal value or unattainable solution.

A conic LP system is called well behaved, if strong duality holds between the primal-dual pair for all objective functions. A system is badly behaved, if it is not well behaved. It has been shown that whether a system is well-behaved or not is related to one of the most fundamental questions of convex analysis: when the linear image of a closed convex cone is still closed? Pataki in [27] has given some intuitive results for this problem and a characterization of well/badly behaved systems based on these results is given in [28] as well.

Due to the lack of strong duality, the simple generalization of the traditional Farkas' lemma for LP doesn't apply to all other conic LPs. The traditional Farkas' lemma states that the infeasibility of the original system is equivalent to the feasibility of a so called alternative system. The alternative system is an exact certificate of infeasibility of the original system. For general conic LPs, the feasibility of the alternative system trivially implies that the original system is infeasible; however, there are also situations that both the original and alternative systems are infeasible, meaning that the simple alternative system fails to be an exact certificate of infeasibility. When there is no such certificate of infeasibility, we call the system being weakly infeasible. From another point of view, these problems are infeasible, but they can become either feasible or infeasible after a small perturbation of the problem data. It is difficult to detect infeasibility for weakly infeasible problems, since practical solvers usually deal with approximate solutions and certificates. Currently, none of the conic optimization solvers indicate to the user that the feasibility of the problem is in question and they simply return an approximate solution or declare feasibility depending on their stopping criteria as pointed out in [33].

There are two fundamental approaches to obtain strong duality without assuming any CQ. One of them is Ramana's extended dual for semidefinite programs in [35], while another approach is the facial reduction algorithm (FRA) proposed by Borwein and Wolkowicz in [11; 10]. Original FRA can be used to prove the correctness of Ramana's extended dual and this connection was illustrated in [36].

Ramana's extended dual is an explicit semidefinite program with a large but polynomially many variables and constraints. It is feasible if and only if the primal problem is bounded and it has the same optimal value with the primal, and the optimal value is attained when the primal is bounded.

Facial reduction algorithm constructs a so-called minimal cone which is a special face of the original closed convex cone K. When we replace the original cone with the minimal cone, the feasible region remains the same and the system becomes strictly feasible; thereby, the strong duality always holds in the new primal-dual pair. FRA has several variants, and they all try to find a sequence of elements in K to construct a decreasing chain of faces starting with K and ending with the minimal cone. The fundamental facial reduction algorithm of Borwein and Wolkowicz ensures strong duality in a possibly nonlinear conic system. While this original algorithm requires the assumption of feasibility of the system, Waki and Muramatsu presented a simplified FRA for conic-LP without assumption of feasibility in [48]. Pataki in [30] also described a simplified facial reduction algorithm and generalized Ramana's dual to conic linear systems over nice cones. This class of cones includes the semidefinite cone, second order cone and other important cones, so this generalization can be applied to many important problems. Facial reduction algorithm can be considered as an theoretical procedure which can provide some intuitive insight on theoretical work related to the lack of strong duality. Also, it has been shown to be a useful preprocessing method for the implementation of semidefinite programming solver.

1.2 Introduction to Semidefinite Programming

Besides the general conic linear programming problem, we will study semidefinite programming (SDP), an important subclass, in detail in this dissertation. Semidefinite programming is defined to optimize a linear function subject to a linear matrix inequality:

$$\begin{array}{lrcl} \sup & \langle c, x \rangle & & \\ s.t. & \sum_{i=1}^m x_i A_i & \preceq & B, \end{array} \tag{1.2.1}$$

where $c, x \in \mathbb{R}^n$, $\langle c, x \rangle$ is the inner product of two n dimensional vectors, and $A \preceq B$ means that $B - A$ is positive semidefinite.

It can be regarded as a natural extension of linear programming in two ways. First, it can be obtained from LP by replacing component wise inequalities between vectors with matrix inequalities. Second, it can also be obtained by replacing the first orthant with the cone of positive semidefinite matrices in the general conic LP definition. It also has a dual problem defined as:

$$\begin{aligned} \inf \quad & B \bullet Y & & \\ s.t. \quad & A_i \bullet Y & = & \; c_i \\ & Y & \succeq & \; 0, \end{aligned} \tag{1.2.2}$$

where the $\bullet$ dot product of symmetric matrices is the trace of their regular product.

Semidefinite programming has great modeling power: positive semidefinite constraints appear directly in many applications, and many types of constraints in convex optimization can also be cast as semidefinite constraints. These constraints include linear inequalities, convex quadratic inequalities, lower bounds on matrix norms, lower bounds on determinants of symmetric positive semidefinite matrices, lower bounds on the geometric mean of a nonnegative vector and so on. Using these constraints, problems such as quadratically constrained quadratic programming, matrix eigenvalue and norm maximization, and pattern separation by ellipsoids, can all be modeled as SDP. A good survey on SDP can be found in [45].

SDP can also be used for developing approximation algorithms for NP-hard combinatorial optimization problems. In many cases, the SDP relaxation is very tight in practice and the optimal solution to the SDP relaxation can be converted to a feasible solution for the original problem with good objective value. One famous example is the maximum cut problem, one of Karp's original NP-complete problems; it tries to find the maximum sum of weights of a cut in a graph. Goemans and Williamson in [19] constructed an SDP relaxation and solve it within an arbitrarily small additive error. By doing so, they have shown that their method achieves an expected approximation ratio of 0.87856 which is much better than any previously known result.

Efficient algorithms have been designed and implemented for solving SDP. Most solvers, such as SeDuMi, SDPT3 and MOSEK, implement interior point methods. While these solvers performs well in general, they can't always tackle problems that are not strictly feasible. Meanwhile, most theoretical convergence results for interior point methods for semidefinite programming rely on the assumption that SDP satisfies strict feasibility. In order to deal with these issues, some preprocessing methods have been proposed. One of them is homogeneous self-dual embedding method and it has been implemented in SeDuMi. More details about the implementation of SeDuMi can be found in [43]. An alternative way to preprocess an SDP is to use FRA. Cheung in [14] studied numerical issues in implementation of this method and showed the result on a semidefinite programming relaxation of the NP-hard side chain positioning problem by obtaining a smaller and more stable SDP problem.

1.3 New contributions and key techniques

In this dissertation, the first contribution is focused on semidefinite programming. Simple generalization of the traditional Farkas' Lemma for LP fails to prove infeasibility for all semidefinite systems. We obtain an exact, short certificate of infeasibility for the following semidefinite system in Chapter 2:

$$\begin{aligned} A_i \bullet X &= b_i \ (i = 1, \ldots, m) \\ X &\succeq 0. \end{aligned} \tag{1.3.3}$$

Our main technique is based on the reformulation of aforementioned semidefinite system. We call our reformulation method elementary semidefinite (ESD-) reformulation. It applies elementary row operations and rotation on the original system and is able to preserve feasibility of the original system.

When the original system is infeasible, we can obtain an ESD-reformulation with a staircase like structure with which its infeasibility is trivial to verify by inspection. This result has a natural application: it can be used to generate all infeasible semidefinite programs. While there is a good selection of infeasible problems in problem libraries for linear, quadratic and general nonlinear optimization problems, there are only a few small and trivial infeasible SDP problems in the SDPLIB library and DIMACS Challenge problems, so a comprehensive library of infeasible problems would be extremely useful for development and testing of new stopping criteria for the SDP solvers as mentioned in [33].

With similar technique, we can obtain an ESD-reformulation for feasible semidefinite systems as well. Strong duality always holds between the reformulated system and its Lagrange dual for all objective functions and it can be verified easily due to the nice combinatorial structure of the reformulated system. This result can be used to generate the constraints of all feasible SDPs whose maximum rank feasible solution has a prescribed rank. It is a good complement to the result in [28] that gives an exact characterization of well/badly behaved semidefinite systems in an inequality constrained form.

The sets of data for infeasible and feasible semidefinite systems are non convex, neither open, nor closed in general, so it is a rather surprising result that we can systematically generate all of their elements. The procedure of obtaining the reformulation for infeasible and feasible semidefinite systems can be unified as one algorithm which can be considered as one theoretical algorithm of detecting infeasibility. We believe that it would also be useful to verify the infeasibility of small instances.

Our another main contribution is short certificates of infeasibility and weak infeasibility for conic LPs. We generalize ESD-reformulation, which is only applied on semidefinite systems, to a reformulation on primal-dual pairs of conic LP's. By using a classic theorem of the alternative, we present a simplified version of facial reduction algorithm with a half page proof of correctness. This facial reduction algorithm serves as our main tool to construct strong duals for conic LP problems for both primal and dual forms, with which we obtain exact certificates for infeasible conic linear systems and conic linear systems that are not strongly infeasible. We combine these two certificates to obtain a certificate for weakly infeasible systems. We generate a library of weakly infeasible SDPs. The status of our instances is easy to verify by inspection, but they turn out to be challenging for commercial and research codes.

Besides, We also derived some fundamental geometric corollaries: 1) an exact characterization of when the linear image of a closed convex cone is closed, 2) an exact characterization of nice cones, and 3) bounds on the number of constraints that can be dropped from, or added to a (weakly) infeasible conic LP while keeping it (weakly) infeasible.

1.4 Outline of dissertation

- In chapter 2, we will focus on semidefinite programming. We demonstrate our result on exact duality in semidefinite programming based on elementary semidefinite (ESD-) reformulations. When the original system is infeasible, we obtain an exact, short certificate of infeasibility by using ESD-reformulation. The reformulated system has a nice combinatorial structure with which its infeasibility is easy to check. When the original system is feasible, we can obtain a similar reformulation, which trivially has strong duality with its Lagrange dual for all objective functions. With these results, we obtain algorithms to systematically generate the constraints of all infeasible semidefinite programs, and the constraints of all feasible SDPs whose maximum rank feasible solution has a prescribed rank. Our method of obtaining the reformulation can be considered as a theoretical algorithm to detect infeasibility and we believe that it is useful to verify the infeasibility of small instances. Main result can be found in section 2.2. Algorithms of construction of ESD-reformulation for infeasible and feasible semidefinite systems can be found in section 2.5 and 2.7 respectively. Two intuitive examples are presented in section 2.6 and 2.8 to illustrate our algorithm.

- In chapter 3, we will present our results on exact certificates of infeasibility and weak infeasibility in general conic LPs. The main result with the proof of easy direction can be found in section 3.2. We further simplified facial reduction algorithm based on a classical theorem of the alternative. A strong dual for problems of both primal and dual forms follows by using facial reduction sequences. They are both discussed in section 3.6. We prove the correctness of certificates of infeasibility and weak infeasibility for conic LPs in section 3.7. We specify our results on SDP in section 3.8. Finally, we demonstrate algorithms to systematically generate weakly infeasible semidefinite systems in section 3.9, and computational experiments on popular SDP solvers with our infeasible/weakly infeasible SDP problem instances are presented in section 3.10.

- In chapter 4, we describes our results on several basic questions in convex analysis. We give the result on when the linear image of a closed convex cone is closed and an exact characterization of nice cones. We also describe bounds on the number of constraints that can be dropped from, or added to a (weakly) infeasible conic LP while keeping it (weakly) infeasible.

CHAPTER 2

Exact duality in semidefinite programming based on elementary reformulations

2.1 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

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

using only elementary row operations, and rotations. When (P) is infeasible, the reformulated system is trivially infeasible. When (P) is feasible, the reformulated system has strong duality with its Lagrange dual for all objective functions. As a corollary, we obtain algorithms to generate the constraints of *all* infeasible SDPs and the constraints of *all* feasible SDPs with a fixed rank maximal solution.

We give two methods to construct our elementary reformulations. One is direct, and based on a simplified facial reduction algorithm, and the other is obtained by adapting the facial reduction algorithm of Waki and Muramatsu.

In somewhat different language, our reformulations provide a standard form of spectrahedra, to easily verify either their emptiness, or a tight upper bound on the rank of feasible solutions.

2.2 Introduction and the main result

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 Lagrange dual, and the simplest version of Farkas' lemma may fail to prove infeasibility in semidefinite programming.

Several alternatives of the traditional Lagrange dual, and Farkas' lemma are known, which we will review in detail below: see Borwein and Wolkowicz [11; 10]; Ramana [35]; Ramana, Tunçel, and Wolkowicz [36]; Klep and Schweighofer [20]; Waki and Muramatsu [49], and the second author [30].

We 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

$$\begin{array}{rcl} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet X & = & 0 \\ \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \bullet X & = & -1 \\ X & \succeq & 0, \end{array} \tag{2.2.1}$$

which is trivially infeasible; to see why, 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 reformulation – in Theorem 2 – which 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

$$\begin{array}{rcl} A_i' \bullet X & = & 0 \, (i = 1, \dots, k) \\ A_{k+1}' \bullet X & = & -1 \\ A_i' \bullet X & = & b_i' \, (i = k+2, \dots, m) \\ X & \succeq & 0 \end{array} \qquad (\mathrm{P}_{\mathrm{ref}})$$

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

$$A_i' = \begin{pmatrix} \overbrace{\qquad \times \qquad}^{r_1 + \ldots + r_{i-1}} & \overbrace{\times}^{r_i} & \overbrace{\qquad \times \qquad}^{n - r_1 - \ldots - r_i} \\ \times & I & 0 \\ \times & 0 & 0 \end{pmatrix}$$

for $i = 1, \dots, k+1$, with $r_1, \dots, r_k > 0, r_{k+1} \geq 0$, the × 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 a very simple, 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

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

contradicting the $(k+1)^{st}$ constraint in (P_{ref}). □

Note that Theorem 1 allows us to systematically generate *all* infeasible semidefinite systems: to do so, we only need to generate systems of the form (P_{ref}), and reformulate them. We comment more on this in Section 2.10.

2.3 Literature Review

We now review relevant literature in detail, and its connection to our results. For surveys and textbooks on SDP, we refer to Todd [44]; Ben-Tal and Nemirovskii [5]; Saigal et al [41]; Boyd and Vandenberghe [12]. For treatments of their duality theory see Bonnans and Shapiro [7]; Renegar [38] and Güler [18].

The fundamental facial reduction algorithm of Borwein and Wolkowicz [11; 10] ensures strong duality in a possibly nonlinear conic system by replacing the underlying cone by a suitable face. Ramana in [35] 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 [36] proved the correctness of Ramana's dual from the algorithm in [11; 10].

While the algorithms in [11; 10] assume that the system is feasible, Waki and Muramatsu in [49] presented a simplified facial reduction algorithm for conic linear systems, which disposes with this assumption, and allows one to prove infeasibility. We state here that our reformulations can be obtained by suitably modifying the algorithm in [49]; we describe the connection in detail in Section 2.10. At the same time we provide a direct, and entirely elementary construction.

More recently, Klep and Schweighofer in [20] proposed 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.

The second author in [30] described a simplified facial reduction algorithm, and generalized Ramana's dual to conic linear systems over *nice* cones (for literature on nice cones, see [15], [40], [29]). We refer to Pólik and Terlaky [32] for a generalization of Ramana's dual for conic LPs over homogeneous cones. Elementary reformulations of semidefinite systems first appear in [28]. There the second author uses them to bring a system into a form to easily check whether it has strong duality with its dual for all objective functions.

Several papers, see for instance Pólik and Terlaky [33] on stopping criteria for conic optimization, point to the need of having more infeasible instances 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 [35; 20], and used it in an SDP based proof of a result of Connelly and Gortler on rigidity [16]. Our Theorem 2 gives such a certificate using elementary 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 [47] for a systematic method to generate weakly infeasible SDPs from Lasserre's relaxation of polynomial optimization problems. Lourenco et al. [22] 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 ($P_{\rm ref}$), 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 2.7 we present our reformulation of feasible systems. Here we modify our algorithm to construct the reformulation ($P_{\rm ref}$) (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.

2.4 Some preliminaries

We denote by $\mathcal{S}^n$, $\mathcal{S}^n_+$, and $\mathsf{PD}n$ 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 $\operatorname{ri} K$, and its dual cone by K^*, i.e.,

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

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 $\operatorname{ri} K = 0 \oplus \mathsf{PD}p$, 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

$$\begin{pmatrix} I_{n-p} & 0 \\ 0 & Q \end{pmatrix}^T Z \begin{pmatrix} I_{n-p} & 0 \\ 0 & Q \end{pmatrix} = \begin{pmatrix} Z_{11} & Z_{12}Q \\ Q^T Z_{12}^T & Q^T Z_{22} Q \end{pmatrix} \tag{2.4.2}$$

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:

$$\begin{array}{rcl} \mathcal{A}(x) & = & b \\ \mathcal{B}(x) & \leq_K & d, \end{array} \tag{2.4.3}$$

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

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

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) (2.4.3) is well behaved, if strong duality holds betweeen (P_{gen}) and (D_{gen}) for *all* c objective functions.

(3) (2.4.3) is strictly feasible, if $d - \mathcal{B}(x) \in \text{ri}K$ for some feasible x.

We will use the following

Lemma 1. If (2.4.3) is strictly feasible, or K is polyhedral, then (2.4.3) is well behaved.

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

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

$$\begin{array}{rcl} y & \in & \mathbb{R}^m \\ \sum_{i=1}^m y_i A_i & \succeq & 0 \\ \sum_{i=1}^m y_i b_i & = & -1; \end{array} \tag{2.4.4}$$

in this case we say that (P) is strongly infeasible. Note that system $(P_{\rm alt})$ generalizes Farkas' lemma from linear programming. However, (P) and $(P_{\rm alt})$ may both be infeasible, in which case we say that (P) is weakly infeasible. For instance, the system (2.2.1) is weakly infeasible.

2.5 The certificate of infeasibility and its proof

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

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

where for some $\ell \geq 0$ and $r_1 > 0, \dots, r_\ell > 0$ the A_i' matrices are as 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

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

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

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

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

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

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

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

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

Figure 2.1: 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 2.1. 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 $A'_1, \ldots, A'_\ell$ remain in the required form: cf. equation (2.4.2).

Second, we prove that after finitely many steps our algorithm ends in Case 2. In each iteration both ℓ and $r = r_1 + \ldots + r_\ell$ 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 $X \in 0 \oplus \mathsf{PD} n - r$ we can choose x_0 to satisfy the last equality constraint of (P_{hom}), hence at this point we are in Case 2. $\square$

2.6 Example of infeasibility reformulation

We next illustrate our algorithm:

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

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

$$A_4 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad A_5 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & -1 & 1 & 0 \end{pmatrix}, \quad A_6 = \begin{pmatrix} -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & -1 & 1 \end{pmatrix},$$

$$b = \quad (0, 6, -3, 2, 1, 3).$$

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

$$A'_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad A'_2 = \begin{pmatrix} -1 & 3 & 1 & -2 \\ 3 & -2 & 2 & 3 \\ 1 & 2 & 4 & -3 \\ -2 & 3 & -3 & 3 \end{pmatrix}, \quad A'_3 = \begin{pmatrix} -1 & 2 & -2 & 0 \\ 2 & -5 & 2 & 2 \\ -2 & 2 & -3 & -2 \\ 0 & 2 & -2 & -1 \end{pmatrix},$$

$$A'_4 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & -1 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad A'_5 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 2 & 0 & -1 \\ 0 & 0 & 1 & 1 \\ 0 & -1 & 1 & 0 \end{pmatrix}, \quad A'_6 = \begin{pmatrix} -1 & 1 & 0 & -1 \\ 1 & -1 & 1 & 1 \\ 0 & 1 & 0 & -1 \\ -1 & 1 & -1 & 1 \end{pmatrix},$$

$$b' = \quad (0, 6, -3, 2, 1, 3).$$

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

$$A'_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad A'_2 = \begin{pmatrix} 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad A'_3 = \begin{pmatrix} -1 & 2 & -6 & 0 \\ 2 & -5 & 12 & 2 \\ -6 & 12 & -31 & -6 \\ 0 & 2 & -6 & -1 \end{pmatrix},$$

$$A'_4 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & -3 & 0 \\ 1 & -3 & 7 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad A'_5 = \begin{pmatrix} 0 & -1 & 2 & 0 \\ -1 & 2 & -4 & -1 \\ 2 & -4 & 9 & 3 \\ 0 & -1 & 3 & 0 \end{pmatrix}, \quad A'_6 = \begin{pmatrix} -1 & 1 & -2 & -1 \\ 1 & -1 & 3 & 1 \\ -2 & 3 & -8 & -3 \\ -1 & 1 & -3 & 1 \end{pmatrix},$$

$$b' = (0, 0, -3, 2, 1, 3).$$

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

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

$$A'_4 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & -3 & 0 \\ 1 & -3 & 7 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad A'_5 = \begin{pmatrix} 0 & -1 & 2 & 0 \\ -1 & 2 & -4 & -1 \\ 2 & -4 & 9 & 3 \\ 0 & -1 & 3 & 0 \end{pmatrix}, \quad A'_6 = \begin{pmatrix} -1 & 1 & -2 & -1 \\ 1 & -1 & 3 & 1 \\ -2 & 3 & -8 & -3 \\ -1 & 1 & -3 & 1 \end{pmatrix},$$

$$b' = (0, 0, -1, 2, 1, 3).$$

2.7 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 maximum rank solution of rank p if and only if it has a a feasible solution with rank p and an elementary reformulation

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

where $A'_1, \ldots, A'_k$ are as in Theorem 1,

$$k \geq 0, \, r_1 > 0, \ldots, r_k > 0, \, r_1 + \cdots + r_k = n - p,$$

and matrices $A'_{k+1}, \ldots, A'_m$ and scalars $b'_{k+1}, \ldots, b'_m$ are arbitrary.

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

$$\sup \{ \, C \bullet X \mid X \text{ is feasible in } (\mathrm{P}_{\mathrm{ref,feas,red}}) \, \} \tag{2.7.5}$$

has strong duality with its Lagrange dual

$$\inf \{ \sum_{i=1}^m y_i b_i \, : \, \sum_{i=1}^m y_i A'_i - C \in (0 \oplus \mathcal{S}^p_+)^* \, \}. \tag{2.7.6}$$

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

Proof of "if" in (1) This implication follows similarly as in Theorem 1.

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

Proof of "only if" in (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 $(k+1)^{st}$ 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

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

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

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

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

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

is feasible in (P$'$). □

2.8 Example of feasibility reformulation

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

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

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

The conversion algorithm produces the following y vectors, and rotation matrices: it produces

$$y = (1,2,-1,-1), \quad V = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.8.7}$$

in step 1, and

$$y = (0,1,-2,-1), \quad V = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.8.8}$$

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

$$A_1' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad A_2' = \begin{pmatrix} -1 & 0 & -1 & 2 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 \end{pmatrix}, \quad A_3' = \begin{pmatrix} 2 & -2 & 1 & -1 \\ -2 & 1 & -2 & 1 \\ 1 & -2 & 1 & 0 \\ -1 & 1 & 0 & 0 \end{pmatrix},$$

$$A_4' = \begin{pmatrix} -1 & 2 & 0 & 2 \\ 2 & 3 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 2 & 0 & 1 & 0 \end{pmatrix}, \quad b' = (0,\ 0,\ 1,\ 0), \qquad X = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

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 ($\mathrm{P}_{\mathrm{ref,feas,red}}$) (with $p = 2$) is strictly feasible, hence well behaved.

2.9 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 [49] 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).

The algorithm of Borwein and Wolkowicz [11; 10] finds the minimal cone of a feasible, but possibly nonlinear conic system. The algorithm of Waki and Muramatsu [49] is a simplified variant which is applicable to conic linear systems, and can 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 $y \in \mathbb{R}^m$ with

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

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 $\ell = 0, 1, \ldots$ we

- choose the rotation matrix to turn the psd part of W into I_{r_ℓ} for some $r_\ell \geq 0$.

- add eros to produce an equality constraint like the ℓth constraint in (P_{ref}), or ($\mathrm{P}_{\mathrm{ref,feas}}$).

In their reduction step they also rely on Theorem 20.2 from Rockafellar [39], while we use explicit SDP pairs. For an alternative approach to ensuring strong duality, called *conic expansion*, we refer to Luo et al [23]; and to [49] 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 ($\mathrm{P}_{\mathrm{ref,feas}}$) from (P). Let $C \in \mathcal{S}^n$. Then one easily verifies

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

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

Clearly, (P) is well behaved, if and only if its ESD-reformulations are. The system (P), or equivalently, system ($\mathrm{P}_{\mathrm{ref,feas}}$) may not be well behaved, of course. We refer to [28] 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

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

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 FEAS(p) we first find constraint matrices in a system like ($\mathrm{P}_{\mathrm{ref,feas}}$), then choose $X \in 0 \oplus \mathsf{PD}p,$ and set $b_i' := A_i' \bullet X$ for all i. By Theorem 2 all elements of 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, and a feasible system with a maximum rank solution having a prescribed rank.

The paper [28] 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 [28].

We next comment on strong infeasibility of (P).

Remark 4. Clearly, (P) is strongly infeasible (i.e., (P_{alt}) 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' in (P_{ref}) 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 $r_1 + \cdots + r_{k-1} + 1$ through $r_1 + \cdots + r_k$ and the last $n - r_1 - \cdots - r_{k+1}$ 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

$$\begin{aligned} A_1 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_2 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \\ b &= \quad (0, -1, 1). \end{aligned} \tag{2.9.9}$$

This system is strongly infeasible ((P_{alt}) 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 co-$\mathcal{NP}$ in the real number model of computing. This result was already proved by Ramana [35] and Klep and Schweighofer [20] via their Farkas' lemma that relies on extra variables. To check the infeasibility of (P) using our methods, we need to verify that (P_{ref}) is a reformulation of (P), using eros, and a rotation matrix V. Alternatively, one can check that

$$A'_i = V^T \big(\sum_{j=1}^{m} t_{ij} A_j \big) V \; (i = 1, \ldots, m)$$

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

2.10 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 do not assume any constraint qualification. Such certificates and duals have been known before: see [11; 10; 35; 36; 49; 20; 30].

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 [25] on semidefinite representations). We can also easily convince a "user" that the system ($\mathrm{P}_{\mathrm{ref,feas,red}}$) is well behaved (i.e., 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.

In a somewhat different language, elementary reformulations provide a standard form of spectrahedra – the feasible sets of SDPs – to easily check their emptiness, or a tight upper bound on the rank of feasible solutions. We hope that these standard forms will be useful in studying the geometry of spectrahedra – a subject of intensive recent research [24; 6; 46; 42].

CHAPTER 3

Short certificates in conic linear programming: infeasibility and weak infeasibility

3.1 Abstract

We describe simple and exact duals, and certificates of infeasibility and weak infeasibility in conic linear programming which do not rely on any constraint qualification and retain most of the simplicity of the Lagrange dual. Some of our infeasibility certificates generalize the row echelon form of a linear system of equations, and the "easy" proofs – as sufficiency of a certificate to prove infeasibility – are elementary. We derive some fundamental geometric corollaries: 1) an exact characterization of when the linear image of a closed convex cone is closed, 2) an exact characterization of nice cones, and 3) bounds on the number of constraints that can be dropped from, or added to a (weakly) infeasible conic LP while keeping it (weakly) infeasible.

Using our infeasibility certificates we generate a public domain library of infeasible and weakly infeasible SDPs. The status of our instances is easy to verify by inspection in exact arithmetic, but they turn out to be challenging for commercial and research codes.

3.2 Introduction and main results

Conic linear programs generalize linear programming by replacing the nonnegative orthant by a closed convex cone. They model a wide variety of practical optimization problems, and inherit some of the duality theory of linear programming: the Lagrange dual provides a bound on their optimal value and a simple generalization of Farkas' lemma yields a proof of infeasibility.

However, strong duality may fail (i.e., the Lagrange dual may yield a positive gap, or not attain its optimal value), and the simple Farkas' lemma may fail to prove infeasibility. All these pathologies occur in semidefinite programs (SDPs) and second order conic programs (SOCPs), arguably the most useful classes of conic LPs.

To ground our discussion, we consider a conic linear program of the form

$$\begin{array}{ll} \sup & \langle c, x\rangle \\ s.t. & Ax \leq_K b, \end{array} \tag{P}$$

where $A : \mathbb{R}^m \rightarrow Y$ is a linear map, Y is a finite dimensional euclidean space, $K \subseteq Y$ is a closed convex cone, and $s \leq_K t$ stands for $t - s \in K$. Letting A^* be the adjoint of $A,$ and K^* the dual cone of $K,$ the Lagrange dual of (P) is

$$\begin{array}{llll} \inf & \langle b, y\rangle & & \\ s.t. & A^*y & = & c \\ & y & \geq_{K^*} & 0. \end{array} \tag{D}$$

Weak duality – the inequality $\langle c, x\rangle \leq \langle b, y\rangle$ between a pair of feasible solutions – is trivial. However, the optimal values of (P) and of (D) may differ, and/or may not be attained.

A suitable conic linear system can prove the infeasibility of (P) or of (D). Since (for convenience) we focus mostly on infeasibility of (D), we state its alternative system below:

$$\begin{array}{rcl} Ax & \geq_K & 0 \\ \langle c, x\rangle & = & -1. \end{array} \tag{D_{alt}}$$

When (D_{alt}) is feasible, (D) is trivially infeasible, and we call it *strongly infeasible.* However, (D_{alt}) and (D) may both be infeasible, and in this case we call (D) *weakly infeasible.* Thus (D_{alt}) is not an exact certificate of infeasibility.

While a suitable constraint qualification (CQ), (as the existence of an interior feasible solution in (P)) can make (D), and (D_{alt}) exact, such CQs frequently fail to hold in practice.

Three known approaches, which we review in detail below, provide exact duals, and certificates of infeasibility for conic LPs: facial reduction algorithms – see Borwein and Wolkowicz [11], Waki and Muramatsu [49], Pataki [30]; extended duals for SDPs and generalizations – see Ramana [35], Klep and Schweighofer [20] and [30]; and elementary reformulation for SDPs – see Pataki [28] and Liu and Pataki [21]. For the connection of these approaches, see Ramana, Tunçel and Wolkowicz [36], [30] and [21].

The nonexactness of the Lagrange dual and of Farkas' lemma is caused by the possible nonclosedness of the linear image of K or a related cone. For related studies, see Bauschke and Borwein [4]; Pataki [27]; and Borwein and Moors [8; 9].

Here we unify, simplify and extend these approaches and develop a robust calculus of exact duals, and certificates of infeasibility in conic LPs with the following features:

(1) They do not rely on a CQ, and inherit most of the simplicity of the Lagrange dual: some of our infeasibility certificates generalize the row echelon form of a linear system of equations, and the "easy" proofs, as weak duality, and the proofs of infeasibility and weak infeasibility are nearly as simple as proofs in linear programming duality (see Sections 3.6 and 3.7). Some of our duals generalize the exact SDP duals of Ramana [35] and Klep and Schweighofer [20] to the context of general conic linear programming.

(2) They provide algorithms to generate *all* infeasible conic LP instances over several important cones (see Sections 3.7 and 3.11), and *all* weakly infeasible SDPs in a natural class (Section 3.9);

(3) The above algorithms are easy to implement, and provide a challenging test set of infeasible and weakly infeasible SDPs: while we can verify the status of our instances by inspection in exact arithmetic, they are difficult for commercial and research codes (Section 3.10).

(4) Of possible independent interest is an elementary facial reduction algorithm (Section 3.6) with a much simplified proof of convergence; and the geometry of the *facial*

reduction cone, a cone that we introduce and use to encode facial reduction algorithms (see Lemma 2).

We now describe our main tools, and some of our main results with full proofs of the "easy" directions. We will often reformulate a conic LP in a suitable form from which its status (as infeasibility) is easy to read off. This process is akin to bringing a matrix to row echelon form, and most of the operations we use indeed come from Gaussian elimination. To begin, we represent A and A^* as

$$Ax \;=\; \sum_{i=1}^{m} x_i a_i,\; A^*y \;=\; (\langle a_1, y\rangle, \dots, \langle a_m, y\rangle)^T, \text{ where } a_i \in Y \;\text{ for } i = 1, \dots, m.$$

Definition 3. We obtain an *elementary reformulation* or *reformulation* of (P)-(D) by a sequence of the operations:

(1) Replace (a_i, c_i) by $(A\lambda, \langle c, \lambda\rangle)$ for some $i \in \{1, \dots, m\}$, where $\lambda \in \mathbb{R}^m$, $\lambda_i \neq 0$.

(2) Switch (a_i, c_i) with (a_j, c_j), where $i \neq j$.

(3) Replace b by $b + A\mu$, where $\mu \in \mathbb{R}^m$.

If $K = K^*$ we also allow the operation:

(4) Replace a_i by $Ta_i (i = 1, \dots, m)$ and b by Tb, where T is an invertible linear map with $TK = K$.

We call operations (1)-(3) *elementary row operations (eros)*. Sometimes we reformulate only (P) or (D), or only the underlying systems, ignoring the objective function. Clearly, a conic linear system is infeasible, strongly infeasible, etc., exactly when its elementary reformulations are.

Facial reduction cones "encode" a facial reduction algorithm, in a sense that we make precise later, and will replace the usual dual cone to make our duals and certificates exact.

Definition 4. The order k facial reduction cone of K is the set

$$\mathrm{FR}_k(K) = \{\, (y_1, \dots, y_k) : k \geq 0,\, y_1 \in K^*,\, y_i \in (K \cap y_1^\perp \cap \dots \cap y_{i-1}^\perp)^*, i = 2, \dots k \,\}.$$

We drop the index k when its value is clear from context. Clearly $K^* = \mathrm{FR}_1(K) \subseteq \mathrm{FR}_2(K) \subseteq \dots$ holds. Surprisingly, $\mathrm{FR}_k(K)$ is convex, which is only closed in trivial cases, but behaves as well as the usual dual cone K^* under the usual operations on convex sets – see Lemma 2.

We now state an excerpt of our main results with full proofs of the "easy" directions:

Theorem I If K is a general closed convex cone, then

(1) (D) is infeasible, if and only if it has a reformulation

$$\begin{array}{rcl} \langle a_i', y \rangle & = & 0\,(i = 1, \dots, k) \\ \langle a_{k+1}', y \rangle & = & -1 \\ \langle a_i', y \rangle & = & c_i'\,(i = k+2, \dots, m) \\ y & \geq_{K^*} & 0 \end{array} \qquad (D_{\mathrm{ref}})$$

where $k \geq 0$, $(a_1', \dots, a_{k+1}') \in \mathrm{FR}(K^*)$.

(2) (D) is not strongly infeasible, if and only if there is $(y_1, \dots, y_{\ell+1}) \in \mathrm{FR}(K)$, such that

$$\begin{array}{rcl} A^* y_i & = & 0\,(i = 1, \dots, \ell) \\ A^* y_{\ell+1} & = & c. \end{array}$$

□

To see how Theorem I extends known results, first assume that K^* is the whole space, hence (D) is a linear system of equations. Then $\mathrm{FR}_{k+1}(K^*) = \{0\}^{k+1}$, and the constraint $\langle 0, y \rangle = -1$ in (D_{ref}) proves infeasibility, thus (D_{ref}) generalizes the row echelon form of a linear system of equations. If $k = 0$ in part (1), then $a'_1 \in K,$ and $(a'_1, -1) = (Ax, \langle c, x \rangle)$ for some $x,$ so (D) is strongly infeasible. If $\ell = 0$ in part (2), then (D) is actually feasible.

Also, the "if" directions are trivial:

Proof of if in part (1) We prove that (D_{ref}) is infeasible, so suppose that y is feasible in it to obtain the contradiction

$$y \in K^* \cap a_1'^{\perp} \cap \ldots \cap a_k'^{\perp} \Rightarrow \langle a'_{k+1}, y \rangle \geq 0.$$

Proof of if in part (2) We prove that (D) is not strongly infeasible, so suppose it is. Let x be feasible in (D_{alt}), and $(y_1, \ldots, y_{\ell+1})$ as stated. Then

$$Ax \in K \cap R(A) \subseteq K \cap y_1^{\perp} \cap \ldots \cap y_\ell^{\perp} \Rightarrow \langle Ax, y_{\ell+1} \rangle \geq 0,$$

which yields the contradiction

$$\langle Ax, y_{\ell+1} \rangle = \langle x, A^* y_{\ell+1} \rangle = \langle c, x \rangle = -1.$$

□

We illustrate Theorem I with a semidefinite system, with $Y = \mathcal{S}^n$ the set of order n symmetric matrices and $K = K^* = \mathcal{S}_+^n$ as the set of order n positive semidefinite matrices. The inner product of $a, b \in \mathcal{S}^n$ is $a \bullet b := \langle a, b \rangle := \mathrm{trace}(ab)$ and we write $\preceq$ in place of $\leq_K$. Note that we denote the elements of $\mathcal{S}^n$ by small letters, and reserve capital letters for operators.

Example 3. The semidefinite system

$$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \bullet y \;=\; 0$$
$$\begin{pmatrix} 0 & 1 \\ 1 & \alpha \end{pmatrix} \bullet y \;=\; -1 \tag{3.2.1}$$
$$y \;\succeq\; 0$$

is infeasible for any $\alpha \geq 0$, and weakly infeasible exactly when $\alpha = 0$.

Since the constraint matrices are in $\mathrm{FR}(\mathcal{S}_+^2)$, we see that (3.2.1) is in the form of (D_{ref}) (and itself is a proof of infeasibility).

Suppose $\alpha = 0$ and let

$$y_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \, y_2 = \begin{pmatrix} 0 & -1/2 \\ -1/2 & 0 \end{pmatrix}.$$

Then $(y_1, y_2) \in \mathrm{FR}(\mathcal{S}_+^2)$, $A^*(y_1) = (0,0)^T$, $A^*(y_2) = (0,-1)^T$, so (y_1, y_2) proves that (3.2.1) is not strongly infeasible.

Classically, the set of right hand sides that make (D) weakly infeasible, is the *frontier* of A^*K^* defined as the difference between A^*K^* and its closure. In this case

$$\mathrm{front}(A^*K^*) \;=\; \mathrm{cl}\, A^*\mathcal{S}_+^2 \setminus A^*\mathcal{S}_+^2 \;=\; \{\, (0,\lambda) : \, \lambda \neq 0 \,\}.$$

For all such right hand sides a suitable (y_1, y_2) proves that (3.2.1) is not strongly infeasible.

We organize the rest of the paper as follows. In the rest of the introduction we review prior work, collect notation, and record basic properties of the facial reduction cone $\mathrm{FR}_k(K)$. In Section 3.6 we present our simple facial reduction algorithm, and our exact duals of (P) and (D).

In Section 3.7 we present our exact certificates of infeasibility and weak infeasibility of general conic LPs, and in Section 3.8 we describe the corresponding certificates for SDPs. We also present our geometric corollaries: an exact characterization of when the linear image of a closed convex cone is closed, an exact characterization of *nice cones* (Pataki [29], Roshchina [40]), and bounds on how many constraints can be dropped from, or added to a (weakly) infeasible conic LP, while keeping its feasibility status. Note that when K^* (and K) is polyhedral, and (D) is infeasible, a single equality constraint obtained using eros and membership in K^* proves infeasibility (by Farkas' lemma). In the general case the number of necessary constraints is related to the length of the longest chain of faces in K. In Section 3.9 we define a natural class of weakly infeasible SDPs, and provide a simple algorithm to generate *all* instances in this class. In Section 3.10 we present a library of infeasible, and weakly infeasible SDPs, and our computational results. Section 3.11 concludes.

3.3 Literature Reviews

Facial reduction algorithms – see Borwein and Wolkowicz [11; 10], Waki and Muramatsu [49], and Pataki [30] – achieve exact duality between (P) and (D) by constructing a suitable smaller cone, say F, to replace K in (P), and to replace K^* by F^* (a larger cone) in (D).

Extended strong duals for semidefinite programs and generalizations – see Ramana [35], Klep and Schweighofer [20], Pataki [30] – use polynomially many extra variables and constraints. We note that Ramana's dual relies on convex analysis, while Klep and Schweighofer's uses ideas from algebraic geometry. The approaches of facial reduction and extended duals are related – see Ramana, Tunçel and Wolkowicz [36], and [30] for proofs of the correctness of Ramana's dual relying on facial reduction algorithms. The paper [30] generalizes Ramana's dual to the context of conic LPs over *nice* cones. See Pólik and Terlaky [32] for a generalization of Ramana's dual for conic LPs over homogeneous cones.

For recent studies on the closedness of the linear image of a closed convex cone we refer to Bauschke and Borwein [4]; and Pataki [27]. The paper [4] gives a necessary and sufficient condition for the continuous image of a closed convex cone to be closed, in terms of the *strong conical hull intersection property.* Pataki [27] gives necessary conditions, which subsume well known sufficient conditions, and are necessary and sufficient for a broad class of cones, called *nice cones.*

Borwein and Moors [8; 9] recently showed that the set of linear maps under which the image is *not* closed is σ-porous, i.e., it has Lebesgue measure zero, and is also small in terms of category. For characterizations of nice cones, see Pataki [29]; and Roshchina [40] for a proof that not all facially exposed cones are nice.

Elementary reformulation for SDPs – see Pataki [28] and Liu and Pataki [21] – use simple operations, as elementary row operations, to bring a semidefinite system into a form from which its status (as infeasibility) is trivial to read off. We refer to Lourenco et al [22] for an error-bound based reduction procedure to simplify weakly infeasible SDPs, and a proof that weakly infeasible SDPs contain another such system whose dimension is at most $n-1$. We generalize this result in Theorem 12.

3.4 Notations and preliminaries

We assume throughout that the operator A is surjective. For x and y in the same Euclidean space we sometimes write x^*y for $\langle x, y\rangle$. For a convex set C we denote its linear span, the orthogonal complement of its linear span, its closure, and relative interior by $\operatorname{lin} C,\ C^{\perp},\ \operatorname{cl} C,$ and $\operatorname{ri} C,$ respectively.

We define the dual cone of K as

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

and for convenience we set

$$K^{*\setminus\perp} := K^* \setminus K^{\perp}.$$

We say that strong duality holds between (P) and (D) if their values agree and the latter is attained when finite. This is true when (P) is strictly feasible, i.e., when there is $x \in \mathbb{R}^m$ with $b - Ax \in \mathrm{ri} K$.

For $F,$ a convex subset of K we say that F is a *face* of $K,$ if $y, z \in K,$ and $1/2(y+z) \in F$ implies $y, z \in F$.

Definition 5. If H is an affine subspace with $H \cap K \neq \emptyset,$ then we call the smallest face of K that contains $H \cap K$ the minimal cone of $H \cap K$.

The traditional alternative system of (P) is

$$\begin{array}{rcl} A^* y & = & 0 \\ b^* y & = & -1 \\ y & \geq_{K^*} & 0; \end{array} \tag{$P_{\rm alt}$}$$

if it is feasible, then (P) is infeasible and we say that it is strongly infeasible.

For a nonnegative integer r we denote by $\mathcal{S}_+^r \oplus \{0\}$ the subset of $\mathcal{S}_+^n$ (where n will be clear from the context) with psd upper left r by r block, and the rest zero, and write

$$\mathcal{S}_+^r \oplus \{0\} = \begin{pmatrix} \oplus & 0 \\ 0 & 0 \end{pmatrix}, \ (\mathcal{S}_+^r \oplus \{0\})^* = \begin{pmatrix} \oplus & \times \\ \times & \times \end{pmatrix} \tag{3.4.2}$$

where the $\times$ stand for matrix blocks with arbitrary elements. All faces of $\mathcal{S}_+^n$ are of the form $t^T(\mathcal{S}_+^r \oplus 0)t$ where t is an invertible matrix [3; 26]. We write

$$\mathrm{Aut}(K) \quad = \quad \{\, T : Y \rightarrow Y \,|\, T \,\text{is linear and invertible}, T(K) = K \,\} \tag{3.4.3}$$

for the automorphism group of a closed convex cone K.

Definition 6. We say that $F_1, \ldots, F_k$ faces of K form a *chain of faces,* if $F_1 \supsetneq F_2 \supsetneq \cdots \supsetneq F_k$ and we write ℓ_K for the length of the longest chain of faces in K.

For instance, $\ell_{\mathcal{S}_+^n} = \ell_{\mathbb{R}_+^n} = n + 1$.

3.5 The properties of the facial reduction cone

In Lemma 2 we record relevant properties of $\mathrm{FR}_k(K)$. Its proof is given in Appendix A.

Lemma 2. For $k \geq 0$ the following hold:

(1) $\mathrm{FR}_k(K)$ is a convex cone.

(2) $\mathrm{FR}_k(K)$ is only closed if K is a subspace or $k = 1$.

(3) If $T \in \mathrm{Aut}(K)$, and $(y_1, \dots, y_k) \in \mathrm{FR}_k(K)$ then

$$(Ty_1, \dots, Ty_k) \in \mathrm{FR}_k(K).$$

(4) If C is another closed convex cone, then

$$\mathrm{FR}_k(K \times C) \,=\, \mathrm{FR}_k(K) \times \mathrm{FR}_k(C).$$

Precisely,

$$\big((y_1, z_1), \dots, (y_{k+1}, z_{k+1})\big) \,\in\, FR_k(K \times C) \tag{3.5.4}$$

if and only if

$$\big(y_1, \dots, y_{k+1}\big) \in \mathrm{FR}_k(K) \, \text{and} \, \big(z_1, \dots, z_{k+1}\big) \in \mathrm{FR}_k(C). \tag{3.5.5}$$

Proof of Lemmas 2 and 3

Proof of Lemma 2

Proof of (1) It is trivial that $\mathrm{FR}_k(K)$ contains all nonnegative multiples of its elements, so we only need to show that it is convex. To this end, we use the following Claim, whose proof is an easy exercise:

Claim If C is a closed, convex cone and $y, z \in C^*$, then

$$C \cap (y+z)^\perp = C \cap y^\perp \cap z^\perp.$$

We let $(y_1, \dots, y_k), (z_1, \dots, z_k) \in \mathrm{FR}_k(K)$, and for brevity, for $i = 1, \dots, k$ we set

$$\begin{aligned} K_{y,i} &= K \cap y_1^\perp \cap \cdots \cap y_i^\perp, \\ K_{z,i} &= K \cap z_1^\perp \cap \cdots \cap z_i^\perp, \\ K_{y+z,i} &= K \cap (y_1+z_1)^\perp \cap \cdots \cap (y_i+z_i)^\perp. \end{aligned}$$

We first prove that for $i = 1, \dots, k$ the relation

$$K_{y+z,i} \;=\; K_{y,i} \cap K_{z,i} \text{ holds.} \tag{3.5.6}$$

For $i = 1$ this follows from the Claim. Suppose now that (3.5.6) is true with $i-1$ in place of i. Then

$$y_i \in K_{y,i-1}^* \subseteq (K_{y,i-1} \cap K_{z,i-1})^* = K_{y+z,i-1}^*, \tag{3.5.7}$$

where the first containment is by definition, the inclusion is trivial, and the equality is by using the induction hypothesis. Analogously,

$$z_i \;\in\; K_{y+z,i-1}^*. \tag{3.5.8}$$

Hence

$$\begin{aligned} K_{y+z,i} &= K_{y+z,i-1} \cap (y_i+z_i)^\perp \\ &= K_{y+z,i-1} \cap y_i^\perp \cap z_i^\perp \\ &= K_{y,i-1} \cap K_{z,i-1} \cap y_i^\perp \cap z_i^\perp \\ &= K_{y,i} \cap K_{z,i}, \end{aligned}$$

where the first equation is trivial. The second follows since by (3.5.7) and (3.5.8) we can use the Claim with $C = K_{y+z,i-1}$, $y = y_i$, $z = z_i$. The third is by the inductive hypothesis, and the last is by definition. This completes the proof of (3.5.6).

Now we use (3.5.7), (3.5.8) and the convexity of $K^*_{y+z,i-1}$ to deduce that

$$y_i + z_i \in K^*_{y+z,i-1} \text{ holds for } i = 1, \ldots, k.$$

This completes the proof of (1).

Proof of (2) Let $L = K \cap -K$, assume $K \neq L$, and $k \geq 2$. Let $\{y_{1i}\} \subseteq \text{ri}K^*,$ s.t. $y_{1i} \to 0.$ Then

$$\begin{array}{rcl} K \cap y_{1i}^{\perp} & = & L, \;\Rightarrow\; (K \cap y_{1i}^{\perp})^* \;=\; L^{\perp} \\ K \cap 0^{\perp} & = & K \;\Rightarrow\; (K \cap 0^{\perp})^* \;=\; K^*. \end{array}$$

Let $y_2 \in L^{\perp} \setminus K^*$. (Such a y_2 exists, since $K^* \neq L^{\perp}$.) Then $(y_{i1}, y_2, 0, \ldots, 0) \in \text{FR}_k(K)$, and it converges to $(0, y_2, 0, \ldots, 0) \notin \text{FR}_k(K)$.

Proof of (3) Let us fix $T \in \text{Aut}(K)$ and let S be an arbitary set. Then we claim that

$$(TS)^* = T^{-1}S^*, \tag{3.5.9}$$

$$(TS)^{\perp} = T^{-1}S^{\perp}, \tag{3.5.10}$$

$$(K \cap (TS)^{\perp})^* = T(K \cap S^{\perp})^* \tag{3.5.11}$$

hold. The first two statements are an easy calculation, and the third follows by

$$\begin{aligned} (K \cap (TS)^{\perp})^* &= (K \cap T^{-1}S^{\perp})^* \\ &= (T^{-1}(K \cap S^{\perp}))^* \\ &= T(K \cap S^{\perp})^*, \end{aligned}$$

where in the first equation we used (3.5.10), in the second equation we used $T^{-1}K = K$ and in the last we used (3.5.9).

Now let $(y_1, \dots, y_k) \in \mathrm{FR}_k(K)$, and $S_i = \{y_1, \dots, y_{i-1}\}$ for $i = 1, \dots, k$. Then by definition we have $y_i \in (K \cap S_i^\perp)^*$ and (3.5.11) implies

$$Ty_i \quad \in \quad (K \cap (TS_i)^\perp)^*,$$

which completes the proof.

Proof of (4) The equivalence of (3.5.4) and of (3.5.5) is trivial for $k = 0$ so let us assume that $k \geq 1$ and we proved it for $0, \dots, k-1$. Statement (3.5.4) is equivalent to

$$\big((y_1, z_1), \dots, (y_k, z_k)\big) \in FR_{k-1}(K \times C)$$

and

$$(y_{k+1}, z_{k+1}) \in \big((K \times C) \cap (y_1, z_1)^\perp \cap \dots \cap (y_k, z_k)^\perp\big)^*. \tag{3.5.12}$$

By the inductive hypothesis the set on the right hand side of (3.5.12) is

$$(K \cap y_1^\perp \cap \dots \cap y_k^\perp)^* \times (C \cap z_1^\perp \cap \dots \cap z_k^\perp)^*,$$

and this completes the proof.

3.6 Facial reduction and strong duality for conic linear program

In this section we present a very simple facial reduction algorithm to find $F,$ the minimal cone of the system

$$H \cap K, \tag{3.6.13}$$

where H is an affine subspace with $H \cap K \neq \emptyset$ and our exact duals of (P) and of (D). While simple facial reduction algorithms are available, the convergence proof of Algorithm 1, with an upper bound on the number of steps, is particularly simple.

To start, we note that if F is the minimal cone of (3.6.13) then $H \cap \mathrm{ri} F \neq \emptyset$ (otherwise $H \cap K$ would be contained in a proper face of F). So if F is the minimal cone of $(\mathcal{R}(A)+b) \cap K$, then replacing K by F in (P) makes (P) strictly feasible, and keeps its feasible set the same. Hence if we also replace K^* by F^* in (D) then strong duality holds between (P) and (D).

To illustrate this point, we consider the following example:

Example 4. The optimal value of the SDP

$$\begin{array}{ll} \sup & x_1 \\ s.t. & x_1 \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \preceq \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array} \tag{3.6.14}$$

is zero. Its usual SDP dual, in which we denote the dual matrix by y and its components by y_{ij}, is equivalent to

$$\begin{array}{ll} \inf & y_{11} \\ s.t. & \begin{pmatrix} y_{11} & 1/2 & -y_{22}/2 \\ 1/2 & y_{22} & y_{23} \\ -y_{22}/2 & y_{23} & y_{33} \end{pmatrix} \succeq 0, \end{array} \tag{3.6.15}$$

which does not have a feasible solution with $y_{11} = 0$ (in fact it has an unattained 0 infimum).

Since all slack matrices in (3.6.14) are contained in $\mathcal{S}^1_+ \oplus 0$ and there is a slack matrix whose $(1,1)$ element is positive, the minimal cone of this system is $F = \mathcal{S}^1_+ \oplus 0$. If in the dual program we replace $\mathcal{S}^3_+$ by F^* then the new dual attains with

$$y := \begin{pmatrix} 0 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \in F^* = \begin{pmatrix} \oplus & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{pmatrix} \tag{3.6.16}$$

being an optimal solution (cf. equation (3.4.2)).

To construct the minimal cone we rely on the following classic theorem of the alternative (recall $K^{*\setminus\perp} = K^* \setminus K^\perp$).

$$H \cap \mathrm{ri} K = \emptyset \Leftrightarrow H^\perp \cap K^{*\setminus\perp} \neq \emptyset. \tag{3.6.17}$$

Algorithm 1 repeatedly applies (3.6.17) to find F :

Algorithm 1 Facial Reduction

```
Initialization: Let $y_0 = 0$, $F_0 = K$, $i = 1$.
  while $\exists y_i \in H^\perp \cap F_{i-1}^{*\setminus\perp}$ do
    Choose such a $y_i$.
    Let $F_i = F_{i-1} \cap y_i^\perp$.
    Let $i = i + 1$.
  end while
```

To analyze Algorithm 1 we need a definition:

Definition 7. For $k \geq 1$ we say that $(y_1, \dots, y_k) \in \mathrm{FR}_k(K)$ is *strict*, if

$$y_i \ \in \ (K \cap y_1^\perp \cap \cdots \cap y_{i-1}^\perp)^{*\setminus\perp} \, \text{for} \, i = 1, \dots, k.$$

We say that it is *pre-strict* if $(y_1, \dots, y_{k-1})$ is strict.

If $(y_1, \dots, y_k)$ is strict, then these vectors are linearly independent. Assuming that they are not, for some $1 \leq i \leq k$ a contradiction follows:

$$y_i \in \mathrm{lin}\,\{\, y_1, \dots, y_{i-1} \,\} \ \subseteq \ (K \cap y_1^\perp \cap \cdots \cap y_{i-1}^\perp)^\perp.$$

Recall that ℓ_K is the length of the longest chain of faces in K (Definition 6).

Theorem 3. The following hold:

(1) If $y_1, \dots, y_k$ are found by Algorithm 1, then $(y_1, \dots, y_k) \in \mathrm{FR}_k(K)$ and

$$F \subseteq K \cap y_1^\perp \cap \cdots \cap y_k^\perp. \tag{3.6.18}$$

(2) Equality holds in (3.6.18) for some strict $(y_1, \ldots, y_k) \in \mathrm{FR}_k(K)$ with

$$k \leq \min\{\, \ell_K - 1,\, \dim H^\perp \,\}.$$

Proof To prove part (1) define

$$F_k = K \cap y_1^\perp \cap \cdots \cap y_k^\perp.$$

The statement follows, since F_k is a face of K that contains $H \cap K$ (by $y_i \in H^\perp$), and F is the smallest such face.

We next prove part (2). If we choose $(y_1, y_2, \ldots)$ in Algorithm 1 to be strict, then the algorithm eventually stops. Suppose it stops after finding $y_1, \ldots, y_k$ and for brevity define F_k as above. Then $\mathrm{ri} F_k \cap H \neq \emptyset$, and

$$\mathrm{ri} F_k \cap H \;=\; \mathrm{ri} F_k \cap H \cap K \;=\; \mathrm{ri} F_k \cap H \cap F,$$

so $\mathrm{ri} F_k \cap F \neq \emptyset$. So by Theorem 18.1 in [39] we obtain $F_k \subseteq F$ with the reverse containment already given. The upper bound on k follows from strictness and the linear independence of $y_1, \ldots, y_k$. □

Definition 8. The singularity degree of the system $H \cap K$ which we denote by $d(H, K)$ is the minimum number of facial reduction steps needed to find its minimal cone.

When (P) (resp. (D)) are feasible, we define the minimal cone (degree of singularity) of (P) and of (D) as the minimal cone (degree of singularity) of the systems

$$(\mathcal{R}(A) + b) \cap K, \text{ and } \{\, y \mid A^* y = c \,\} \cap K^*.$$

We write $d(P)$ and $d(D)$ for the singularity degrees.

Example 5. (Example 4 continued) Algorithm 1 applied to this example may output the sequence

$$\begin{aligned} y_1 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \; F_1 = \begin{pmatrix} \bigoplus & & 0 \\ & & 0 \\ 0 & 0 & 0 \end{pmatrix}, \\ y_2 &= \begin{pmatrix} 0 & 0 & -1 \\ 0 & 2 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \; F_2 = F. \end{aligned} \tag{3.6.19}$$

Note that $y_i \in H^{\perp}$ in this context means that y_1 and y_2 are orthogonal to all constraint matrices, and that the singularity degree of (3.6.14) is two.

From Theorem 3 we immediately obtain an extended strong dual for (P), described below in (D_{ext}). Note that (D_{ext}) is an explicit conic linear program whose data is the same as the data of (P), thus it extends the exact SDP duals of Ramana [35] and Klep and Schweighofer [20] to the context of general conic LPs. The underlying cone in (D_{ext}) is the facial reduction cone: thus, somewhat counterintuitively, we find an exact dual of (P) over a convex cone, which is not closed in all important cases.

Theorem 4. Let $k \geq d(P)$. Then the problem

$$\begin{array}{lrcl} \inf & b^* y_{k+1} & & \\ s.t. & A^* y_{k+1} & = & c \\ & A^* y_i & = & 0 \, (i = 1, \ldots, k) \\ & b^* y_i & = & 0 \, (i = 1, \ldots, k) \\ & (y_1, \ldots, y_{k+1}) & \in & \mathrm{FR}_{k+1}(K) \end{array} \tag{D_{ext}}$$

is a strong dual of (P). If $k = d(P)$ and the value of (P) is finite, then (D_{ext}) has an optimal solution which is pre-strict.

Proof We first prove weak duality. Suppose that x is feasible in (P) and $(y_1, \dots, y_{k+1})$ in (D_{ext}) then

$$\begin{aligned} \langle b, y_{k+1} \rangle - \langle c, x \rangle &= \langle b, y_{k+1} \rangle - \langle A^* y_{k+1}, x \rangle \\ &= \langle b - Ax, y_{k+1} \rangle \geq 0, \end{aligned}$$

where the last inequality follows from $b - Ax \in K \cap y_1^\perp \cap \dots \cap y_k^\perp$.

To prove the rest of the statements, first assume that (P) is unbounded. Then by weak duality (D_{ext}) is infeasible. Suppose next that (P) has a finite value v and let F be the minimal cone of (P). Let us choose $y \in F^*$ to satisfy the affine constraints of (D) with $b^* y = v$. We have that

$$F = K \cap y_1^\perp \cap \dots \cap y_k^\perp$$

for some $(y_1, \dots, y_k) \in \text{FR}_k(K)$, with all y_i in $(\mathcal{R}(A) + b)^\perp$ and this sequence can be chosen strict, if $k = d(P)$.

Hence $(y_1, \dots, y_k, 0, \dots, 0, y)$ (where the number of zeros is $k - d(P)$) is feasible in (D_{ext}) with value v. This completes the proof. □

Example 6. (Example 4 continued) If we choose y_1, y_2 as in (3.6.19) and y as in (3.6.16), then (y_1, y_2, y) is an optimal solution to the extended dual of (3.6.14).

Theorem 5. If (D) is feasible then it has a strictly feasible reformulation

$$\begin{array}{llcl} \inf & b^* y & & \\ s.t. & \langle a_i', y \rangle & = & 0 \, (i = 1, \dots, k) \\ & \langle a_i', y \rangle & = & c_i' \, (i = k+1, \dots, m) \\ & y & \in & K^* \cap a_1'^\perp \cap \dots \cap a_k'^\perp, \end{array} \tag{$D_{\text{ref,feas}}$}$$

with $k \geq 0$, $(a_1', \dots, a_k') \in \text{FR}(K^*)$, which can be chosen strict.

Proof Let G be the minimal cone of (D) i.e., we fix $y \in Y$ s.t. $A^*y = c$ and let G be the minimal cone of $(\mathcal{N}(A^*) + y) \cap K^*$. By Theorem 3 there is $k \geq 0$ and a strict $(a_1', \ldots, a_k') \in \mathrm{FR}(K^*)$ such that

$$\begin{array}{rcl} a_i' & \in & \mathcal{R}(A) \cap y^\perp \, (i = 1, \ldots, k), \\ G & = & K^* \cap a_1'^\perp \cap \cdots \cap a_k'^\perp. \end{array}$$

Since $a_1', \ldots, a_k'$ are linearly independent, we can expand them to

$$A' = [a_1', \ldots, a_k', a_{k+1}', \ldots, a_m'], \, \text{a basis of}\, \mathcal{R}(A).$$

Let us write $A' = AZ$ with Z an invertible matrix. Replacing A by A' and c by $Z^T c$ yields the required reformulation, since

$$Z^T c \;=\; Z^T A^* y \;=\; A'^* y,$$

so the first k components of $Z^T c$ are zero. □

We now contrast Theorem 4 with Theorem 5. In the former the minimal cone of (P) is

$$K \cap y_1^\perp \cap \cdots \cap y_k^\perp,$$

where $(y_1, \ldots, y_k, y_{k+1})$ is feasible in (D_{ext}). In the latter the minimal cone of (D) is displayed by simply performing elementary row operations on the constraints. To illustrate Theorem 5, we continue Example 4:

Example 7. (Example 4 continued) We can rewrite the feasible set of this example in an equality constrained form, and choose

$$
\begin{aligned}
\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \bullet y &= 0 \\
\begin{pmatrix} 0 & 0 & -1 \\ 0 & 2 & 0 \\ -1 & 0 & 0 \end{pmatrix} \bullet y &= 0
\end{aligned}
$$

to be among the constraints: these matrices form a sequence in $\mathrm{FR}(\mathcal{S}^3_+)$.

To find the y_i in Algorithm 1 one needs to solve a certain *reducing conic linear program,* and if K is the semidefinite cone, one needs to solve an SDP. This task may not be easier than solving the original problem (P), since the reducing conic LP is strictly feasible, but its dual is not. We know of two approaches to overcome this difficulty. The first approach by Cheung et al [13] is using a modified subproblem whose dual is also strictly feasible. The second, by Permenter and Parrilo in [31] is a "partial" facial reduction algorithm, where they solve linear programming approximations of the SDP subproblems.

3.7 Certificates of infeasibility and weak infeasibility in conic LPs

We now describe a collection of certificates of infeasibility and weak infeasibility of (P) and of (D) below in Theorem 6, which contains Theorem I. The idea is simple: the exact dual of (P) provides an exact certificate of infeasibility of (P) by homogenization, and the remaining certificates are found by using duality and elementary linear algebra.

Theorem 6. When K is a general closed, convex cone, the following hold:

(1) (D) is infeasible, if and only if it has a reformulation

$$\begin{array}{rcl} \langle a_i', y \rangle & = & 0 \, (i = 1, \dots, k) \\ \langle a_{k+1}', y \rangle & = & -1 \\ \langle a_i', y \rangle & = & c_i' \, (i = k+2, \dots, m) \\ y & \geq_{K^*} & 0 \end{array} \qquad (D_{\text{ref}})$$

where $(a_1', \dots, a_{k+1}') \in \text{FR}(K^*)$.

(2) (D) is not strongly infeasible, if and only if there is $(y_1, \dots, y_{\ell+1}) \in \text{FR}(K)$, such that

$$\begin{array}{rcl} A^* y_i & = & 0 \, (i = 1, \dots, \ell) \\ A^* y_{\ell+1} & = & c. \end{array}$$

(3) (P) is infeasible, if and only if there is $(y_1, \dots, y_{k+1}) \in \text{FR}(K)$ such that

$$\begin{array}{rclrcl} A^* y_i & = & 0, & b^* y_i & = & 0 \; (i = 1, \dots, k) \\ A^* y_{k+1} & = & 0, & b^* y_{k+1} & = & -1 \end{array}$$

(4) (P) is not strongly infeasible, if and only if it has a reformulation

$$\sum_{i=1}^{m} x_i a_i' \leq_K b' \qquad (P_{\text{ref}})$$

where $(a_1', \dots, a_\ell', b') \in \text{FR}(K^*)$ for some $\ell \geq 0$.

In all parts the facial reduction sequences can be chosen to be pre-strict.

□

We note that parts (1) through (4) in Theorem 6 should be read separately: the k integers in parts (1) and (3), the ℓ in parts (2) and (4), etc. may be different. We use the current notation for brevity. Also note that since K is a general closed, convex cone, in the reformulations we only use elementary row operations.

If $k = 0$ in part (3) then (P) is strongly infeasible; and if $\ell = 0$ in part (4) then (P) is actually feasible. The reader can check that the "if" directions are all trivial.

Part (3) in Theorem 6 essentially follows from [49], though their infeasbility certificate is not stated as a conic linear system.

Note that Part (1) Theorem 6 allows us to generate all infeasible conic LP instances over cones, whose facial structure (and hence their facial reduction cone) is well understood: to do so, we only need to generate systems of the form (D_{ref}) and reformulate them. By Part (4) we can systematically generate all systems that are not strongly infeasible, though this seems less interesting.

Example 3 already illustrates parts (1) and (2). A larger example, which also depicts the frontier of A^*K^* with $K = K^* = \mathcal{S}^3_+$ follows.

Example 8. Let

$$a_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \, a_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \, a_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Then one easily checks

$$\begin{aligned} \mathrm{cl}(A^*\mathcal{S}^3_+) &= \mathbb{R}_+ \times \mathbb{R} \times \mathbb{R}_+ \\ \mathrm{front}(A^*\mathcal{S}^3_+) \;=\; \mathrm{cl}(A^*\mathcal{S}^3_+) \setminus A^*\mathcal{S}^3_+ &= \{(0, \lambda, \mu) \,|\, \lambda \neq \mu \geq 0\,\}. \end{aligned}$$

The set $A^*\mathcal{S}^3_+$ is shown on Figure 3.1 in blue, and its frontier in green. Note that the blue diagonal segment inside the green frontier actually belongs to $A^*\mathcal{S}^3_+$.

To see how parts (1) and (2) of Theorem 6 certify that elements of front$(A^*\mathcal{S}^3_+)$ are indeed in this set, for concreteness, consider the system

$$\begin{array}{rcl} A^*(y) & = & (0,1,2)^T \\ y & \succeq & 0. \end{array} \tag{3.7.20}$$

The operations: 1) multiply the second equation by 3 and 2) subtract twice the third equation from it, bring (3.7.20) into the form of (D_{ref}) and show that it is infeasible. The y_1 and y_2 below prove that it is not strongly infeasible:

$$y_1 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix},\, y_2 = \begin{pmatrix} 0 & 0 & -1/2 \\ 0 & 2 & 0 \\ -1/2 & 0 & 0 \end{pmatrix}. \tag{3.7.21}$$

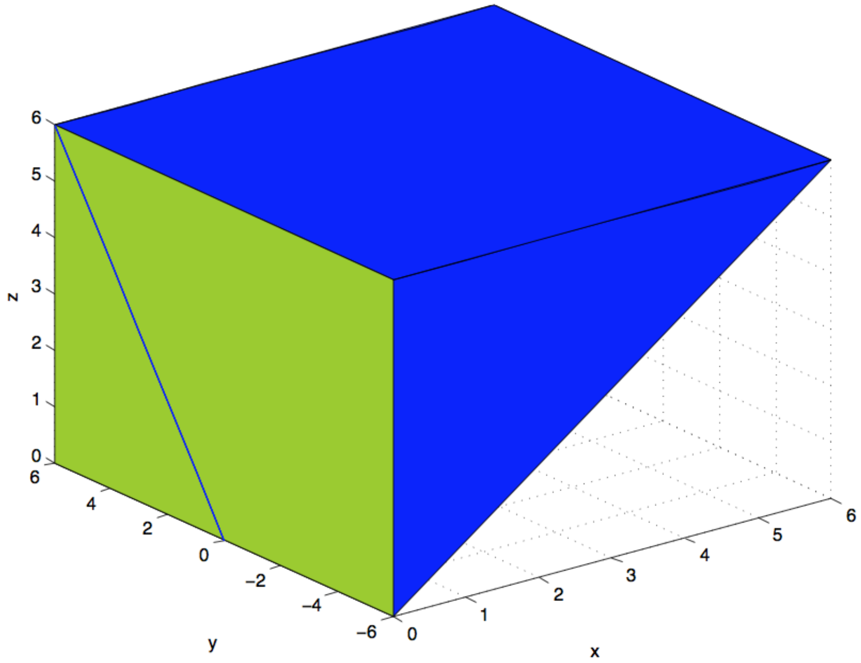


Figure 3.1: The set $A^*\mathcal{S}^3_+$ is in blue, and its frontier is in green

To illustrate parts (3) and (4) in Theorem 6, we modify Example 4 by simply exchanging two constraint matrices.

Example 9. (Example 4 continued) The semidefinite system below is weakly infeasible.

$$x_1 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + x_2 \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \preceq \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}. \tag{3.7.22}$$

To prove it is infeasible, we use part (3) of Theorem 6 with (y_1, y_2, y_3), where y_1, y_2 are given in (3.6.19) and

$$y_3 = \begin{pmatrix} 0 & 0 & -1/2 \\ 0 & 0 & 0 \\ -1/2 & 0 & 0 \end{pmatrix},$$

To prove it is not strongly infeasible, we use part (4). We write a_1, a_2, and b for the constraint matrices, and observe that $(a_1, b) \in \mathrm{FR}_2(\mathcal{S}^3_+)$, and is pre-strict, and $(a_1, a_2, b) \in \mathrm{FR}_3(\mathcal{S}^3_+)$.

We now prove Theorem 6. To make the proofs concise, we prove the statements out of order.

Proof of (3) : Since the conic LP

$$\sup\{x_0 : Ax - bx_0 \leq_K 0\} \tag{3.7.23}$$

has value 0 iff (P) is infeasible, our claim follows from considering the strong dual of (3.7.23) from Theorem 4.

Proof of only if in (1) : Fix $y \in Y$ such that $A^* y = c$. By part (3) there is $k \geq 0$ and a pre-strict $(a'_1, \dots, a'_k, a'_{k+1}) \in \mathrm{FR}(K^*)$ such that

$$\begin{array}{rcl} a'_i & \in & \mathcal{R}(A) \cap y^{\perp} \, (i = 1, \dots, k), \\ a'_{k+1} & \in & \mathcal{R}(A), \, \langle a'_{k+1}, y \rangle = -1. \end{array}$$

Since $(a'_1, \dots, a'_k, a'_{k+1})$ is pre-strict, $a'_1, \dots, a'_k$ are linearly independent. Since $\langle a'_{k+1}, y \rangle \neq 0$, also $a'_1, \dots, a'_k, a'_{k+1}$ are linearly independent. The proof now can be completed verbatim as the proof of Theorem 5.

Proof of only if in (2) Since (D) is not strongly infeasible, the alternative system (D_{alt}) is infeasible. By Lemma 4 we deduce that

$$\mathrm{FR}_k(K \times \{0\}) = \mathrm{FR}_k(K) \times \mathbb{R}^{k+1} \, \text{holds for all} \, k \geq 0.$$

Combining this with part (3), there is a pre-strict $(y_1, \dots, y_{k+1}) \in \mathrm{FR}(K)$ and $(z_1, \dots, z_{k+1}) \in \mathbb{R}^{k+1}$ s.t.

$$\begin{array}{rclrcl} A^* y_i + c^* z_i & = & 0, & z_i & = & 0 \, (i = 1, \dots, k) \\ A^* y_{k+1} + c^* z_{k+1} & = & 0, & z_{k+1} & = & -1, \end{array}$$

so our claim follows.

Proof of (4) Since (P) is not strongly infeasible, the system (P_{alt}) is infeasible, hence by part (1) it has a reformulation

$$\begin{array}{rcl} \langle a'_i, y \rangle & = & 0 \, (i = 1, \dots, \ell) \\ \langle b', y \rangle & = & -1 \\ \langle a'_i, y \rangle & = & c'_i \, (i = \ell + 1, \dots, m) \\ y & \geq_{K^*} & 0 \end{array} \tag{3.7.24}$$

with $(a_1', \dots, a_\ell', b') \in \mathrm{FR}(K^*)$ for some $\ell \geq 0$.

Since in (P_{alt}) the only constraint with a nonzero right hand side is $\langle b, y \rangle = -1$, we must have $b' = b + A\mu$ for some $\mu \in \mathbb{R}^m$. Since (3.7.24) is the alternative system of (P_{ref}), the latter cannot be strongly infeasible. This completes the proof. □

3.8 Certificates of infeasibility and weak infeasibility for SDPs

In this section we specialize the certificates of infeasibility and weak infeasibility of Section 3.7 to semidefinite programming. For this purpose we first introduce regularized facial reduction sequences in $\mathcal{S}^n_+$. These sequences have a certain staircase like structure and we will use them in Theorem 7, which is essentially obtained from Theorem 6 by replacing facial reduction sequences by regularized ones.

Definition 9. The set of order k regularized facial reduction sequences for $\mathcal{S}^n_+$ is

$$\mathrm{REGFR}_k(\mathcal{S}^n_+) \;=\; \Bigg\{ (y_1, \dots, y_k) : \, y_i \;=\; \begin{array}{c} \begin{array}{ccc} p_1 + \ldots + p_{i-1} & p_i & n - \sum_{j=1}^{i} p_j \end{array} \\ \left(\begin{array}{ccc} \times & \times & \times \\ \times & I & 0 \\ \times & 0 & 0 \end{array} \right) \end{array} \quad \text{where}\, p_i \geq 0,\, i = 1, \dots, k \Bigg\},$$

where the × symbols correspond to blocks with arbitrary elements. We drop the index k if its value is clear from the context. We say that $(y_1, \dots, y_k)$ has block sizes $p_1, \dots, p_k$ if the order of the identity block in y_i is p_i for all i.

Note that the constraint matrices in most examples actually form regularized facial reduction sequences. Clearly,

$$\mathrm{REGFR}(\mathcal{S}^n_+) \subseteq \mathrm{FR}(\mathcal{S}^n_+)$$

holds, and a sequence $(y_1, \ldots, y_k) \in \mathrm{REGFR}(\mathcal{S}^n_+)$ with block sizes $p_1, \ldots, p_k$ is strict, iff $p_1, \ldots, p_k$ are positive.

The set $\mathrm{REGFR}_k(\mathcal{S}^n_+)$ is not convex. However, Lemma 3 below shows that any element of $\mathrm{FR}_k(\mathcal{S}^n_+)$ can be rotated to reside in $\mathrm{REGFR}_k(\mathcal{S}^n_+)$, i.e., $\mathrm{FR}_k(\mathcal{S}^n_+)$ is the *orbit* of $\mathrm{REGFR}_k(\mathcal{S}^n_+)$ under the automorphism group of $\mathcal{S}^n_+$. The proof of Lemma 3 is given in Appendix A.

Lemma 3. Let $(y_1, \ldots, y_k) \in \mathrm{FR}(\mathcal{S}^n_+)$. Then there is an invertible matrix t such that

$$(t^T y_1 t, \ldots, t^T y_k t) \in \mathrm{REGFR}(\mathcal{S}^n_+).$$

□

The main result of this section follows.

Theorem 7. When $K = \mathcal{S}^n_+$, the following hold:

(1) (D) is infeasible, if and only if it has a reformulation

$$\begin{array}{rcl} \langle a_i', y \rangle & = & 0 \, (i = 1, \ldots, k) \\ \langle a_{k+1}', y \rangle & = & -1 \\ \langle a_i', y \rangle & = & c_i' \, (i = k+2, \ldots, m) \\ y & \succeq & 0, \end{array} \qquad (D_{\mathrm{ref,sdp}})$$

where $(a_1', \ldots, a_{k+1}') \in \mathrm{REGFR}(\mathcal{S}^n_+)$.

(2) (D) is not strongly infeasible, if and only if it has a reformulation with data (A'', c'') and $(y_1, \ldots, y_{\ell+1}) \in \mathrm{REGFR}(\mathcal{S}^n_+)$ such that

$$\begin{array}{rcl} A''^* y_i & = & 0 \, (i = 1, \ldots, \ell) \\ A''^* y_{\ell+1} & = & c''. \end{array}$$

(3) (P) is infeasible, if and only if it has a reformulation with data (A', b') and $(y_1, \dots, y_{k+1}) \in \mathrm{REGFR}(\mathcal{S}^n_+)$ such that

$$\begin{array}{rclcrcl} A'^* y_i & = & 0, & & b'^* y_i & = & 0 \ (i = 1, \dots, k) \\ A'^* y_{k+1} & = & 0, & & b'^* y_{k+1} & = & -1 \end{array}$$

(4) (P) is not strongly infeasible, if and only if it has a reformulation

$$\sum_{i=1}^{m} x_i a_i'' \preceq b'' \qquad (P_{\mathrm{ref,sdp}})$$

where $(a_1'', \dots, a_\ell'', b'') \in \mathrm{REGFR}(\mathcal{S}^n_+)$ for some $\ell \geq 0$.

In all parts the facial reduction sequences can be chosen as pre-strict. □

We first note that parts (1) and (2) above should be read together, but separately from parts (3) and (4), and vice versa. (So the k integers in parts (1) and in part (3) may be different, and so on.) We use the double primes to emphasize that the reformulations in the first two and the last two parts are different.

Example 3 illustrates part (1) in Theorem 7, since the constraint matrices are in $\mathrm{REGFR}(\mathcal{S}^2_+)$. It also illustrates part (2), after we apply a trivial rotation on the constraint matrices:

Example 10. (Example 3 continued) After exchanging the first row and column in this example, and assuming $\alpha = 0$ we obtain the system

$$\begin{array}{rcl} \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \bullet y & = & 0 \\ \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \bullet y & = & -1 \\ y & \succeq & 0. \end{array} \qquad (3.8.25)$$

The fact that (3.8.25) is not strongly infeasible is proved by

$$y_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix},\ y_2 = \begin{pmatrix} 0 & -1/2 \\ -1/2 & 0 \end{pmatrix},$$

and $(y_1, y_2) \in \mathrm{REGFR}(\mathcal{S}^2_+)$.

Example 8 also illustrates part (2) in Theorem 7, since after a trivial rotation of the y_j in (3.7.21) they are in $\mathrm{REGFR}(\mathcal{S}^3_+)$. Example 9 illustrates parts (3) and (4).

Proof of Theorem 7 To see part (1) we consider the reformulation given in part (1) of Theorem 6, and a t invertible matrix such that $(t^T a'_1 t, \ldots, t^T a'_{k+1} t) \in \mathrm{REGFR}(\mathcal{S}^n_+)$. We replace a'_i by $t^T a'_i t$ for all i and obtain $(D_{\mathrm{ref,sdp}})$.

To see (2) we consider the sequence $(y_1, \ldots, y_{\ell+1}) \in \mathrm{FR}(\mathcal{S}^n_+)$ given by part (2) of Theorem 6, and a t invertible matrix such that

$$(t^T y_1 t, \ldots, t^T y_{\ell+1} t) \in \mathrm{REGFR}(\mathcal{S}^n_+).$$

For $i = 1, \ldots, m$ and $j = 1, \ldots, \ell + 1$ we have

$$\langle t^{-1} a_i t^{-T}, t^T y_j t \rangle \quad = \quad \langle a_i, y_j \rangle. \tag{3.8.26}$$

We set $a''_i := t^{-1} a_i t^{-T}$ and replace y_j by $t^T y_j t$ for all i and j and this completes the proof.

The proof of (3) is analogous to the proof of (2); and the proof of (4) to the proof of (1), hence we omit these. □

Note that part (1) in Theorem 7 recovers Theorem 1 in [21]. Parts (2) and (4) are related to the recent paper of Lourenco et al [22]. The authors there show that if a semidefinite system of the form (P) is weakly infeasible, then a sequence $(a'_1, \ldots, a'_\ell) \in \mathrm{REGFR}_\ell(\mathcal{S}^n_+)$ can be found by taking linear combinations of the a_i and applying rotations.

In contrast, we exactly characterize systems that are *infeasible* and systems that are *not strongly infeasible.* Putting these parts together yields our algorithm to generate weakly infeasible SDPs (in Section 3.9).

3.9 Generating infeasible, and weakly infeasible SDPs

We now turn to a practical aspect of our work, generating infeasible, and weakly infeasible SDP instances. Having a library of such instances is important, since detecting infeasibility is a weak point of commercial and research codes: when they report this status, they also return a feasible solution to the alternative system (D_{alt}). When the instance is weakly infeasible, the returned certificate is necessarily inaccurate.

We first state an elementary algorithm, based on part (1) of Theorem 7, to generate infeasible SDPs.

Algorithm 2 Infeasible SDP

```
1: Choose integers $m, n, k, p_1, \dots, p_k > 0$ and $p_{k+1} \geq 0$ s.t. $k+1 \leq m$, $\sum_{i=1}^{k+1} p_i \leq n$.
2: Let $(a_1, \dots, a_{k+1}) \in \text{REGFR}_{k+1}(\mathcal{S}^n_+)$ with block sizes $p_1, \dots, p_{k+1}$ and $c_1 = \cdots = c_k = 0$, $c_{k+1} = -1$.
3: Let $a_{k+2}, \dots, a_m \in \mathcal{S}^n$ and $c_{k+2}, \dots, c_m \in \mathbb{R}$ be arbitrary.
```

By Theorem 7 all infeasible SDPs are a reformulation of a possible output of Algorithm 2. This algorithm may generate a strongly or a weakly infeasible SDP, and the latter outcome is likelier if k is large with respect to $m,$ but weak infeasibility is not guaranteed.

Next we turn to generating weakly infeasible SDP instances with a proof of weak infeasibility. We first note that Waki in [47] described a method to generate such SDPs from Lasserre's relaxation of polynomial optimization problems. His instances turned out to be very useful in computational testing of SDP solvers. In contrast, we will generate our instances by solving simple systems of equations. In fact, we will define a natural class of weakly infeasible SDPs, and show that a simple algorithm generates *all* instances in this class.

Although our framework is different – since we generate objects in an uncountably infinite set – our algorithms to generate *all* SDP instances in a certain class fit into the framework of *listing* combinatorial objects, as cycles, paths, spanning trees and cuts: see e.g., [37; 34].

We will use part (1) of Theorem 6 to find an infeasible instance, and part (2) to find a (y_j) sequence to prove that it is not strongly infeasible, so we will solve a *bilinear* system of equations over the (a_i) and (y_j). While this may be difficult in general, it is easy if we impose a structure: we will require that the (a_i) be regularized (cf. Definition 9), and that the (y_j) have the same structure, but "reversed" in the sense defined below:

Definition 10. The set of order ℓ reversed regularized facial reduction sequences in $\mathcal{S}^n_+$ is

$$\mathrm{REVREGFR}_\ell(\mathcal{S}^n_+) \quad = \quad \Bigg\{(y_1,\dots,y_\ell) : \, y_i \; = \; \begin{array}{c} \begin{array}{ccc} n-\sum_{j=1}^{i} q_j & q_i & \sum_{j=1}^{i-1} q_j \end{array} \\ \left(\begin{array}{ccc} 0 & 0 & \times \\ 0 & I & \times \\ \times & \times & \times \end{array}\right) \end{array}$$
$$\text{where}\, q_i \geq 0, \, i=1,\dots,\ell\Bigg\},$$

where the × symbols correspond to blocks with arbitrary elements. We drop the subscript, if its value is clear from the context.

For instance, the y_i matrices in Example 3 are in $\mathrm{REVREGFR}(\mathcal{S}^2_+)$.

Definition 11. An SDP instance

$$\begin{array}{rcl} A^*y & = & c \\ y & \succeq & 0 \end{array} \tag{3.9.27}$$

is *nonoverlapping weakly infeasible*, if

(1) it is in the form (D_{ref}) as in part (1) of Theorem 6 with $(a_1,\dots,a_{k+1}) \in \mathrm{REGFR}(\mathcal{S}^n_+)$ for some $k \geq 1$.

(2) There is $(y_1, \ldots, y_{\ell+1}) \in \mathrm{REVREGFR}(\mathcal{S}_+^n)$ as in part (2) of Theorem 6 which proves it is not strongly infeasible;

(3) The block sizes p_i of $(a_1, \ldots, a_{k+1})$ and the block sizes q_j of $(y_1, \ldots, y_{\ell+1})$ satisfy

$$\sum_{i=1}^{k+1} p_i + \sum_{j=1}^{\ell+1} q_j \leq n. \tag{3.9.28}$$

Note that condition (3.9.28) means that the identity blocks in the (a_i) and (y_j) sequences do not overlap. Example 3 is such an instance with $p_1 = q_1 = 1$ and $p_2 = q_2 = 0$.

A larger example follows:

Example 11. The SDP

$$\begin{array}{rcl} a_i \bullet y & = & 0 \, (i = 1, 2), \\ a_3 \bullet y & = & -1, \\ y & \succeq & 0 \end{array} \tag{3.9.29}$$

is weakly infeasible, where a_1, a_2, a_3 are given below:

$$a_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \, a_2 = \begin{pmatrix} 5 & 1 & 2 & \underline{2} & \underline{0} \\ 1 & 1 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 \\ \underline{2} & 0 & 0 & 0 & 0 \\ \underline{0} & 0 & 0 & 0 & 0 \end{pmatrix}, \, a_3 = \begin{pmatrix} 3 & 2 & 1 & 3 & -2 \\ 2 & 0 & 0 & \underline{0.5} & \underline{1} \\ 1 & 0 & 1 & 0 & 0 \\ 3 & \underline{0.5} & 0 & 0 & 0 \\ -2 & \underline{1} & 0 & 0 & 0 \end{pmatrix}.$$

(Some matrix entries are underlined, since we will return to this instance to explain our algorithm.)

Matrices y_1, y_2, y_3 (again with some underlined entries) below show that (3.9.29) is nonoverlapping weakly infeasible:

$$y_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \; y_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & \underline{1} \\ 0 & 0 & 0 & 0 & \underline{2} \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ \underline{1} & \underline{2} & 1 & 0 & 0 \end{pmatrix}, \; y_3 = \begin{pmatrix} 0 & 0 & 0 & \underline{0} & 3 \\ 0 & 0 & 0 & \underline{1} & 5 \\ 0 & 0 & 0 & 4 & 1 \\ \underline{0} & \underline{1} & 4 & 1 & 2 \\ 3 & 5 & 1 & 2 & 3 \end{pmatrix}.$$

Note that here

$$k = \ell = 2, \, p_1 = p_2 = p_3 = 1, q_1 = q_2 = 1, \, q_3 = 0.$$

It is of course easy to see directly that (3.9.29) is weakly infeasible. However, if we generate matrices $a_i \, (i = 4, 5, \dots,)$ orthogonal to y_1 and y_2 and add the contraints $a_i \bullet y = a_i \bullet y_3$ to (3.9.29), the resulting system is still weakly infeasible, but this would be difficult to confirm directly. (A simple dimension count shows that this way we can extend (3.9.29) to have 13 constraints.)

To proceed with stating our algorithm, for $(a_1, \dots, a_{k+1}) \in \text{REGFR}(\mathcal{S}^n_+)$ with block sizes $p_1, \dots, p_{k+1}$ we denote the ith block containing p_i integers by P_i, i.e.,

$$P_1 = \big\{1, \dots, p_1\big\}, \, P_2 = \{p_1 + 1, \dots, p_1 + p_2\}, \dots$$

For $(y_1, \dots, y_{\ell+1}) \in \text{REVREGFR}(\mathcal{S}^n_+)$ with block sizes $q_1, \dots, q_{\ell+1}$ we similarly denote the jth block containing q_j integers by

$$Q_1 = \big\{n - q_1 + 1, \dots, n\big\}, \, Q_2 = \big\{n - q_1 - q_2 + 1, \dots, n - q_1\big\}, \dots$$

For instance, in Example 11

$$P_1 = \{1\}, \, P_2 = \{2\}, \, P_3 = \{3\}, \, Q_1 = \{5\}, \, Q_2 = \{4\}, \, Q_3 = \emptyset. \tag{3.9.30}$$

For $a \in \mathcal{S}^n$ and $P, Q \subseteq \{1, \dots, n\}$ we denote by $a(P, Q)$ the union of the block of a indexed by rows corresponding to P and columns corresponding to Q; and the block symmetric with it (i.e., rows indexed by Q and columns indexed by P.).

We are now ready to state our algorithm. The input of Algorithm 3 is $(a_1, \dots, a_{k+1}) \in \text{REGFR}(\mathcal{S}^n_+)$ and $(y_1, \dots, y_{\ell+1}) \in \text{REVREGFR}(\mathcal{S}^n_+)$ with block sizes p_i and q_j which satisfy inequality (3.9.28). We fix all entries of all a_i and of all y_j in advance, except we leave free the entries in

$$a_i(P_{i-1}, Q_1 \cup \dots \cup Q_\ell) \,\text{and}\, y_j(Q_{j-1}, P_1 \cup \dots \cup P_k). \tag{3.9.31}$$

The algorithm then sets the entries in these free blocks to satisfy the equations

$$a_i \bullet y_j = \begin{cases} 0 & \text{if } (i,j) \neq (k+1, \ell+1), \\ -1 & \text{if } (i,j) = (k+1, \ell+1). \end{cases} \tag{3.9.32}$$

This way we find the first $k+1$ equations in (D) and the last part of the algorithm generates the remaining $m-k-1$.

Algorithm 3 Nonoverlapping weakly infeasible SDP

```
for $j = 2 : (\ell + 1)$ do
    for $i = 2 : (k + 1)$ do
        (*) Set $a_i(P_{i-1}, Q_{j-1})$ and $y_j(P_{i-1}, Q_{j-1})$ to satisfy the equation for $a_i \bullet y_j$.
    end for
end for
Find $a_{k+2}, \dots, a_m$ orthogonal to $y_1, \dots, y_\ell$.
Set $c = (0, \dots, 0, -1, a_{k+2} \bullet y_{\ell+1}, \dots, a_m \bullet y_{\ell+1})^T$.
```

Algorithm 3 can generate Example 3 by starting with only the offdiagonal element of a_2 and y_2 free, then setting these to satisfy the equation $a_2 \bullet y_2 = -1$.

Algorithm 3 can also generate Example 11. It starts with the underlined entries free, and successively sets the entries in the following submatrices (note the definition of P_i and Q_j in (3.9.30))

(1) $a_2(P_1, Q_1)$ and $y_2(P_1, Q_1)$

(2) $a_3(P_2, Q_1)$ and $y_2(P_2, Q_1)$

(3) $a_2(P_1, Q_2)$ and $y_3(P_1, Q_2)$

(4) $a_3(P_2, Q_2)$ and $y_3(P_2, Q_2)$

Theorem 8. Algorithm 3 always succeeds, and every nonoverlapping weakly infeasible instance is among its possible outputs.

Proof To show that the algorithm always succeeds assume that at some point we execute Step (*). All previously satisfied equations which involve a_i have left hand side

$$a_i \bullet y_t \,\text{with}\, t \leq j-1.$$

Since for all such t we have

$$y_t(P_{i-1}, Q_{j-1}) = 0, \;\text{since}\, P_{i-1} \subseteq \{1, \ldots, n\} \setminus (Q_1 \cup \cdots \cup Q_{\ell+1}),$$

all these equations remain satisfied. Similarly, all previously satisfied equations that involve y_j remain true.

It is trivial to prove that all nonoverlapping instances are among the outputs: suppose that such an instance is identified by $(\bar{a}_1, \ldots, \bar{a}_m)$ and $(\bar{y}_1, \ldots, \bar{y}_{\ell+1})$ with $(\bar{a}_1, \ldots, \bar{a}_{k+1})$ having block sizes $p_1, \ldots, p_{k+1}$, and the $\bar{y}_j$ having block sizes q_j. Suppose that before we start Algorithm 3 we set all entries in $(a_1, \ldots, a_{k+1})$ and $(y_1, \ldots, y_{\ell+1})$ other than the ones in (3.9.31) to the corresponding values in the $(\bar{a}_i)$ and $(\bar{y}_j)$. Then there is a possible run of the algorithm which completes the a_i and y_j to be equal to the $\bar{a}_i$ and $\bar{y}_j$. $\square$

3.10 Computational experiments

To generate a test suite of challenging infeasible and weakly infeasible SDPs (in the dual form (D)) we implemented Algorithms 2 and 3 in Matlab. We ran Algorithm 2 with parameters

$$n = 10,\ k = 2,\ p_1 = 2, p_2 = 3, p_3 = 2,\ m = 10 \,\text{or}\, m = 20, \tag{3.10.33}$$

and we call its outputs *infeasible* instances (these may be strongly or weakly infeasible). All entries in the generated instances are integers.

We ran Algorithm 2 with parameters

$$n = 10,\ k = 2,\ \ell = 1,\ r = (2, 3, 2),\ s = (2, 1),\ m = 10 \,\text{or}\, m = 20, \tag{3.10.34}$$

and we call the instances it generates *weakly infeasible.* (These are guaranteed to be weakly infeasible.) We chose the components of the $a_1, \ldots, a_{k+1}$ in the support of the y_j as integers in $[-2, 2]$ so the entries of $a_{k+2}, \ldots, a_m$ and of y_1, y_2 turn out to be "near" integers with components in $\{0, \pm 1, \pm 1/2, \ldots, \pm 1/7\}$.

Hence one can easily verify the status of our instances in exact arithmetic.

To generate instances, in which the structure proving (weak) infeasibility is less readily apparent, we add the optional

Messing step: Choose $t = (t_{ij}) \in \mathbb{Z}^{m \times m}$ and $v = (v_{ij}) \in \mathbb{Z}^{n \times n}$ random invertible matrices with entries in $[-2, 2]$ and let

$$a_i \ = \ v^T \big(\sum_{j=1}^{m} t_{ij} a_j \big) v \text{ for } i = 1, \ldots, m.$$

The t matrix encodes elementary row operations performed on (D), and v encodes a rotation.

We call the instances output by Algorithms 2 and 3 *clean*, and the instances we find after the Messing step *messy.*

The choices: "clean/messy, infeasible/weakly infeasible, $m = 10/m = 20$" provide eight categories and we generated 100 instances in each. We set the objective function as I to ensure that the primal problem (P) is feasible.

We tested four solvers: we first ran the solvers Sedumi, SDPT3 and MOSEK from the YALMIP environment, and the preprocessing algorithm of Permenter and Parrilo [31] interfaced with Sedumi. The latter is marked by "PP+SEDUMI" in our tables.

As the solvers consider our dual problem to be the primal, the only correct solution status is "primal infeasible." In Tables 3.1 and 3.2 we report the number of solved instances out of 100 for the various solvers.

	Infeasible		Weakly Infeasible	
	Clean	Messy	Clean	Messy
SEDUMI	87	27	0	0
SDPT3	10	5	0	0
MOSEK	63	17	0	0
PP+SEDUMI	100	27	100	0

Table 3.1: Results with $n = 10, m = 10$

	Infeasible		Weakly Infeasible	
	Clean	Messy	Clean	Messy
SEDUMI	100	100	1	0
SDPT3	100	96	0	0
MOSEK	100	100	11	0
PP+SEDUMI	100	100	0	0

Table 3.2: Results with $n = 10, m = 20$

We can see that

(1) The standalone solvers do better when m goes from 10 to 20 as for larger m the portion of strongly infeasible instances is likely to be higher.

(2) The standalone solvers mostly fail on the weakly infeasible problems, though MOSEK detects infeasibility of some. These are "almost" strongly infeasible, i.e., the alterna-

tive system (D_{alt}) is almost feasible. (Of course, in exact arithmetic (D_{alt}) is infeasible.)

(3) The preprocessing of [31] considerably helps Sedumi when $m = 10$; and it is the only method to work consistently well on the weakly infeasible instances with $m = 10$. Somewhat surprisingly, it does not work, however, on the clean infeasible instances with $m = 20$.

Clearly, a preprocessing algorithm like [31] could easily scan for entire facial reduction sequences in the input, and it is likely that some of the instances coming from applications also contain such sequences.

The SDP instances are available from

```
www.unc.edu/~pataki/SDP.zip
```

3.11 Discussion and Conclusion

Here we briefly discuss how some of our results can be further extended. First we note that by Theorem 6 and part 4 in Lemma 2 we can write exact duals, and exact certificates of infeasibility for more involved conic linear systems. For instance, the system

$$\begin{array}{rcl} A_1 x & \leq_{K_1} & b_1, \\ A_2 x & \leq_{K_2} & b_2 \end{array} \tag{3.11.35}$$

(where K_1 and K_2 are closed convex cones) is infeasible iff there is $k \geq 0$ and $(y_1, \dots, y_{k+1}) \in \text{FR}_{k+1}(K_1^*)$ and $(z_1, \dots, z_{k+1}) \in \text{FR}_{k+1}(K_2^*)$ with

$$\begin{array}{rclrcl} A_1^* y_i + A_2^* z_i & = & 0, & b_1^* y_i + b_2^* z_i & = & 0 \ (i = 1, \dots, k) \\ A_1^* y_{k+1} + A_2^* z_{k+1} & = & 0, & b_1^* y_{k+1} + b_2^* z_{k+1} & = & -1, \end{array}$$

and at least one of the y_j and z_j sequences can be chosen pre-strict.

Further, if the cone K is well-described – which is the case for all the cones over which one can efficiently optimize, as $\mathcal{S}_+^n$, polyhedral and p-order cones – then so is $\mathrm{FR}_k(K)$. We will say that K is a *smooth cone* if it is pointed, full-dimensional, and all faces distinct from $\{0\}$ and K itself are one-dimensional (i.e., extreme rays). For instance, the p-order cone

$$\{\, (x_0, x) \,|\, x_0 \geq \|\, x \,\|_p \,\}$$

is a smooth cone, when p is not equal to 1 or ∞. The facial reduction cone $\mathrm{FR}(K)$ is trivial for such cones. Hence $\mathrm{FR}_k(K)$ is also trivial if K is a direct product of such cones, using part (4) of Lemma 2.

Therefore, using part (1) of Theorem 6 we can easily generate all infeasible conic LP instances over direct products of smooth cones.

CHAPTER 4

Results in convex analysis

In this chapter we use the preceding results to address several fundamental questions in convex analysis.

We begin by asking the question:

- Under what conditions is the linear image of a closed convex cone closed?

This question is fundamental, due to its role in constraint qualifications in convex programming. Due to its importance, Chapter 9 in Rockafellar's classic text [39] is entirely devoted to it: see e.g. Theorem 9.1 therein. Surprisingly, the literature on the subject (beyond [39] and other textbooks) appears to be scant. Bauschke and Borwein [4] gave a necessary and sufficient condition for the continuous image of a closed convex cone to be closed. Their condition (due to its greater generality) is more involved than Theorem 9.1 in [39]. See also [2], and the references in [27]. See Borwein and Moors [8; 9] for proofs that the set of linear maps under which the image is not closed is small both in terms of measure and category.

For convenience we restate our question in an equivalent form:

- Given A and $K,$ when is A^*K^* closed?

In [27] we gave the very simple *necessary* condition

$$\mathcal{R}(A) \cap (\operatorname{cl}\operatorname{dir}(z, K) \setminus \operatorname{dir}(z, K)) = \emptyset, \tag{4.0.1}$$

for A^*K^* to be closed: here z is in the relative interior of $\mathcal{R}(A) \cap K,$ and $\operatorname{dir}(z, K)$ is the set of feasible directions at z in K. Note that (4.0.1) subsumes two seemingly unrelated classical sufficient conditions for the closedness of A^*K^*, as it trivially holds when K is polyhedral, or when $z \in \mathrm{ri}K$. It is also sufficient, when the set

$$K^* + F^\perp$$

is closed, where F is the minimal cone of $\mathcal{R}(A) \cap K$. Thus (4.0.1) becomes an exact characterization when $K^* + F^\perp$ is closed for *all* F faces of K. Such cones are called *nice*, and reassuringly, most cones that occur in optimization (such as polyhedral, semidefinite, and p-order cones) are nice. Nice cones also play a role in simpifying consstraint qualifications in conic LPs: see [11; 10].

As a byproduct of the preceding results, here we obtain an *exact* and simple characterization of when A^*K^* is closed when K is an arbitrary closed convex cone.

We build on the following basic fact:

$$A^*K^* \text{ is not closed } \Leftrightarrow \ (D) \text{ is weakly infeasible for some } c. \tag{4.0.2}$$

Theorem 9. The set A^*K^* is not closed, if and only if there is $(a_1, \dots, a_{k+1}) \in \mathrm{FR}_{k+1}(K^*)$ with $k \geq 1$, and $(y_1, \dots, y_{\ell+1}) \in \mathrm{FR}_{\ell+1}(K)$ with $\ell \geq 1$ such that

$$\begin{array}{rcl} a_i & \in & \mathcal{R}(A)\,(i = 1, \dots, k+1), \\ y_j & \in & \mathcal{N}(A^*)\,(j = 1, \dots, \ell) \end{array} \tag{4.0.3}$$

and

$$\langle a_i, y_{\ell+1} \rangle = \begin{cases} 0 & \text{if } i \leq k \\ -1 & \text{if } i = k+1. \end{cases} \tag{4.0.4}$$

Proof Starting with the forward implication, we choose c such that (D) is weakly infeasible. We take $(a_1, 0), \dots, (a_k, 0), (a_{k+1}, -1)$ as constraints in a reformulation that proves infeasibility of (D), and $(y_1, \ldots, y_{\ell+1})$ that proves that it is not strongly infeasible: cf. parts (1) and (2) in Theorem 6.

For the backward implication, fix $a := (a_1, \ldots, a_{k+1})$ and $y := (y_1, \ldots, y_{\ell+1})$ as stated. First we prove that they can be assumed to be pre-strict, so suppose that, say, a is not. Then

$$a_{i+1} \in (K \cap a_1^\perp \ldots a_i^\perp)^\perp \text{ for some } i < k.$$

Then $a_{i+1}^\perp \supseteq K \cap a_1^\perp \cdots \cap a_i^\perp,$ so

$$K \cap a_1^\perp \cdots \cap a_i^\perp \; = \; K \cap a_1^\perp \cdots \cap a_i^\perp \cap a_{i+1}^\perp,$$

so we can drop a_{i+1} from a while keeping all required properties of a and y. Continuing like this we arrive at both a and y being pre-strict, and to ease notation, we still assume $a \in \mathrm{FR}_{k+1}(K)$ and $y \in \mathrm{FR}_{\ell+1}(K^*)$.

Now $a_1, \ldots, a_k$ are linearly independent. Since $\langle a_{k+1}, y_{\ell+1} \rangle \neq 0$, so are $a_1, \ldots, a_k, a_{k+1}$.

Thus we can expand

$$A' = [a_1, \ldots, a_k, a_{k+1}, a_{k+2}, \ldots, a_m] \text{ a basis of } \mathcal{R}(A),$$

and let

$$c' = (0, \ldots, 0, -1, \langle a_{k+1}, y_{\ell+1} \rangle, \ldots, \langle a_m, y_{\ell+1} \rangle)^T.$$

Write $A' = TA,$ with T an m by m invertible matrix, and let $c = T^{-1}c'$. Then (D) with this c is weakly infeasible (since it has a reformulation with data (A', c') proving infeasibility; and $(y_1, \ldots, y_{\ell+1})$ proving that it is not strongly infeasible: cf. Theorem 6). □

It is also of interest to characterize nice cones. To review previous results on nice cones we recall that $y \in K^*$ is said to *expose* the face $K \cap y^\perp$; a face G of K is said to be *exposed*, if it equals $K \cap y^\perp$ for some $y \in K^*$; and it is not exposed iff

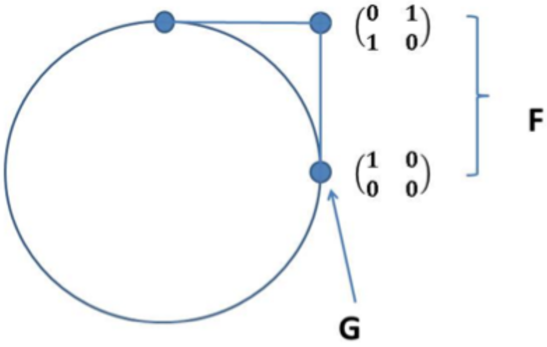
$$K^* \cap G^\perp = K^* \cap F^\perp \tag{4.0.5}$$

for some F face of K that strictly contains G (i.e. all vectors that expose G actually expose a larger face). The cone K is said to be facially exposed if all of its faces are exposed.

Example 12 below shows a cone which is not facially exposed.

Example 12. Define K be as the sum of $\mathcal{S}_+^2$ and the cone comprising all nonnegative multiples of the matrix

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



$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$
F
$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$
G

Figure 4.1: Cross section of a cone which is not facially exposed

The cross-section of this cone is shown on Figure 4.1, with faces $G \subsetneq F$ that satisfy (4.0.5).

For characterizations of nice cones, and a proof that they must be facially exposed, we refer to [29]; for an example of a facially exposed, but not nice cone, see [40]; and [15] for a proof that the linear pre-image of a nice cone is also nice.

Theorem 9 also characterizes when a cone is (not) nice:

Theorem 10. Let F be a face of K. Then $K^* + F^\perp$ is not closed, if and only if there is $(a_1, \dots, a_{k+1}) \in \mathrm{FR}_{k+1}(K^*)$ with $k \geq 1$, and $(y_1, \dots, y_{\ell+1}) \in \mathrm{FR}_{\ell+1}(K)$ with $\ell \geq 1$ such that

$$\begin{aligned} a_i &\in \operatorname{lin} F \,(i = 1, \ldots, k+1), \\ y_j &\in F^{\perp} \,(j = 1, \ldots, \ell) \end{aligned} \tag{4.0.6}$$

and

$$\langle a_i, y_{\ell+1} \rangle = \begin{cases} 0 & \text{if } i \leq k \\ -1 & \text{if } i = k+1. \end{cases} \tag{4.0.7}$$

Proof The result follows from Theorem 9 by considering a linear operator A with $\mathcal{R}(A) = \operatorname{lin} F$, $\mathcal{N}(A^*) = F^{\perp}$ and noting that A^*K^* is not closed, iff $\mathcal{N}(A^*) + K^*$ is. □

Theorems 9 and 10 provide a hierarchy of conditions, and it is natural to ask, how these relate to the simpler, but less general known conditions on closedness, and niceness. To address this question, we need a definition:

Definition 12. We say that the nonclosedness of A^*K^* (of $K^* + F^{\perp}$) has a $(k+1, \ell+1)$-proof, if there is $(a_1, \ldots, a_{k+1})$ and $(y_1, \ldots, y_{\ell+1})$ as in Theorem 9 (Theorem 10).

Theorem 11. The following hold:

(1) Suppose that condition (4.0.1) is violated, and let ℓ be the degree of singularity of $\mathcal{R}(A) \cap K$. Then there is a $(2, \ell+1)$-proof of the nonclosedness of A^*K^*.

(2) Suppose that K has a nonexposed face, say G, and F is the smallest exposed face of K that contains it. Then there is a $(2,2)$-proof that $K^* + F^{\perp}$ is not closed.

Since the proof of this result is somewhat technical, we defer it to Appendix B. It is also natural to ask, as to what values of k and ℓ are actually necessary to prove nonclosedness of A^*K^* (or of $K^* + F^{\perp}$). A recent result of Drusviyatsky et al [17] shows a surprising connection between the degree of singularity of the dual problem (D), and the exposedness of the smallest face of A^*K^* that contains c. It would also be interesting to explore the connection to this result and we will do so in a followup paper.

Also, in recent work, Roshchina and Tunçel gave a condition to strengthen the facial exposedness condition of [29]: it would be interesting to see how their condition fits into our hierarchy.

We next turn to another basic question in the theory of conic LPs: given a (weakly) infeasible conic linear system

$$H \cap K, \tag{4.0.8}$$

where H is an affine subspace, what is the maximal/minimal dimension of an affine subspace H' with $H' \supseteq H$ (or $H' \subseteq H$) such that $H' \cap K$ has the same feasibility status as (4.0.8)?

Note that by (weak) infeasibility of (4.0.8) we mean (weak) infeasibility of a representation in either the primal (P) or the dual (D) form.

For instance, if K is polyhedral, and (4.0.8) is infeasible, then by Farkas' lemma we can take H' as an affine subspace defined by a single equality constraint. To further illustrate this question, consider the semidefinite system

$$\begin{aligned} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \bullet y &= 0, \\ \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \bullet y &= -1, \end{aligned}$$

which is weakly infeasible. Dropping the first constraint keeps it weakly infeasible, and so does adding a constraint that fixes y_{12} to zero.

To state our main result, we recall that ℓ_K denotes the length of the longest chain of faces in K (see Definition 6). Assume $K = \mathcal{S}_+^n$. Then in part (1) of Theorem 12 below the bound n in place of ℓ_K follows from Theorem 1 in [21] when $K = \mathcal{S}_+^n$ and the bound on the dimension of H'' in part (4) follows from [22].

Theorem 12. The following hold.

(1) If (4.0.8) is infeasible, then there is $H' \supseteq H$ such that

$$\operatorname{codim} H' \leq \ell_K \text{ and } H' \cap K \text{ is infeasible.}$$

(2) If (4.0.8) is not strongly infeasible, then there is $H'' \subseteq H$ such that

$$\dim H'' \leq \ell_K - 1 \text{ and } H'' \cap K \text{ is not strongly infeasible.}$$

(3) If (4.0.8) is weakly infeasible, then there is $H'' \subseteq H \subseteq H'$ as in parts (1) and (2) such that

$$H' \cap K \text{ and } H'' \cap K \text{ are both weakly infeasible.}$$

(4) If $K = K^*$ is the cone of psd matrices, then the bounds above can be tightened to

$$\operatorname{codim} H' \leq \ell_{\mathcal{S}_+^n} - 1 = n, \text{ and } \dim H'' \leq \ell_{\mathcal{S}_+^n} - 2 = n - 1.$$

Proof For part (1) we first represent (4.0.8) as a dual type problem (D) (with K in place of K^*), and apply part (1) of Theorem 6. We let H' be the affine subspace defined by the first $k+1$ constraints in (D_{ref}), and by pre-strictness of $a_1', \ldots, a_k'$ we deduce

$$\operatorname{codim} H' = k + 1 \leq \ell_K - 1 + 1 = \ell_K,$$

as required. For part (2) we represent (4.0.8) as a primal type problem (P) and apply part (4) of Theorem 6. We let H'' be spanned by the first ℓ generators and the right hand side in (P_{ref}). By pre-strictness, we find

$$\ell \leq \ell_K - 1,$$

and this completes the proof.

For part (3) we choose H' as in part (1). Since (4.0.8) is not strongly infeasible, and $H' \supseteq H,$ the system $H' \cap K$ is also not strongly infeasible. We construct H'' as in part (2) with an analogous justification.

For part (4) suppose that (4.0.8) is infeasible, represent it as a dual problem (with $\mathcal{S}_+^n$ in place of K^*) and apply part (1) of Theorem 7, by which in the reformulated system (D_{ref}) we can drop all but $k+1$ constraints while keeping it infeasible. We have

$$k \leq \ell_{\mathcal{S}_+^n} - 1 = n.$$

If $k < n,$ then there is nothing to prove. If $k = n,$ then letting $a''_{n-1} = a'_{n-1} + \lambda a'_n$ for some suitable $\lambda > 0$ we have that the lower right two by two block of a''_{n-1} is positive definite, hence in this case a subsystem with $n-1$ constraints is infeasible. (In fact, one can show inductively that a linear combination of all $a'_1, \ldots, a'_n$ is positive definite, so in this case a subsystem with only two constraints is infeasible.) □

BIBLIOGRAPHY

[1] Abdo Alfakih. On Farkas lemma and dimensional rigidity of bar frameworks. Technical report, http://arxiv.org/abs/1405.2301, 2014.

[2] Alfred Auslender. Closedness criteria for the image of a closed set by a linear operator. *Numer. Funct. Anal. Optim.*, 17:503–515, 1996.

[3] George Phillip Barker and David Carlson. Cones of diagonally dominant matrices. *Pacific J. Math.*, 57:15–32, 1975.

[4] Heinz Bauschke and Jonathan M. Borwein. Conical open mapping theorems and regularity. In *Proceedings of the Centre for Mathematics and its Applications 36*, pages 1–10. Australian National University, 1999.

[5] Aharon Ben-Tal and Arkadii Nemirovskii. *Lectures on modern convex optimization.* MPS/SIAM Series on Optimization. SIAM, Philadelphia, PA, 2001.

[6] Grigoriy Blekherman, Pablo Parrilo, and Rekha Thomas, editors. *Semidefinite Optimization and Convex Algebraic Geometry.* MOS/SIAM Series in Optimization. SIAM, 2012.

[7] Frédéric J. Bonnans and Alexander Shapiro. *Perturbation analysis of optimization problems.* Springer Series in Operations Research. Springer-Verlag, 2000.

[8] Jonathan M. Borwein and Warren B. Moors. Stability of closedness of convex cones under linear mappings. *J. Convex Anal.*, 16(3–4), 2009.

[9] Jonathan M. Borwein and Warren B. Moors. Stability of closedness of convex cones under linear mappings. *Journal of Nonlinear Analysis and Optimization: Theory & Applications*, 1(1), 2010.

[10] Jonathan M. Borwein and Henry Wolkowicz. Facial reduction for a cone-convex programming problem. *J. Aust. Math. Soc.*, 30:369–380, 1981.

[11] Jonathan M. Borwein and Henry Wolkowicz. Regularizing the abstract convex program. *J. Math. Anal. App.*, 83:495–530, 1981.

[12] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[13] Vris Cheung, Henry Wolkowicz, and Simon Schurr. Preprocessing and regularization for degenerate semidefinite programs. Technical report, University of Waterloo, 2012.

[14] Yuen-Lam Cheung. *Preprocessing and Reduction for Semidefinite Programming via Facial Reduction: Theory and Practice.* PhD thesis, University of Waterloo, 2013.

[15] Check-Beng Chua and Levent Tunçel. Invariance and efficiency of convex representations. *Math. Program. B*, 111:113–140, 2008.

[16] Robert Connelly and Steven Gortler. Iterative universal rigidity. Technical report, http://arxiv.org/abs/1401.7029, 2014.

[17] Dimitry Drusviyatsky, Gábor Pataki, and Henry Wolkowicz. Coordinate shadows of semi-definite and euclidean distance matrices. *SIAM J. Opt.*, 2015.

[18] Osman Güler. *Foundations of Optimization.* Graduate Texts in Mathematics. Springer, 2010.

[19] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 1995.

[20] 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.

[21] Minghui Liu and Gábor Pataki. Exact duality in semidefinite programming based on elementary reformulations. *SIAM J. Opt.*, 2015.

[22] Bruno Lourenco, Masakazu Muramatsu, and Takashi Tsuchiya. A structural geometrical analysis of weakly infeasible SDPs. *Optimization Online.*

[23] 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.

[24] Tim Netzer, Daniel Plaumann, and Markus Schweighofer. Exposed faces of semidefinitely representable sets. *SIAM J. Opt.*, 20(4):1944–1955, 2010.

[25] David Papp and Farid Alizadeh. Semidefinite characterization of sum-of-squares cones in algebras. *SIAM J. Opt.*, 23(3):1398–1423.

[26] Gábor Pataki. The geometry of semidefinite programming. In Romesh Saigal, Lieven Vandenberghe, and Henry Wolkowicz, editors, *Handbook of semidefinite programming.* Kluwer Academic Publishers, also available from www.unc.edu/~pataki, 2000.

[27] Gábor Pataki. On the closedness of the linear image of a closed convex cone. *Math. Oper. Res.*, 32(2):395–412, 2007.

[28] 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.

[29] Gábor Pataki. On the connection of facially exposed and nice cones. *J. Math. Anal. App.*, 400:211–221, 2013.

[30] 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.

[31] Frank Permenter and Pablo Parrilo. Partial facial reduction: simplified, equivalent sdps via approximations of the psd cone. Technical report, http://arxiv.org/abs/1408.4685, 2014.

[32] Imre Pólik and Tamás Terlaky. Exact duality for optimization over symmetric cones. Technical report, Lehigh University, Betlehem, PA, USA, 2009.

[33] Imre Pólik and Tamás Terlaky. New stopping criteria for detecting infeasibility in conic optimization. *Optim. Lett.*, 3(2):187–198, 2012.

[34] J. Scott Provan and Douglas R. Shier. A paradigm for listing (s, t)-cuts in graphs. *Algorithmica*, 15(4):351–372, 1996.

[35] Motakuri V. Ramana. An exact duality theory for semidefinite programming and its complexity implications. *Math. Program. Ser. B*, 77:129–162, 1997.

[36] Motakuri V. Ramana, Levent Tunçel, and Henry Wolkowicz. Strong duality for semidefinite programming. *SIAM J. Opt.*, 7(3):641–662, 1997.

[37] R.C. Read and R.E. Tarjan. Bounds on backtrack algorithms for listing cycles, paths, and spanning trees. *Networks*, 5:237–252, 1975.

[38] James Renegar. *A Mathematical View of Interior-Point Methods in Convex Optimization.* MPS-SIAM Series on Optimization. SIAM, Philadelphia, USA, 2001.

[39] Tyrrel R. Rockafellar. *Convex Analysis.* Princeton University Press, Princeton, NJ, USA, 1970.

[40] Vera Roshchina. Facially exposed cones are not nice in general. *SIAM J. Opt.*, 24:257–268, 2014.

[41] Romesh Saigal, Lieven Vandenberghe, and Henry Wolkowicz, editors. *Handbook of semidefinite programming.* Kluwer Academic Publishers, 2000.

[42] Rainer Sinn and Bernd Sturmfels. Generic spectrahedral shadows. Technical report, http://arxiv.org/abs/1407.5219, 2014.

[43] Jos Sturm. Implementation of interior point methods for mixed semidefinite and second order cone optimization problems. *Optimization Methods and Software*, 17(6):1105–1154, 2002.

[44] Michael J. Todd. Semidefinite optimization. *Acta Numer.*, 10:515–560, 2001.

[45] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM Review*, 1996.

[46] Cynthia Vinzant. What is ... a spectrahedron? *Notices Amer. Math. Soc.*, 61(5):492–494, 2014.

[47] Hayato Waki. How to generate weakly infeasible semidefinite programs via Lasserre's relaxations for polynomial optimization. *Optim. Lett.*, 6(8):1883–1896, 2012.

[48] Hayato Waki and Masakazu Muramatsu. Facial reduction algorithms for conic optimization problems. Technical Report Report CS-09-01, Department of Computer Science, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182-8585 Japan, 2009.

[49] Hayato Waki and Masakazu Muramatsu. Facial reduction algorithms for conic optimization problems. *J. Optim. Theory Appl.*, 158(1):188–215, 2013.